Enhanced finite element methods using neural networks

Nicolas Victorion[1], Frédérique Lecourtier[2], Hélène Barucq[3], Michel Duprez[4], Florian Faucher[6], Emmanuel Franck[5], Vanessa Lleras[7], and Victor Michel-Dansac[8]

[5]Macaron team, INRIA Nancy grand Est, IRMA
[8]Macaron team, INRIA Nancy grand Est, IRMA
[1]Makutu team, INRIA Bordeaux, TotalEnergies
[3]Makutu team, INRIA Bordeaux, TotalEnergies
[6]Makutu team, INRIA Bordeaux, TotalEnergies
[2]Mimesis team, INRIA Nancy grand Est, Icube
[4]Mimesis team, INRIA Nancy grand Est, Icube
[7]Montpellier University

May 6, 2024

Abstract

Your abstract.

1 Introduction

(TOODOO: plus complet) The finite element method is one of the reference methods for solving linear elliptic PDEs, which will be considered here. The method starts by constructing a finite-dimensional vector space from basis functions localized to the degrees of freedom of the mesh. Once this space has been constructed, the weak form of the equation is projected onto the finite-dimensional space. The method converges with an error proportional to the characteristic mesh size. However, it requires the inversion of a potentially large matrix. If you have to solve the problem thousands of times (as in optimal control or uncertainty propagation), the cost starts to become prohibitive.

In recent years, learning-based alternatives such as PINNs [**?**]-[**?**] and Deep Ritz [**?**] methods have emerged. The idea is to approximate the solution to our problem using a network with physical coordinates as input. These methods are currently not competitive with finite element methods for classical PDE problems, mainly due to the lack of precision of network-based methods. However, these methods become interesting in large physical or parametric dimensions [**?**].

The idea here is to use a parametric PINNs to compute a large family of offline solutions and then compute an online solution for a parameter by inferring the network and correcting the network solution using very small finite elements. This will make it possible to obtain a method capable of quickly predicting an online solution while retaining convergence properties. In the following we will propose two finite element correction approaches. For both approaches, we will explain the finite element error as a function of the network error relative to the true solution. These estimators will show both that the correction can be done with a very small mesh if the prediction of the network is reasonable and how the network must be trained to be efficient.

1.1 Problem under consideration

TODO: add function spaces for u, v, r, a, d, f and g

Idée général : * garder une seule notation $\|\cdot\|_k$ ou $\|\cdot\|_{H^k}$?
* $|\ | \to |\text{left}|\cdot|\text{right}|$

In this paper, we consider general elliptic equations with Dirichlet boundary conditions, defined on a space domain $\Omega \subset \mathbb{R}^d$. The problem of interest is given by:

$$\begin{cases} r(x)u(x) + a(x) \cdot \nabla u(x) - \dfrac{1}{P_e}\nabla \cdot (d(x)\nabla u(x)) = f(x), & \forall x \in \Omega, \\ u(x) = g(x), & \forall x \in \partial\Omega, \end{cases} \tag{1}$$

where u is the unknown function, r is the reaction coefficient, a the convection coefficient, d the diffusion coefficient, f a source term, g a boundary condition, and P_e Péclet number, which models the ratio between convection and diffusion. The finite element method used in this work relies on the weak form of (1). To obtain this weak form, one multiplies the PDE (1) by a test function v, and integrates the result on Ω. The final weak form is obtained after arguing the divergence theorem on the second-order term, and we get

$$\text{Seek } u \in H_0^m(\Omega) \text{ such that, } \forall v \in H_0^m(\Omega),\ a(u,v) = l(v). \tag{2}$$

In (2), we have defined a bilinear form

$$a(u,v) = \int_\Omega r(x)\, u(x)\, v(x)\, dx + \int_\Omega v(x)\, (a(x) \cdot \nabla u(x))\, dx + \frac{1}{P_e}\int_\Omega (d(x)\nabla u(x), \nabla v(x))\, dx$$

and a linear form:

$$l(v) = \int_\Omega f(x)\, v(x) dx.$$

2 Classical finite element method

The goal of this section is to recall the classical finite element method, and to introduce several important definitions. The finite element method consists in restricting the weak formulation (2) to an approximate function space, denoted by V_h. To construct V_h, we consider a mesh Ω_h of the space domain Ω, defined by

$$\Omega_h = \bigcup_{i=1}^{N} K_i,$$

where each cell $K_i \subset \mathbb{R}^d$ is a regular polygon.

Definition 2.1. *A finite element is a triplet $\{K, P, \Sigma\}$, where*

- *K is a compact Lipschitz subset of $\mathbb{R}^d$,*
- *P is a vector space of functions $p : K \to \mathbb{R}^d$, Victor: n'importe quelles fonctions ? pas de continuité etc demandée ?*
- *Σ is a set of $q+1$ linear forms acting on P, such that the linear mapping*

$$G : p \in P \mapsto (\sigma_0(p), \dots, \sigma_q(p)) \in \mathbb{R}^q$$

*is bijective. The linear forms $\{\sigma_0, \dots, \sigma_q\}$ are called **local degrees of freedom**.*

Definition 2.2. *A finite element is **P-unisolvant** if each element of P is uniquely determined by the **local degree of freedom**.*

Proposition 2.3. *A finite element is **P-unisolvant** if*

$$\operatorname{Dim} P = \operatorname{Card} \Sigma = q+1 \qquad \text{and} \qquad \forall \sigma \in \Sigma,\ \sigma(\phi) = 0 \implies \phi = 0.$$

Definition 2.4. *Define the basis $\{\phi_0, \dots, \phi_q\}$ of P such that*

$$\forall\, i, j \in \{0, \dots, q\}, \quad \sigma_i(\phi_j) = \delta_{ij}.$$

*The elements $(\phi_i)_{i \in \{0,\dots,q\}}$ are called **local shape functions**.*

Thanks to this general definition, one can construct local basis function in each element, supported by degrees of freedom. For example, in the case of the $\mathbb{P}_k$ Lagrange approximation, the degrees of freedom are pointwise values within the element or on its boundary, and the local shape functions are polynomials of degree k or lower.

Definition 2.5 (Local interpolator)**.** *Consider a finite element (K, P, Σ). Let V_K be a function space of functions $v \in K \to \mathbb{R}^d$. Victor: n'importe quelles fonctions ? pas de continuité etc demandée ? The local interpolation operator $\mathcal{I}_K$ is then defined by:*

$$\forall f \in V_K, \qquad \mathcal{I}_K f(x) = \sum_{i=1}^{q} \sigma_i(f)\phi_i(x).$$

Definition 2.6 (Global interpolation)**.** *We consider a mesh Ω_h and a family N of finite element $\{K, P_K, \Sigma_K\}_{K\in\Omega_h}$. We define the global*

$$W = \left\{v_h \in L^1(\Omega_h); \forall K \in \Omega_h, v_{|K} \in V_k\right\}$$

with V_k a space function on the element and the global approximation space:

$$W_h = \left\{v_h \in L^1(\Omega_h); \forall K \in \Omega_h, v_{|K} \in P_{\theta,K}\right\}$$

*On this approximation space we define a **global interpolation operator** which create function in the approximation space:*

$$\forall f \in W, \quad \mathcal{I}_h f(x) = \sum_{k=1}^{N}\sum_{i=1}^{q} \sigma_{k,i}(f_{i|k})\phi_{k,i}(x)$$

with $\phi_{k,i}(x)$ implicitly extend to zero outside the element k

For the moment we don't have continuity of our approximation space. To construct the H_1 conformal approximation space we must impose the continuity and obtain the final approximation space:

$$V_h = \left\{v_h \in W_h, \quad \forall f \in \mathcal{F}, v_{h|K_l}(\mathbf{x}_{K_l,i}) = v_{h|K_r}(\mathbf{x}_{K_r,i})\right\}$$

with K_l and K_r the left and right cell associated to the face f. The classical finite element method use a same space to represent the solution and for the test function but we can use different element space. We speak about the Petrov-Galerkin approach which consist to solve:

$$\text{Seek } u_h \in U_h \text{ such that, } \forall v_h \in V_h,\ a(u_h, v_h) = f(v_h). \tag{3}$$

Equipped with the basics of finite element methods, we now propose, in Sections 3 and 4, two methods based on enhancing Lagrange finite elements. Note that they could be extended to other element types, but we chose Lagrange elements for simplicity.

