Rapport de TP-Eléments finis

Résolution d'une équation d'advection-diffusion avec FreeFem++

21 septembre 2020

CROGUENNEC Guillaume
DAUPHIN Mathias
DUPONT Ronan

LAURENT Arthur
MOTTIER Romain
PLANQUE Baptiste
(Chef de projet)

Table des matières

1 Introduction

Durant ce rapport, nous nous intéresserons à un problème physique que nous allons discrétiser afin de pouvoir effectuer une approximation numérique. Cette approximation numérique sera faite par le biais de la méthode des éléments finis. Puis nous allons modéliser ce problème numérique à l'aide de la méthode des éléments finis, afin d'obtenir une visualisation temporelle en 3 dimensions d'une solution approchée.

Nous allons commencer par trouver la formulation variationnelle de l'équation (1) présentée dans la partie suivante. Pour ce faire, nous utiliserons le cours [3]. Il s'agira ensuite de l'implémenter (en langage C) afin de résoudre le problème en utilisant le logiciel FreeFem++.

Pour vérifier que nous obtenons des résultats satisfaisants, nous introduisons un cas-test. En effet, en se donnant une solution φ connue, on peut calculer le second membre f associé. Cela nous permettra de valider la méthode numérique par un calcul d'erreur en norme L^2.

Ensuite, nous déterminerons l'ordre de convergence de la simulation numérique, correspondant à la pente de la courbe $ln(|\varphi_{exacte} - \phi_{calculé}|) = g(ln(h))$ où h est le pas spatial. Nous allons répéter cette opération en faisant varier le type de maillage (P1 ou P2) et l'utilisation ou non de la fonction *adaptmesh* de FreeFem. Cette dernière est un outil très puissant car elle permet d'affiner automatiquement le maillage aux endroits où la solution varie localement très fortement ou, à l'inverse, elle rend le maillage plus grossier aux endroits où la solution varie peu, où il n'est pas nécessaire qu'il y ait beaucoup d'éléments. C'est donc une fonction qui amène une solution satisfaisante, à moindre coût. Il est d'ailleurs possible de spécifier l'erreur permise lors de son utilisation et elle se charge de créer le nouveau maillage selon la précision définie.

Enfin, nous allons résoudre un problème d'advection diffusion. Un récipient carré de petite profondeur contient un fluide en mouvement dans lequel, à l'instant initial, on verse du sel à une certaine concentration. Le but va être de déterminer la concentration en tout point du récipient.

2 Introduction aux EDP, contexte physique

2.1 Équation d'advection diffusion

Dans le modèle physique étudié, le mode de propagation qu'est le rayonnement est négligé, car il se fait à l'aide d'une onde électro-magnétique. Donc le sel se propage seulement par diffusion et par advection. Il se propage par diffusion car ce mode de propagation ne nécessite aucun déplacement de matière, et par advection car le fluide dans le récipient induit un terme de propagation pour le sel. Ceci nous donne l'équation suivante :

$$\frac{\partial u}{\partial t} + c\nabla u - \varepsilon \Delta u = f \quad \text{avec} \quad x \in \Omega \text{ et } t > 0 \tag{1}$$

et ε un facteur de viscosité pris égal à 1 que l'on peut faire varier au moment de la simulation numérique si l'on souhaite modifier la contribution de la diffusion. En effet, ce terme est dû à la diffusion, tandis que les deux autres du même côté de l'égalité sont dûs à l'advection.

Ces deux termes peuvent être écrits en un seul terme car :

$$\frac{Du}{Dt} = \frac{\partial u}{\partial t} + c\nabla u \tag{2}$$

On a alors :

$$\frac{Du}{Dt} - \varepsilon \Delta u = f \quad \text{avec} \quad x \in \Omega \text{ et } t > 0 \tag{3}$$

Avec $u(0, x) = u_0(x)$ la condition initiale.