3 Enhanced Finite element with additive prior

From now on, we consider that we have prior knowledge of the solution This prior is denoted by $u_\theta \in C_0^m(\Omega)$. At this level, we assume that this prior on the solution is known; its construction will be discussed later. The objective is to develop a finite element method that can leverage this prior knowledge to decrease its error. In the classical finite element method, the solution is projected on the vector space V_h. Here, we propose a method based on the Petrov-Galerkin approximation. In the Petrov-Galerkin approximation, the trial space to represent the solution and the test space for the basis function are different. Here we propose to use an affine trial space

$$U_h = \{u_h = u_\theta + p_h, \quad p_h \in V_h\}$$

and to use our classical vector space V_h as a test space. Since we have assumed that $u_\theta \in C_0^m(\Omega)$, U_h is also a subset of $H_0^m(\Omega)$, like V_h. This leads to the following reformulation of the Petrov-Galerkin approach (3):

$$\text{Seek } u_h \in U_h \text{ such that, } \forall v_h \in V_h,\ a(u_\theta + p_h, v_h) = f(v_h). \tag{4}$$

In our case, using the affine space U_h for the solution is equivalent to writing a classical finite element method approximating the residual between the solution and the prior. Indeed, since a is bilinear, (4) can be further reformulated as

$$\text{Seek } u_h \in U_h \text{ such that, } \forall v_h \in V_h,\ a(p_h, v_h) = f(v_h) - a(u_\theta, v_h). \tag{5}$$

This Petrov-Galerkin problem then becomes a classical Galerkin approximation with a modified source term. In practice, we solve the modified problem (5) with modified boundary conditions. Indeed, if our Dirichlet problem satisfies $u(x) = g(x)$ on $\partial\Omega$, then p_h has to satisfy

$$p_h = g - u_\theta \quad \text{on } \partial\Omega,$$

which become homogeneous boundary conditions if the prior is exact at the boundary.

3.1 Convergence analysis

In this subsection we propose to prove that the finite element method associated to 4 is convergent and we will explicit the error estimate using the a error estimation associated to the prior. This will demonstrate the potential benefits of the approach.

Lemma 3.1. *We consider the u the solution of problem (2) satisfying the Lax-Milgram theorem with a coercivity constant α. The discrete solution of the finite element method associated to problem (4) with V_h a $\mathbb{P}_q$ and Lagrange space and $u_\theta(x) \in C_0^m(\Omega)$ so we obtain*

$$\| u - u_h \|_m \leq C_{gain} \left(\frac{M}{\alpha} C h^{q+1-m} \mid u \mid_{H^{q+1}} \right)$$

with

$$C_{gain} = \frac{\mid u - u_\theta \mid_{H^{k+1}}}{\mid u \mid_{H^{k+1}}}$$

Proof. This prove is just a small adaptation of the Cea lemma.For the additive prior method the numerical solution is given by

$$u_h(x) = u_\theta(x) + p_h(x).$$

with $p(x) \in V_h \subset V$. We have

$$a(u - u_h, u - u_h) = a((u - u_\theta) - p_h, (u - u_\theta) - p_h)$$
$$a(u - u_h, u - u_h) = a((u - u_\theta) - p_h, (u - u_\theta) - p_h - v_h + v_h), \quad \forall v_h \in V_h$$
$$a(u - u_h, u - u_h) = a((u - u_\theta) - p_h, (u - u_\theta) - v_h) + a((u - u_\theta) - p_h, v_h - p_h), \quad \forall v_h \in V_h$$

Since we have $a(u, v_h) = l(v_h), \quad \forall v_h \in V_h$ (we use that $V_h \subset V$) and $a(u_h, v_h) = l(v_h), \quad \forall v_h \in V_h$ we have

$$a(u - u_h, v_h) = a((u - u_\theta) - p_h, v_h) = 0, \quad \forall v_h \in V_h$$

Since we have $v_h - p_h \in V_h$ (it is a vector space) we have

$$a(u - u_h, u - u_h) = a((u - u_\theta) - p_h, (u - u_\theta) - v_h), \quad \forall v_h \in V_h$$

Using the coercivity for the left term and the Cauchy-Schwartz inequality for the right term we have

$$\alpha \| u - u_h \|_m^2 \leq M \| (u - u_\theta) - p_h \|_m \| (u - u_\theta) - v_h \|_m$$

using $u - u_h = (u - u_\theta) - p_h$ we have

$$\| u - u_h \| \leq \frac{M}{\alpha} \| (u - u_\theta) - v_h \| = \frac{M}{\alpha} \| (u - u_\theta) - I_h(u - u_\theta) \|$$

Now we use the classical interpolation result on global interpolant find [?] on the interpolation gives us:

$$\|v - \mathcal{I}_h(v)\|_{H^m} \leq C h^{q+1-m} \mid v \mid_{H^{q+1}}$$

So using this result we obtain:

$$\| u - u_h \| \leq \frac{M}{\alpha} C h^{q+1-m} \mid u - u_\theta \mid_{H^{q+1}}$$

□

If we rewrite to introduce the error associated to the classical finite element without prior we obtain

$$\| u - u_h \|_m \leq C_{gain} \left(\frac{M}{\alpha} C h^{q+1-m} \mid u \mid_{H^{q+1}} \right)$$

with

$$C_{gain} = \frac{\mid u - u_\theta \mid_{H^{k+1}}}{\mid u \mid_{H^{k+1}}}$$

Remark 3.2. *This constant show that if the prior is close the solution the constant of convergence associated to the finit element method we will be smaller. So the finite element with additive prior will be more accurate. However the gain in related to the L^2 error associated to the $q+1$ derivatives. This results shows that the prior* ***must approximate accuratly the derivatives of the solution more than the solution herself****.*

4 Enhanced Finite element method with multiplicative prior

In this section we propose a second possibility to enrich the finite element discretization with a prior. In the previous section we use to prior to define affine trial space. Here we propose a version when we enrich directly the finite dimensional vector space V_h. Firstly we will introduce the construction of the new finite element space and in a second time we will gives a convergence estimate for this new space depending to the error between the prior and the solution.

Definition 4.1. *We consider $u_\theta(x) \in C^1(\mathbb{R}, \mathbb{R}^+)$. The 1D* ***modified*** *Lagrange finite element are given by:*

- $K = [x_0, ..., x_q]$
- $P_{\theta,K} = \left\{ \bar{\phi} = \phi u_\theta, \quad \phi \in \mathbb{P}_1([x_0, ..., x_q]) \right\}$
- $\Sigma_{\theta,K} = \{\bar{\sigma}_0, ..., \bar{\sigma}_q\}$. *The $q+1$ local degrees of freedom are given by*

$$\forall \bar{\phi} \in P_{\theta,K}, \quad \bar{\sigma}_i(\bar{\phi}) = \frac{\bar{\phi}(x_i)}{u_\theta(x_i)} = \sigma_i(\phi), \quad 1 \leq i \leq q$$

- *The local shape functions:*

$$\bar{\phi}_i(x) = \phi_i(x) u_\theta(x)$$

with ϕ_i the local shape function associated to the classical $\mathbb{P}_K([x_0, ..., x_q])$ element

Proposition 4.2. *The 1D modified Lagrange finite $(K, P_{\theta,K}, \Sigma_{\theta,K})$ element is unisolvent.*

Proof. It is trivial and given by the unisolvent property of the classical finite element. We will compute $\bar{\sigma}_i(\bar{\phi})$ with $\bar{\phi} \in P_{\theta,K}$. We obtain

$$\bar{\sigma}_i(\bar{\phi}) = \frac{\bar{\phi}(x_i)}{u_\theta(x_i)} = \phi(x_i) = \sigma_i(\phi)$$

since σ and ϕ are associated to the classical finite element which are unisolvent so it is also the case for the modified finite element space. □

Proposition 4.3. *For the element $(K, P_{\theta,K}, \Sigma_{\theta,K})$, the space $P_{\theta,K}$ is invariant for $\bar{\mathcal{I}}_h$ the local interpolator:*

$$\bar{\mathcal{I}}_h(f) = \sum_{i=0}^{q} \bar{\sigma}_i(f) \bar{\phi}_i(x)$$

Proof. We consider $\phi(x) \in P_{\theta,K}$ so we have

$$\phi = \sum_{i=0}^{q} a_i \bar{\phi}_i(x)$$

Now we compute the interpolation of this function. We have

$$\sum_{i=0}^{q} \sigma_i(\sum_{j=0}^{q} a_j \bar{\phi}_j(x))\bar{\phi}_i(x)$$

$$\sum_{i=0}^{q} \left(\sum_{j=0}^{q} \frac{a_j}{u_{\theta,K}(x_i)} \bar{\phi}_j(x_i) \right) \bar{\phi}_i(x)$$

$$\sum_{i=0}^{q} \left(\sum_{j=0}^{q} a_j \phi_j(x_i) \right) \bar{\phi}_i(x)$$

using the property of the classical basis we obtain:

$$\sum_{i=0}^{q} a_i \bar{\phi}_i(x) = 0$$

□

Proposition 4.4. *The **global shape function** $(\bar{\phi}_1, ..., \bar{\phi}_N)$ defined by*

$$\bar{\phi}_{i|k}(x) \in P_{\theta,K}$$

and

$$\bar{\phi}_i(x_j) = \frac{\phi_i(\mathbf{x}_j)}{u_\theta(\mathbf{x}_j)} = \delta_{ij}, \quad 1 \leq j \leq N_{vertices}$$

can be written as $\bar{\phi}_i(x) = \phi_i(x) u_\theta(x)$ with $\phi_i(x)$ the classical global shape function of the $\mathbb{P}_q$ finite element, is a basis of V_h and $\{\sigma_1, ...\sigma_{N_{vertices}}\}$ such that

$$\sigma_i(\phi) = \frac{\phi(\mathbf{x}_i)}{u_\theta(\mathbf{x}_i)}$$

is a basis for V_h^*.

Proof. We denote the classical approximation space by:

$$\bar{V}_h = \left\{ \bar{v}_h \in \bar{W}_h, \quad \forall f \in \mathcal{F}, \bar{v}_{h|K_l}(\mathbf{x}_{K_l,i}) = \bar{v}_{h|K_r}(\mathbf{x}_{K_r,i}) \right\}$$

with

$$\bar{X}_h = \left\{ \bar{v}_h \in L^1(\Omega); \forall K \in \Omega_h, \bar{v}_{|K} \in \mathcal{P}_1(K) \right\}$$

Let the linear application L such that

$$\begin{aligned} L: \quad \bar{V}_h &\to V_h \\ \bar{\phi} &\mapsto \phi = \bar{\phi} u_\theta \end{aligned}$$

L is bijective, then L is an isomorphism between $\bar{V}_h$ and V_h.
Since $\{\bar{\phi}_0, .., \bar{\phi}_N\}$ is a basis of $\bar{V}_h$ then $\{L(\bar{\phi}_0), .., L(\bar{\phi}_N)\}$ is a basis of V_h.
Hence $\{\phi_0, .., \phi_N\}$ is a basis of V_h.
Moreover, one has

$$\forall (i,j) \in \{1, ..N_{\text{vertices}}\}^2 \quad \sigma_i(\phi_j) = \delta_{i,j}$$

Then $\{\sigma_0, .., \sigma_N\} = \{\phi_0^*, .., \phi_N^*\}$ is the dual basis associated to $\{\phi_0, .., \phi_N\}$.

□

Proposition 4.5. *The global interpolation operator can be rewrite as:*

$$\bar{\mathcal{I}}_h : v \in C_0(\Omega) \to \sum_{i=1}^{N_{vertices}} \sigma_i(v) \bar{\phi}_i(x) \in \bar{V}_h$$

with the global basis function given by $\widetilde{\phi_i}(x) = u_\theta(x) \bar{\phi}_i(x)$ with $\bar{\phi}_i(x)$ the classical global shape function of the P_k elements.

Now that we have introduced our new approximation space V_h and demonstrated some classical properties, we will show that the error in this new approximation space is better than the error in the classical approximation P_k. We will then comment on the technical details associated with this new space.

4.1 Convergence results and analysis

In this section we propose to prove that the modified FE method converge with the same type of estimate than the classical one. We will focus on the homogeneous boundary Dirichlet problem. The definition of the global degree of freedom associated to our new approximation space need that $u_\theta(x) \neq 0$. Since the solution can apriori pass through zero, we propose to look at the following modified problem:

$$\begin{cases} \text{Seek } u_M = M + u_0 \in H_0^m(\Omega), & \text{such that} \\ a(u_0, v) = l(v), & \forall v \in H_0^m(\Omega) \end{cases} \tag{6}$$

where M is a constant choosen that $u_M(x) > 0$. Since $u_\theta(x)$ will approximate the solution $u'_M(x)$ it will stay positive also. Solving this problem we recover the solution make $u(x) = u_M(x) - M$.

4.1.1 Convergence result for FE with multiplicative prior

Here we assume that the solution is always non equal to zero. For that we use the trick proposed just before.

Lemma 4.6. *We consider the u the solution of problem (2) satisfying the Lax-Milgram theorem with a coercivity constant α. The discrete solution of the finite element method associated to problem (4) with $\bar{V}_h$ based on $\mathbb{P}_q$ Lagrange finite element space and $u_\theta(x) \in C_0^m(\Omega)$ so we obtain*

$$\| u - u_h \|_m \leq C_{gain} \left(\frac{M}{\alpha} h^{q+1-m} \mid u \mid_{H^{q+1}} \right)$$

with

$$C_{gain} = C \left| \frac{u}{u_\theta} \right|_{H^{q+1}} \frac{\mid u_\theta \mid_{W^{m,\infty}}}{\mid u \mid_{H^{q+1}}}$$

Proof. To begin we use the classical Cea lemma which say that

$$\| u - u_h \|_m \leq \frac{M}{\alpha} \| u - \bar{\mathcal{I}}_h u \|_{H^m}$$

After that we use the proposition (4.5) to rewrite the interpolation

$$\|u - \bar{\mathcal{I}}_h(u)\|_{H^m} = \|u - \mathcal{I}_h(\frac{u}{u_\theta})u_\theta\|_{H^m}$$

Considering Ω a bounded domain, it is assumed that u_θ belongs to $C^\infty(\Omega)$, then there exists $M_n \geq 0$ depending on the dimension n such that

$$\|u - \bar{\mathcal{I}}_h(u)\|_{H^m} \leq M_n \|u_\theta\|_{W^{m,\infty}} \left\| \frac{u}{u_\theta} - \mathcal{I}_h(\frac{u}{u_\theta}) \right\|_{H^m}$$

TODO

DOO too (more details) with $\mathcal{I}_h$ the classical finite element associated to the $\mathbb{P}_q$ elements. The classical theory gives uses that

$$\|v - \mathcal{I}_h(v)\|_{H^m} \leq C h^{q+1-m} \mid v \mid_{H^{q+1}}$$

Using that we obtain

$$\|u - \bar{\mathcal{I}}_h(u)\|_{H^m} \leq M_n \|u_\theta\|_{W^{m,\infty}} C h^{q+1-m} \mid \frac{u}{u_\theta} \mid_{H^{q+1}} \tag{*}$$

To conclude we just rewrite the term to obtain the gain term compare to the classical error. □

4.1.2 Analysis of the M parameter

Now we will analyse the effect of the factor M on the error and the potential gain of the method. We consider the problem:

$$\begin{cases} \mathcal{L}(u, \partial_x u, \partial_{xx} u) = f \\ u = 0 \end{cases}$$