2.2 Formulation variationnelle du problème

Déterminons la formulation variationnelle de ce problème. Pour cela nous noterons v une fonction test. En multipliant (3) par v, on obtient :

$$\frac{Du}{Dt} v - \Delta u.v = fv$$

Puis en intégrant sur le domaine Ω on a :

$$\int_\Omega \frac{Du}{D_t} v \; dx - \int_\Omega \Delta u.v \; dx = \int_\Omega fv \; dx$$

En utilisant la formule de Green, on obtient :

$$\boxed{\int_\Omega \frac{Du}{D_t} v \; dx + \int_\Omega \nabla u.\nabla v \; dx - \int_{\partial\Omega} (\nabla u, \vec{n}) \, v \; d\sigma = \int_\Omega fv \; dx}$$

On n'impose que v soit dans $H_0^1 = \{v \in H^1 \mid \gamma_0(v) = 0\}$ où γ_0 désigne l'application trace obtenue en prolongeant par continuité l'application $v|_{\partial\Omega}$. Autrement dit, on impose v nulle sur les bords.

Ainsi, on a la formulation variationnelle suivante :

$$\text{"Trouver une solution } u \in H_0^1 \text{ telle que } a(u,v) = L(v),\ \forall v \in H_0^1\text{"} \tag{4}$$

avec :

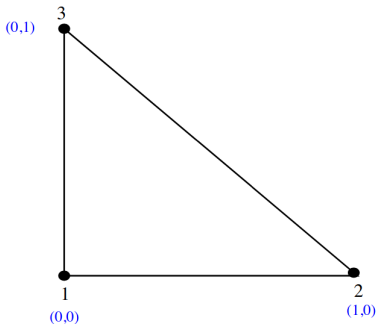
$$\begin{cases} a(u,v) = \int_\Omega \frac{Du}{Dt} v\ dx + \int_\Omega \Delta u \Delta v\ dx \\ L(v) = \int_\Omega fv\ dx \end{cases} \tag{5}$$

2.3 P1, P2

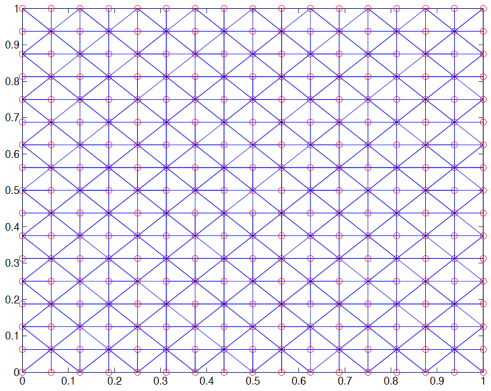
P1 et P2 sont des approximations utilisant des triangles comme éléments de référence afin de discrétiser une surface en 2D. Ceux-là sont bien plus adaptés que des rectangles ou carrés car ils permettent d'obtenir un maillage bien plus précis de par sa forme. On pourra s'aider du cours [2] afin de comprendre ces approximations. Ce cours étant très bien illustré, il nous a permis d'observer les différents maillages en fonction des différents éléments de références.

2.3.1 P1 Lagrange

Le maillage P1 a pour élément de référence le triangle ci-dessous comprenant les 3 noeuds suivant :

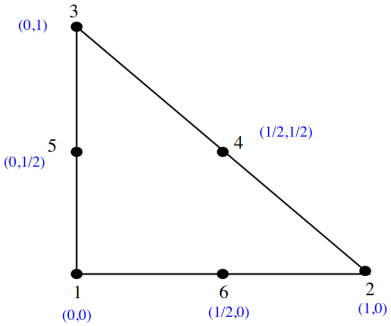


Le maillage d'une grille sur $[0,1]^2$ sera donc représenté par la figure ci-dessous. Celui-ci est discrétisé en 1024 triangles.