We assume that for this problem we have a prior u_θ. If the solution can vanish somewhere we solve the modified problem given by

$$\begin{cases} \mathcal{L}(\hat{u}, \partial_x \hat{u}, \partial_{xx} \hat{u}) = f \\ \hat{u} = M \end{cases}$$

with $u_M(x) = u(x) + M$ and $u_{\theta,M}(x) = u_\theta(x) + M$. When we apply the error estimate to the second problem. So we obtain

$$\|\hat{u} - \mathcal{I}_h(\hat{u})\|_m \leq |u_{\theta,M}|_{W^{m,\infty}} C h^{q+1-m} \mid \frac{\hat{u}}{u_{\theta,M}} \mid_{H^{q+1}}$$

which is equivalent to

$$\|\hat{u} - \mathcal{I}_h(\hat{u})\|_m \leq |u_\theta|_{W^{m,\infty}} C h^{q+1-m} \mid \frac{u+M}{u_\theta + M} \mid_{H^{q+1}}$$

We assume that $u(x) = u_\theta(x) + \epsilon p(x)$ with $|p|_{W^{m,\infty}} \leq 1$. Plugging this we obtain:

$$\|\hat{u} - \mathcal{I}_h(\hat{u})\|_m \leq C h^{q+1-m} \mid \frac{u_\theta + M + \epsilon p}{u_\theta + M} \mid_{H^{q+1}} (\|M + u\|_{W^{m,\infty}} + \epsilon)$$

$$\|\hat{u} - \mathcal{I}_h(\hat{u})\|_m \leq C h^{q+1-m} \mid 1 + \frac{\epsilon p}{u_\theta + M} \mid_{H^{q+1}} (|M + u|_{W^{m,\infty}} + \epsilon)$$

$$\|\hat{u} - \mathcal{I}_h(\hat{u})\|_m \leq \epsilon C h^{q+1-m} \mid \frac{p}{u_\theta + M} \mid_{H^{q+1}} (|M + u|_{W^{m,\infty}} + \epsilon) \tag{7}$$

It is not because the norm of $\| p(x) \|_{H^{q+1}} = O(1)$ that the term $| \frac{p(x)}{u_\theta + M} |_{H^{q+1}}$ is small. It depend also of the function u_θ so of the solution.We propose to develop the semi norm in the case $q = 1$ we obtain:

$$\left(\frac{p(x)}{u_\theta + M}\right)'' = \left(\frac{p(x)}{u_\theta(x) + M}\right)'' \tag{8}$$

$$= \frac{u_\theta(x) p''(x) - u_\theta''(x) p(x)}{(u_\theta(x) + M)^2} + 2 \frac{(p'(x) u_\theta(x) - u_\theta'(x) p(x))(u_\theta(x) + M)'}{(u_\theta(x) + M)^3} \tag{9}$$

$$= \frac{u(x) p''(x) - u''(x) p(x)}{(u(x) + \epsilon p(x) + M)^2} + 2 \frac{(p'(x) u(x) - u'(x) p(x))(u_\theta(x) + M)'}{(u(x) + \epsilon p(x) + M)^3} + O(\epsilon) \tag{10}$$

$$= \frac{u(x) p''(x) - u''(x) p(x)}{(u(x) + \epsilon p(x) + M)^2} + 2 \frac{(p'(x) u(x) - u'(x) p(x))(u(x) + M)'}{(u(x) + \epsilon p(x) + M)^3} + O(\epsilon) \tag{11}$$

(12)

Nicolas :

$$\left(\frac{p(x)}{u_\theta + M}\right)'' = \frac{(u_\theta(x) + M) p''(x) - u_\theta''(x) p(x)}{(u_\theta(x) + M)^2} + 2 \frac{(p'(x)(u_\theta(x) + M) - u_\theta'(x) p(x)) u_\theta'(x)}{(u_\theta(x) + M)^3}$$

$$= \frac{p''(x) + 2 p'(x) u_\theta'(x)}{u_\theta(x) + M} - \frac{u_\theta''(x) p(x)}{(u_\theta(x) + M)^2} - 2 \frac{(u_\theta'(x))^2 p(x)}{(u_\theta(x) + M)^3} - O(\epsilon)$$

Since ϵ is small, since we have assume that the norm of $p(x)$ and all derivate of $p(x)$, the norm of u are $O(1)$, and if we choose M small we obtain that

$$\| \left(\frac{p(x)}{u_\theta + M}\right)'' \|_{L^2} = O(\| u'' \|_L^2) + O(\| u' \|_L^2) \tag{13}$$

With our approach the error decrease with a factor ϵ like show in (7). However our formal computation that show the gain can be erase if the L^2 norm of the first and/or second derivate are large (given by 13). If the solution is a high frequency sinuosidal function the gain obtain plugging the prior in the

basis can quickly disappears. Now we will propose to understand the behavior of the error (7) with large M.

$$\left\| \left(\frac{p(x)}{u_\theta + M} \right)'' \right\|_{L^2} (\|M + u\|_{W^{m,\infty}}) = \left\| \left(\frac{p(x)}{u_\theta(x) + M} \right)'' \right\|_{L^2} * |M + u|_{W^{m,\infty}} \tag{14}$$

$$= \left\| \left(\frac{p(x)}{M(\frac{1}{M}u_\theta(x) + 1)} \right)'' \right\|_{L^2} M \left(\|1 + \frac{1}{M} u\|_{W^{m,\infty}} \right) \tag{15}$$

$$= \left\| \left(\frac{p(x)}{(\frac{1}{M}u_\theta(x) + 1)} \right)'' \right\|_{L^2} \left(\|1 + \frac{1}{M} u\|_{W^{m,\infty}} \right) \tag{16}$$

$$\tag{17}$$

We observe that

$$\lim_{M \to +\infty} \left\| \left(\frac{p(x)}{u_\theta + M} \right)'' \right\|_{L^2} (\|M + u\|_{W^{m,\infty}}) = \sqrt{|\Omega|} p''(x)$$

Since we have assume that $p(x)$ admit a norm close to 1 and also the same for the derivative we obtain that when M is large the gain of our method is really ϵ. So this computation seems to show that it is better to take M large and that the larger the first and second derivatives of the solution are, the greater the gain with M large should be. However, this study makes particular assumptions about the error which are not necessarily true in practice. It is therefore not possible to ensure that this conclusion is always correct. Now we will show the behavior of the interpolator compared with M. The interpolator is given by

$$\overline{I_h}(u + M) = \sum_{i=1}^{N} \left(\frac{u(x_i) + M}{u_\theta(x_i) + M} \right) \phi_i(x)(u_\theta(x) + M)$$

$$\overline{I_h}(u + M) = \sum_{i=1}^{N} \left(\frac{u_\theta(x_i) + \epsilon p(x_i) + M}{u_\theta(x_i) + M} \right) \phi_i(x)(u_\theta(x) + M)$$

$$\overline{I_h}(u + M) = \sum_{i=1}^{N} \left(1 + \epsilon \frac{p(x_i)}{u_\theta(x_i) + M} \right) \phi_i(x)(u_\theta(x) + M)$$

$$\overline{I_h}(u + M) = \left(\sum_{i=1}^{N} \phi_i(x) \right)(u_\theta(x) + M) + \epsilon \sum_{i=1}^{N} \left(\frac{p(x_i)}{u_\theta(x_i) + M} \right) \phi_i(x)(u_\theta(x) + M)$$

The partition of the unity and $\overline{I_h}(u) = \overline{I_h}(u + M) - M$ gives

$$\overline{I_h}(u) = u_\theta(x) + \epsilon \sum_{i=1}^{N} \left(\frac{p(x_i)}{u_\theta(x_i) + M} \right) \phi_i(x)(u_\theta(x) + M)$$

which gives

$$\overline{I_h}(u) = u_\theta(x) + \epsilon \sum_{i=1}^{N} \left(\frac{p(x_i)}{u_\theta(x_i) + M} \right) \phi_i(x)\left(u_\theta(x_i) + M + (x - x_i)u'_\theta(x_i) + O(h^2)\right)$$

$$\overline{I_h}(u) = u_\theta(x) + \epsilon \sum_{i=1}^{N} p(x_i)\phi_i(x) \left(1 + \frac{(x - x_i)u'_\theta(x_i)}{u_\theta(x_i) + M} + O(\frac{h^2}{M}) \right)$$

So

$$u - \overline{I_h}(u) = \epsilon \left(p(x) - \sum_{i=1}^{N} p(x_i)\phi_i(x) \left(1 + \frac{(x - x_i)u'_\theta(x_i)}{u_\theta(x_i) + M} + O(\frac{h^2}{M}) \right) \right)$$

This computation show that when M tend to infinity show that interpolate u with the modified basis is exactly equivalent to interpolate p with the classical P_k finite element. If M is small compare to the derivative of u_θ we interpolate not p but a function of p and u. This computation confirm the result the previous of convergence.

4.2 Practical details

In this subsection we will gives the technical details for the implementation of the enhanced finite element with multiplicative prior. When we use the multiplicative prior we will solve the problem (6) with M choose such that the solution $\tilde{u} \neq 0$. After to comeback to our principal problem we use that $u(x) = \tilde{u}(x) - M$. If we have a prior $u_\theta(x)$ on u the prior $\tilde{\psi}$ for u_θ is given by $\tilde{\psi} = \psi + M$.
To solve the problem (6) we discretize u_0 and v in the modified space $\bar{V}_h$ with the prior ψ. To construct the matrix and the RHS we use the same strategy as before excepted at one level. For classical P_q finite elements we have basis functions which are piecewise polynomial. To integrate the terms in the matrix and RHS we use in general classical quadrature like Gauss Lobatto or Gauss Legendre with the minimal size to exactly compute each term. In the new space $\bar{V}_h$ the basis functions are given by the product of the classical basis function and the prior ψ. Since the prior is not polynomial the classical quadrature formula could be not sufficient. In practice we use a high order quadrature formula compared to the polynomial order q. On the figure (1) we show the classical P_1 basis function and the modified P_1 basis with a sinuosidal prior. Usual for the results we use a quadrature formula of order

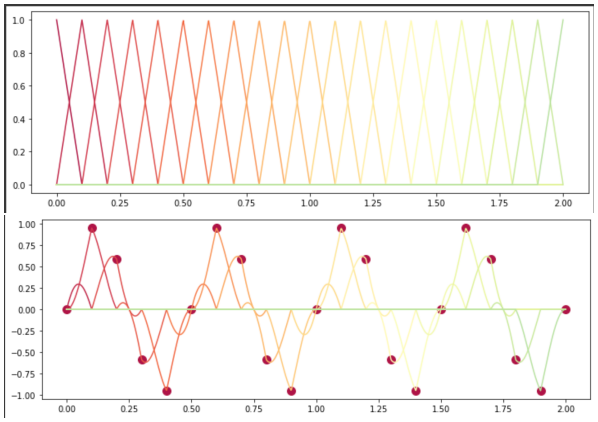

Figure 1:

8.

4.2.1 Boundary condition

The last point is to determinate how to encode the classical boundary condition, mainly the Dirichlet. We will introduce the different approaches in 1D.

For the classical approach, since the coefficient interpolate the solution, we now that α_0, the degree of freedom associated to the left value must be equal to $u(x_0)$ and α_N the degree of freedom associated to the right value must be equal to $u(x_N)$. So on the first and last line of the matrix we put $A_{ij} = 0$ for $i \neq j$ and $A_{ii} = 1$ and chose $b_0 = g(x_0)$ and $b_N = g(x_N)$.

For this approach there is two way to implement the boundary condition. The first one consist to impose in the matrix the boundary condition as in the classical approach. With the multiplicative prior, the degree of freedom associated with the finite element with multiplicative prior must be interpolate $\frac{u(x)}{u_\theta(x)}$. Consequently we must choose for the modified problem (**??**): on the first and last line of the matrix we put $A_{ij} = 0$ for $i \neq j$ and $A_{ii} = 1$ $b_0 = \frac{g(x_0)+M}{u_\theta(x_0)+M}$ and $b_N = \frac{g(x_N)+M}{u_\theta(x_N)+M}$ to impose the good boundary condition. If $u_\theta(x)$ satisfy the boundary condition we will have $b_0 = b_N = 1$ in the right hand side.
Another solution is not to impose boundary conditions in the matrix, but to impose them solely through a prior. This approach can only work for homogeneous Dirichlet. To do this, we construct the base functions associated with the edge nodes and impose no conditions on the edge parameters. Since we multiply by the prior, which is zero at the edges, we impose the right boundary conditions. This approach leaves more freedom to capture the right derivatives at the boundaries.

5 Prior construction and PINNs

In the previous section we have introduced a modified FE method. Now we will propose to construct a prior some a set of simulation using Physic informed Neural Networks (PINNs). To begin we will recall the principle of the PINNs

5.1 Physic informed neural network

The physic informed neural network were introduced in 2017 in ([]) in order to solve a PDE using neural networks. The central idea is to solve a PDE as an optimization problem. We will illustrate the method in the general case of a PDE of type:

$$\begin{cases} \partial_t \mathbf{U} = \mathcal{N}(\mathbf{U}, \partial_x \mathbf{U}, \partial_{xx} \mathbf{U}, \boldsymbol{\beta}) \\ \mathbf{U}(t,x) = g(x), \quad \forall x \in \partial\Omega \\ \mathbf{U}(0,x) = \mathbf{U}_0(x, \boldsymbol{\alpha}) \end{cases}$$

with $\boldsymbol{\mu} = (\boldsymbol{\alpha}, \boldsymbol{\beta})$ a set of parameters. The first idea of PINNs comes from the observation that by construction neural networks are regular functions that are many times derivable with respect to their weights and inputs. Therefore neural networks whose derivatives are easily evaluated by automatic differentiation form natural candidates for approximating PDEs. A PINNs is a neural network that takes as input (t,x), we will note here $\mathbf{U}_\theta(t,x)$. Once this network is defined, solving the PDE can be rewritten as the following variational problem:

$$\min_\theta J_r(\theta) + J_b(\theta) + J_i(\theta) + J_{data}(\theta) \tag{18}$$

with the residual loss

$$J_r(\theta) = \int_0^T \int_\Omega \parallel \partial_t \mathbf{U}_\theta(t,x) - \mathcal{N}(\mathbf{U}_\theta, \partial_x \mathbf{U}_\theta, \partial_{xx} \mathbf{U}_\theta, \boldsymbol{\beta})(t,x) \parallel_2^2 \, dxdt \tag{19}$$

the boundary loss

$$J_b(\theta) = \int_0^T \int_{\partial\Omega} \parallel \mathbf{U}_\theta(t,x) - g(x) \parallel_2^2 \, dxdt, \quad J_i(\theta) = \int_\Omega \parallel \mathbf{U}_\theta(0,x) - \mathbf{U}_0(x, \boldsymbol{\alpha}) \parallel_2^2 \, dx \tag{20}$$

This minimisation problem amounts to minimising the residual when we incorporate our approximate solution into the PDE with initial and boundary conditions. If we are able to calculate the previous integrals we can solve the minimisation problem (learning phase) by a gradient method (with respect to θ). The second ingredient in the PINNS method concerns the calculation of the integral. The most natural idea is to estimate the integral using a **Monte-Carlo** method. We will thus give ourselves a certain number of points called "collocation" and approach the integral with. We give ourselves for example:

$$(t_1, x_1, ...t_N, x_N)$$

Using these points we can approximate the integral with the following formulas:

$$\int_0^T \int_\Omega \parallel \partial_t \mathbf{U}_\theta(t,x) - \mathcal{N}(\mathbf{U}_\theta, \partial_x \mathbf{U}_\theta, \partial_{xx} \mathbf{U}_\theta, \beta)(t,x) \parallel_2^2 \approx \frac{1}{N} \sum_{i=1}^N \parallel \partial_t \mathbf{U}_\theta(t_i,x_i) - \mathcal{N}(\mathbf{U}_\theta, \partial_x \mathbf{U}_\theta, \partial_{xx} \mathbf{U}_\theta, \boldsymbol{\beta})(t_i,x_i) \parallel_2^2$$

The general behavior of the PINNs is detailed of the figure (2)