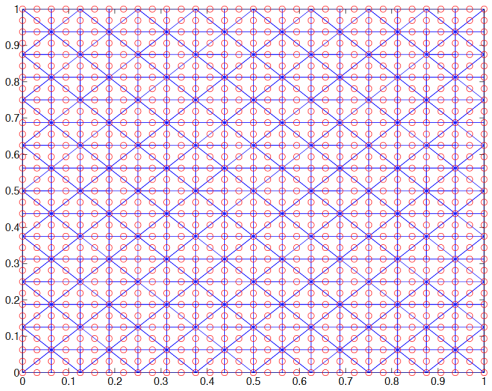


2.3.2 P2

Pour un maillage en P2, nous avons un maillage fait avec pour élément de référence le triangle ci-dessous. Il a 6 noeuds.



Le maillage d'une grille sur $[0,1]^2$ sera donc représenté par la figure ci-dessous. Celui-ci est de même discrétisé en 1024 triangles.



Celui-ci est bien plus précis que le précèdent car il a beaucoup plus de noeuds.

2.4 Mailage adaptatif

Lorsque le comportement local de la solution à calculer varie de manière conséquente sur le domaine considéré il est nécessaire d'utiliser une méthode par maillage adaptatif.

En effet ce type de maillage adaptera la taille des éléments en fonction du comportement de la solution. Si son gradient est élevé alors elle subit de fortes variations et il est nécessaire d'utiliser un maillage constitué d'éléments de plus petites tailles, à l'inverse si la solution subit de faibles variations on pourra garder le maillage régulier pour le calcul de ces zones.

Un maillage adaptatif fonctionne de la manière suivante ; un premier calcul est réalisé sur un maillage régulier, une fois le gradient appliqué à la solution calculée on obtient une carte des zones de maillage à raffiner. Une fois cette carte obtenue on peut alors lancer un deuxième calcul en tenant compte des zones à fort et faible gradient. Ce genre de maillage contribue à réduire les erreurs de calcul tout en améliorant la précision du schéma.

Dans ce TP on utilisera des maillages adaptatifs de base P1 et P2.

3 Modélisation numérique

3.1 Méthode de résolution

Nous reprenons la formulation variationnelle de la partie 2.2. En utilisant la méthode des caractéristiques pour l'équation de transport (voir *Méthode des caractéristiques de Galerkin dans le cas d'une équation de transport*) ainsi que le schéma implicite on obtient l'équation suivante :

$$\int_\Omega \frac{u^{n+1} - u \circ X^n}{\delta t} v + \Delta u^{n+1} \Delta v \; dx = \int_\Omega f v \; dx \tag{6}$$

Avec :

$$\frac{dX}{dt} = c$$

Sous cette forme, l'équation (6) pourra être implémentée sur Freefem++ grâce à la commande *convect* qui permet notamment de calculer $u \circ X^n$.

3.2 Cas test

3.2.1 Cas test étudié

Dans un premier temps, nous nous sommes intéréssé à la résolution d'un cas test connu afin de vérifier la validité de notre méthode de résolution.

(peut être à enlever : Afin de calculer l'erreur, nous devons calculer la valeur de la solution exacte quelque soit t, x, y.

Nous avons choisis arbitrairement la Gaussienne suivante :

$$u = e^{-a(x^2+y^2+t)} \tag{7}$$

On applique cette fonction à l'équation (1) :

$$\partial_t u + c\nabla u - \varepsilon \Delta u = f \quad \text{avec} \quad c = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$$

Par dérivation, on trouve le second membre suivant :

$$f = -ae^{-a(x^2+y^2+t)} - 2c_1 axe^{-a(x^2+y^2+t)} - 2c_2 aye^{-a(x^2+y^2+t)} - 4\epsilon a(ax^2+ay^2-1)e^{-a(x^2+y^2+t)} \tag{8}$$

Ce second membre nous permettra par la suite l'aide de FreeFem++ d'approcher la gaussienne u choisie (voir *Code FreeFem*).

3.2.2 Résultats obtenus

Une fois le second membre f implémenté dans FreeFem++ nous avons pu déterminer une approximation de notre gaussienne u et déterminer l'erreur commise lors du calcul car nous connaissons la solution exacte.