Contrary to the usual numerical methods the boundary condition and the initial condition are not satisfy exactly by a PINNs. It can generate problem in some cases. To avoid this in (ref) the author propose to define the solution of the PINNs as

$$\mathbf{U}_\theta(t,x) = b(x,t)\bar{\mathbf{U}}_\theta + \mathbf{U}_0(x)$$

with $\bar{\mathbf{U}}_\theta$ the neural network and $b(t,x)$ a simpler function which is equal to zero in $t=0$ and such as $\mathbf{U}_\theta$ satisfy exactly the BC. It allows to avoid the two loss on the initial condition and the boundary condition and keep only the residual discrete loss:

$$J_{res}(\theta) = \sum_{i=1}^N \parallel \partial_t \mathbf{U}_\theta(t_i,x_i) - \mathcal{N}(\mathbf{U}_\theta, \partial_x \mathbf{U}_\theta, \partial_{xx} \mathbf{U}_\theta, \boldsymbol{\beta})(t_i,x_i) \parallel_2^2$$

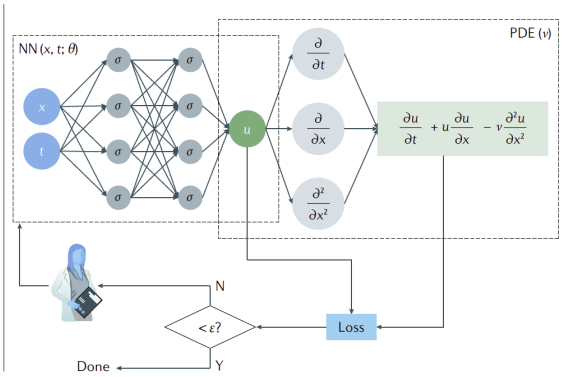

Figure 2:

Additionally we can add which could from to another solver of real data. We obtain the second loss:

$$J_{data}(\theta) = \sum_{i=1}^{N_{data}} + \left(\| \mathbf{U}_\theta(t_i, x_i) - \mathbf{U}_i^{n_i} \|_2^2 + \| \partial_x \mathbf{U}_\theta(t_i, x_i) - (\partial_x \mathbf{U})_i^{n_i} \|_2^2 \right)$$

The final loss is a combination of these two loss $J_{res}(\theta)$ and $J_{data}(\theta)$. The classical PINNs approachs dont use data on the derivative but some recents works that it could be useful.

The PINNs can be compare to a PDE solver. We gives the parameters, the BC and the initial conditions and one training gives you the solution. The two mains advantages of the PINNs is that is a mesh-less method and increasing the dimension is less costly than in the classical approach. However comparing to the usual numerical methods this method seems often slower and admit less guaranties of convergence.

5.2 PINNs par parametric PDE

As say just before the PINNs are able to treat high dimensional problem. Indeed the fully connected neural network are able to learn high dimensional function and the approximation of the integral residue on the loss is made by Monte-Carlo approach which also converge independently of the dimension.

TOO DOO

5.3 Neural operator and Nomad approach

As say just before the PINNs are able to treat high dimensional problem. Indeed the fully connected neural network are able to learn high dimensional function and the approximation of the integral residue on the loss is made by Monte-Carlo approach which also converge independently of the dimension.

TOO DOO

6 Numerical results

6.1 1D Laplacian

In this section we solve the following problem:

$$-\partial_{xx}u = \alpha\sin(2\pi x) + \beta\sin(4\pi x) + \gamma\sin(6\pi x)$$

with $\Omega = [0,1]$, Dirichlet BC. For this problem the solution is given by

$$u(x) = \pi^2(4\alpha\sin(2\pi x) + 16\beta\sin(4\pi x) + 36\gamma\sin(6\pi x))$$

For the higher frequency we expect have poor quality results with standard FE on coarse grids. Firstly we will construct two prior $u_{\theta,p}(x,\alpha,\beta,\gamma)$ with a PINNs and $u_{\theta,d}(x,\alpha,\beta,\gamma)$ just with data gives by the exact solutions. After we will compare the additive approach and the multiplicative approach for different value of the coefficient M. In the two cases the network and the training is given by:

- Size inputs: 4, size outputs: 1,
- Layers: $[20, 80, 80, 80, 20, 10]$,
- Activation functions: $\sin(x)$.
- Training domain of parameters: $\alpha \in [0,1]$, $\beta \in [0,1]$, $\gamma \in [0,1]$
- Data (for $u_{\theta,d}$): 5000 random 4-uplet (x,α,β,γ) and $u(x,\alpha,\beta,\gamma)$
- PINNs: 5000 collocation random 4-uplet (x,α,β,γ) by epoch
- learning rate: $9e-3$. Decay: 0.99

For the finite element we use P_1 element with a 8 Gauss points by cells and for the boundary condition we change the matrix and impose homogeneous boundary condition imposing that boundary degree of freedom at one.

We will begin convergence results for 4 examples given by the PINNs network:

- case 1: $(\alpha,\beta,\gamma) = (0.3, 0.2, 0.1)$
- case 2: $(\alpha,\beta,\gamma) = (0.4, 0.6, 0.3)$
- case 3: $(\alpha,\beta,\gamma) = (0.7, 0.4, 0.6)$
- case 4: $(\alpha,\beta,\gamma) = (0.8, 0.5, 0.8)$

and after we will statistical results of gain compared to the classical approach on very grids.

Case 1	Classic		Add prior			Mul prior M=3			Mul prior M=100		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$3e^{-2}$	–	$4.4e^{-4}$	–	68	$3.9e^{-4}$	–	76	$4.4e^{-4}$	–	68
20	$7.8e^{-3}$	1.94	$1.21e^{-4}$	1.87	64	$1.07e^{-4}$	1.88	73	$1.e^{-4}$	1.86	64
40	$1.97e^{-3}$	1.98	$3.2e^{-5}$	1.92	61	$2.89e^{-5}$	1.88	68	$3.18e^{-5}$	1.92	62
80	$4.94e^{-4}$	2.00	$8.17e^{-6}$	1.97	60	$7.44e^{-6}$	1.96	66	$8.12e^{-6}$	1.97	61
160	$1.24e^{-4}$	2.00	$2.05e^{-6}$	1.99	60	$1.87e^{-6}$	1.99	65	$2.04e^{\|6}$	1.99	60

The results on the fourth test cases shows firstly that all the enhanced FE allows to increase the accuracy of the method on all the test cases and these modified Finite element converges also as the classical approach. The theoretical analysis show that multiplicative approach can be less efficient if the $k+1$ derivative is large for small M and admit the same error that the additive approach for $M \to \infty$.**Nicolas :** if the derivative is large ? It is confirmed by the results. For the case 1 where the second derivative is not so big, the multiplicative approach with M small is a little bit better and for the other case where the second derivative is large the additive approach is more accurate for M small.

Case 2	Classic		Add prior			Mul prior M=3			Mul prior M=100		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$8.74e^{-2}$	–	$2.82e^{-4}$	–	309	$1.22e^{-3}$	–	71	$2.91e^{-4}$	–	300
20	$2.28e^{-2}$	1.94	$7.34e^{-5}$	1.84	310	$2.98e^{-4}$	2.04	76	$7.44e^{-5}$	1.97	306
40	$5.77e^{-3}$	1.98	$1.9e^{-5}$	1.95	303	$7.77e^{-5}$	1.94	74	$1.92e^{-5}$	1.95	299
80	$1.45e^{-3}$	2.00	$4.8e^{-6}$	1.98	301	$1.96e^{-5}$	1.98	73	$4.86e^{-6}$	1.99	297
160	$3.62e^{-4}$	2.00	$1.2e^{-6}$	2.00	300	$4.93e^{-6}$	2.00	73	$1.22e^{-6}$	2.0	297

Case 3	Classic		Add prior			Mul prior M=3			Mul prior M=100		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$1.34e^{-1}$	–	$3.05e^{-4}$	–	410	$6.67e^{-4}$	–	201	$3.01e^{-4}$	–	446
20	$3.55e^{-2}$	1.92	$9.24e^{-5}$	1.72	384	$2.59e^{-4}$	1.36	137	$9.19e^{-5}$	1.71	386
40	$9.0e^{-3}$	1.98	$2.37e^{-5}$	1.96	380	$6.76e^{-5}$	1.94	133	$2.36e^{-5}$	1.96	382
80	$2.26e^{-3}$	2.00	$5.98e^{-6}$	1.99	378	$1.72e^{-5}$	1.97	131	$5.95e^{-6}$	1.99	379
160	$5.65e^{-4}$	2.00	$1.5e^{-6}$	2.00	377	$4.32e^{-6}$	1.99	131	$1.49e^{-6}$	2.0	379

Case 4	Classic		Add prior			Mul prior M=3			Mul prior M=100		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$1.78e^{-1}$	–	$5.96e^{-4}$	–	298	$2.76e^{-3}$	–	64	$6.00e^{-4}$	–	297
20	$4.70e^{-2}$	1.92	$1.57e^{-4}$	1.93	300	$1.25e^{-3}$	1.14	38	$1.57e^{-4}$	1.93	299
40	$1.19e^{-2}$	1.98	$4.01e^{-5}$	1.96	297	$3.56e^{-4}$	1.81	33	$4.03e^{-5}$	1.97	296
80	$2.99e^{-3}$	2.00	$1.01e^{-5}$	1.98	295	$9.25e^{-5}$	1.64	32	$1.02e^{-5}$	1.99	294
160	$7.48e^{-4}$	2.00	$2.54e^{-6}$	2.00	294	$2.34e^{-5}$	1.99	32	$2.55e^{-6}$	2.0	293

As expected also the multiplicative approach gives the same results that the additive one for M large. In all the case the gain given by the modified FE code is large ans as expected also the gain is larger for high-frequency case. Now we will compare statically the different approach for the PINNs and for the neural network train with data on coarse grids. For that we make 100 simulations with random parameters in the training domain (but not the training values) and compute the average gain and the variance associated.

The result on (6.1) shows that the average gain giving by the Enhanced FE with PINNs prior is very important mainly with the "add prior version". We obtain the same accuracy in P_1 with 100 cells where with use the additive prior than the classical FE with 160 cells. This results confirm also the behavior of the multiplicative prior method for varying M. The same method with a neural network train with data and not loss functions don't gives good results. To explain that we will compare the solution, the derivative and the second derivative between the exact solution and the solution obtained by the network.

The results on the figure (4) - (3) shows that the two approaches: PINNs and neural networks on the data gives a very good approximation on the solution. The main difference is on the derivatives and mainly the second one which is realy learned better by the PINNs. Since the error of the Enhanced FE with prior is mainly related to the fact that the $k+1$ derivative of the network is close the $k+1$ derivative of the solution (with k the polynomial order) it explains why the modified FE with classical dont gives good results. Here the PINNs admit two advantages: they don't need data and gives better results for one default which is that the training is longer. If you have data on first and second derivative you can use it to improve the classical network.

6.2 1D general elliptic system and convection dominate regime

In this section we solve the following problem:

$$v\partial_x u - \frac{1}{P_e}\partial_{xx} u = r$$

method :	average gain	variance gain
Add prior with PINNs	273	13000
mul prior $M = 3$ with PINNs	92	4000
mul prior $M = 100$ with PINNs	272	13000
Add prior with NN	15	18
mul prior $M = 3$ with NN	11	17.5
mul prior $M = 100$ with NN	15	18

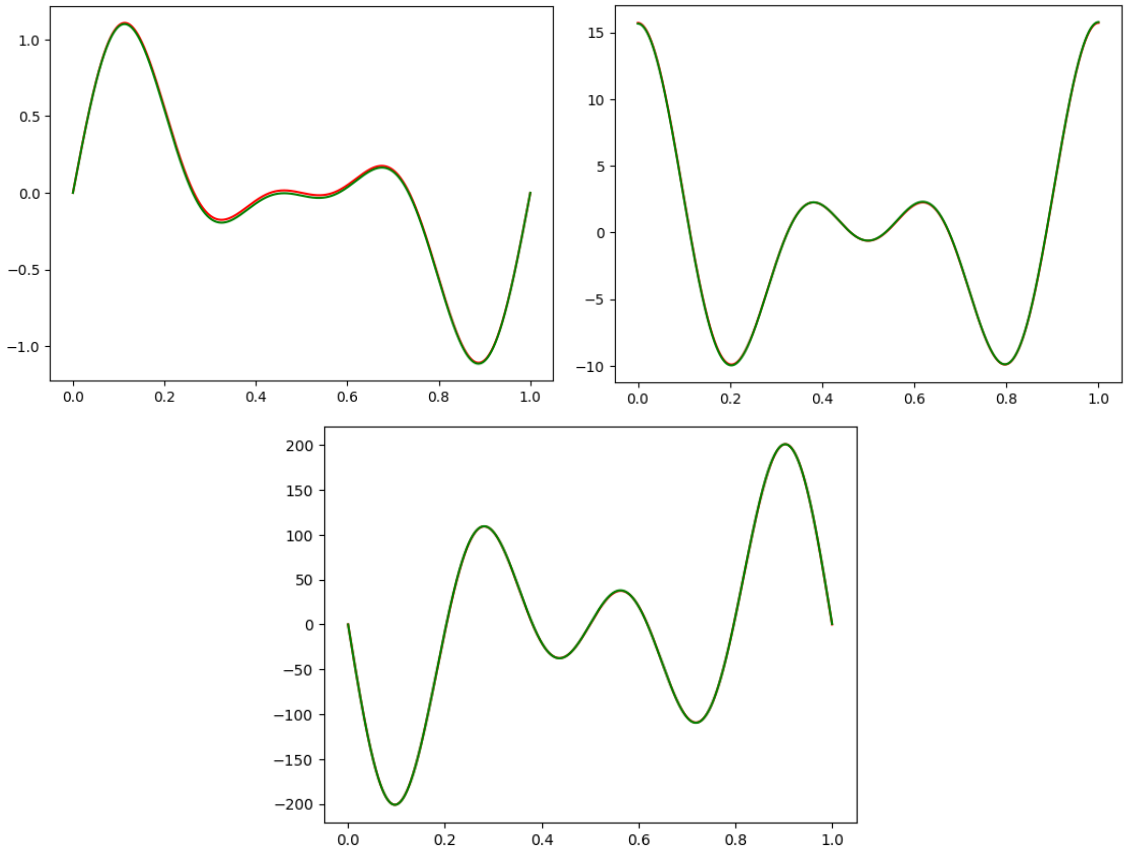

Figure 3:

with $\Omega = [0, 1]$, Dirichlet BC, v the velocity, r the reaction constant term. For this problem the solution is given by

$$u(x) = \frac{r}{1}(x - \frac{e^{P_e x} - 1}{e^{P_e} - 1})$$

The Péclet number describe the ratio between convective and diffusion term. In large Péclet regime the classical finite element method generate oscillations. In the two cases the network and the training is given by:

- Size inputs: 1, size outputs: 1, **Nicolas :** Size inputs: 3?
- Layers: $[40, 40, 40, 40, 40]$,
- Activation functions: $\tanh(x)$.
- Training domain of parameters: $r \in [1, 2]$, $Pe \in [10, 100]$
- PINNs: 5000 collocation random 4-uplet (x, r, Pe) by epoch **Nicolas :** 3-uplet ?
- learning rate: $1e - 3$. Decay: 0.99

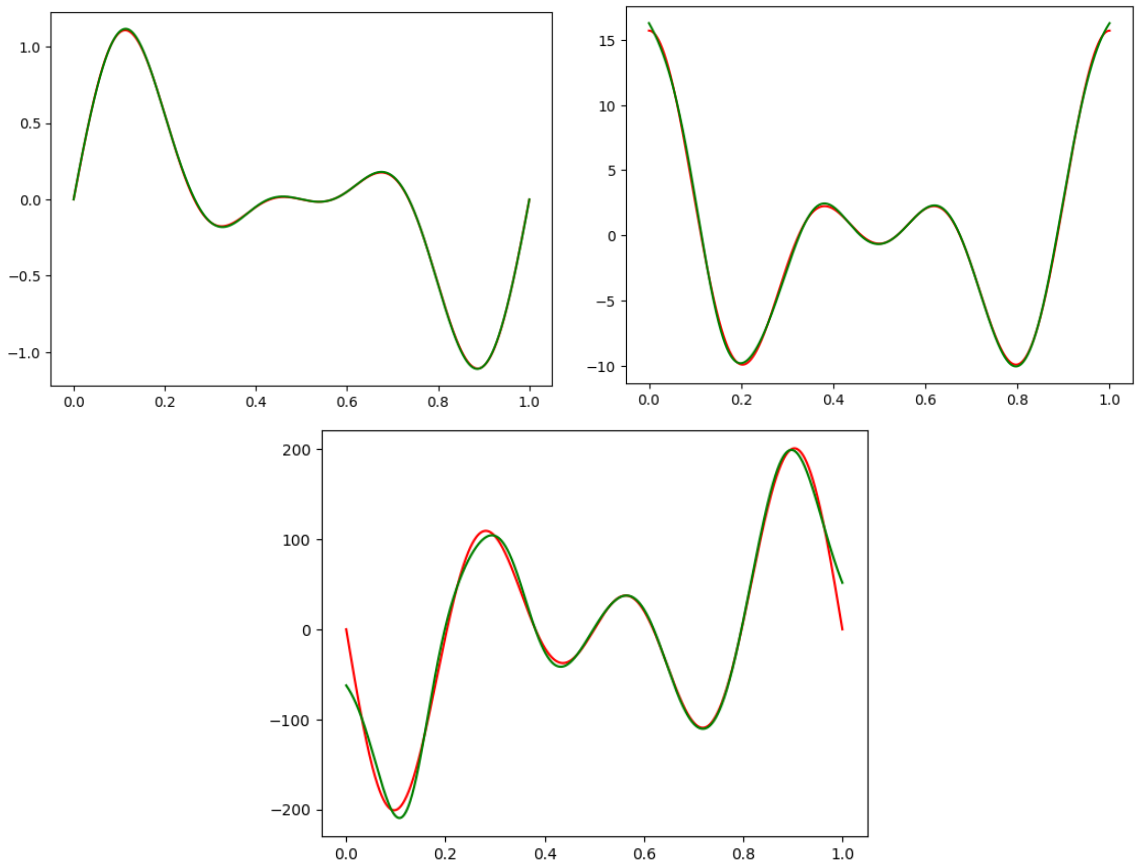

Figure 4:

For the finite element we use P_1 element with a 8 Gauss points by cells and for the boundary condition it is homogeneous Boundary condition. We will test the two implementation: when we force the degree of freedom at the boundary are equation to one (*strong*) and when we keep the basis on the boundary and don't modify the basis (*weak*).
Nicolas : je comprends pas fort et faible We will begin convergence results for 2 examples given by the PINNs network:

- case 1: $(r, Pe) = (1.2, 40)$
- case 2: $(r, Pe) = (1.5, 90)$

In this case the solution is positive so we can keep $M = 0$.

Case 1	Classic		Add prior			Mul prior BC strong			Mul prior BC weak		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$1.07e^{-1}$	–	$2.70e^{-3}$	–	40	$3.21e^{-3}$	–	33	$2.29e^{-4}$	–	467
20	$3.36e^{-2}$	1.97	$8.00e^{-4}$	1.76	42	$1.57e^{-3}$	1.03	21	$9.06e^{-5}$	1.93	371
40	$9.09e^{-3}$	1.89	$2.01e^{-4}$	2.00	45	$7.93e^{-4}$	0.98	11	$2.63e^{-5}$	1.97	345
80	$2.32e^{-3}$	1.97	$5.01e^{-5}$	1.99	46	$4.07e^{-4}$	0.96	6	$6.37e^{-6}$	1.99	365
160	$5.82e^{-4}$	1.99	$1.30e^{-6}$	1.97	45	$2.08e^{-4}$	0.97	3	$1.77e^{-6}$	2.0	289

The results are given by a PINNs without data. The results given by additive prior method are correct for low Péclet number but not good for large one. It comes from to the fact that the PINNs approximation is not close to the solution. It is also the case for the multiplicative prior method with strong BC where the results are never good. However for the multiplicative method with weak BC

Case 2	Classic		Add prior			Mul prior BC strong			Mul prior BC weak		
	error	order	error	order	gain	error	order	gain	error	order	gain
10	$2.65e^{-1}$	–	$1.51e^{-1}$	–	1.7	$1.47e^{-1}$	–	1.8	$9.33e^{-4}$	–	284
20	$1.06e^{-1}$	1.32	$6.04e^{-2}$	1.33	1.7	$6.61e^{-1}$	1.15	1.6	$3.84e^{-4}$	1.28	276
40	$3.46e^{-2}$	1.62	$1.96e^{-2}$	1.62	1.8	$3.13e^{-2}$	1.08	1.1	$1.13e^{-4}$	1.76	305
80	$9.50e^{-3}$	1.86	$5.32e^{-3}$	1.87	1.8	$1.62e^{-2}$	0.95	0.6	$3.26e^{-5}$	1.80	291
160	$2.43e^{-3}$	1.86	$2.43e^{-3}$	1.86	1.8	$8.5e^{-3}$	0.93	0.3	$8.67e^{-6}$	1.91	280

the results are very good since the gin factor is around 300/400 on the two tests. These results can be explain by which type of approximation the PiNNs gives.

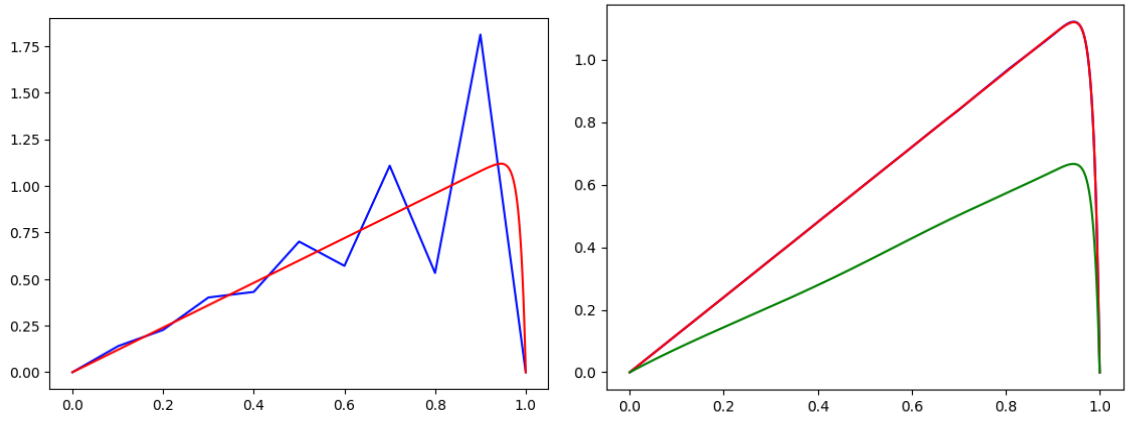

Figure 5:

In the left of 5 we obtain then solution in red when the classical FE solution with 10 cells in blue which oscillate a lot. It is a classical phenomena for convective problem. In the right we have the solution in red, the numerical solution given by FE with multiplicative prior and weak BC which is very good and in green we have the network solution. The network solution is far to the real one which explain the poor results of the additive prior method. However we can see that the solution given by the PINNs is good with a multiplicative constant. It is why the multiplicative method work well and gives impressive results for this case. Now we need to explain the difference between BC implementation. If the aprioi and the solution are close to a constant r pret then the error is given by $\frac{u''}{u_\theta''} \approx (r)'' = 0$. However, the strong method for imposing the BCs will mean that our prior will be close at constant r to the solution everywhere except at the edge where the constant will be 1 and so the $\frac{u''}{u_\theta''}$ will become large. The multiplicative method therefore recovers errors with a multiplicative ready constant only with the so-called weak BCs.)

6.3 2D .....

6.3.1 2D parametric case

6.3.2 2D high frequency case

6.3.3 High orde projection

todo: use H_1 et H_2 projection

References