On obtient ainsi les répartitions spatiales suivantes à l'instant initial à gauche et à l'instant final à droite :

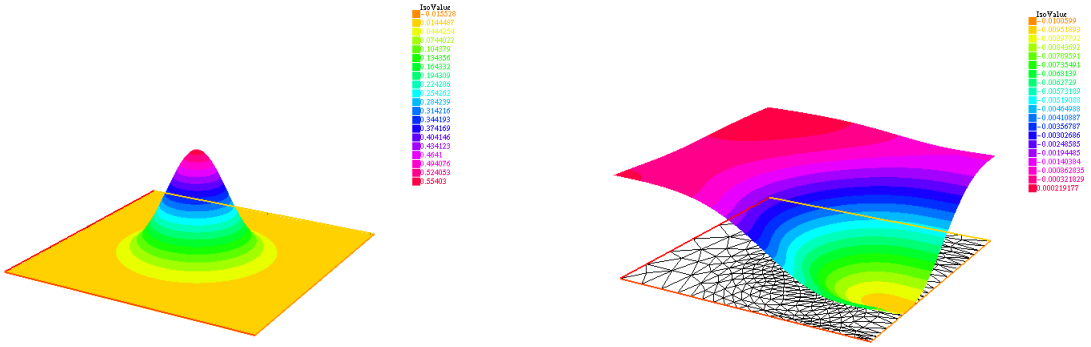


FIGURE 1 – Résultats à $t = 0$ s et $t = 1$ s

Les résultats ont été tracés pour le paramétrage suivant :

$$\begin{cases} \varepsilon = 1 \\ a = 10 \\ dt = 0.01 \\ T_f = 1 \end{cases}$$

On obtient alors :

Pour P1 :

- Un temps de compilation de 0.004459s
- Un temps d'exécution de 24.7857s

Pour P2 :

- Un temps de compilation de 0.007098s,
- Un temps d'exécution de 52.8193s,

Pour P1 adaptative :

- Un temps de compilation de 0.00904s,
- Un temps d'exécution de 18.8511s,

Pour P2 adaptative :

- Un temps de compilation de 0.006053s,
- Un temps d'exécution de 34.348s,

On observe donc conformément à ce que nous avons dit plus haut que le temps d'exécution pour les maillages adaptatifs est au moins deux fois supérieurs à celui des maillages réguliers.

3.2.3 Ordre de convergence

Afin de valider la méthode de calcul et de vérifier que les résultats obtenus sont corrects, nous avons calculé la solution d'un problème dont nous avons déjà la solution. En effet à l'aide l'équation 7, on peut calculer l'erreur produite par la méthode. Le calcul de cette erreur permet ensuite de trouver l'ordre de convergence p (équation 3.2.3). Cet ordre de convergence nous permet de valider le modèle et de voir avec quelle méthode la solution converge le mieux.

$$u_{exacte} = u_{calculée} + \mathcal{O}(h^p) \;\Rightarrow\; |u_{exacte} - u_{calculée}| = \mathcal{O}(h^p) \tag{9}$$

On peut ensuite appliquer le logarithme à l'équation précédente. Cela nous permet de nous ramener à une équation linéaire par rapport au pas h avec une pente p.

$$ln(|u_{exacte} - u_{calculée}|) = p.ln(h) + A$$

Ainsi en calculant plusieurs erreurs pour des finesses de maillage différentes on peut obtenir l'ordre de convergence.

Ici on peut prendre le nombre de triangles dans le domaine pour traduire la finesse du maillage. Et nous avons calculé l'erreur L^2 à l'aide de la formule (10).

$$error = \int_\Omega |u_{exacte} - u_{calculée}|^2 \tag{10}$$

La syntaxe du programme a été trouvée grâce à la documentation de FreeFem++ [1]. Nous avons donc tracé l'erreur pour des nombres de triangles différents. Le nombre de triangles est resté faible car si on diminue la taille des mailles l'erreur devient trop petite pour percevoir de réelles variations comme on peut le voir sur la figure 2. Dans ce cas la régression linéaire n'était pas intéressante, car le coefficient de détermination valait 0.83.

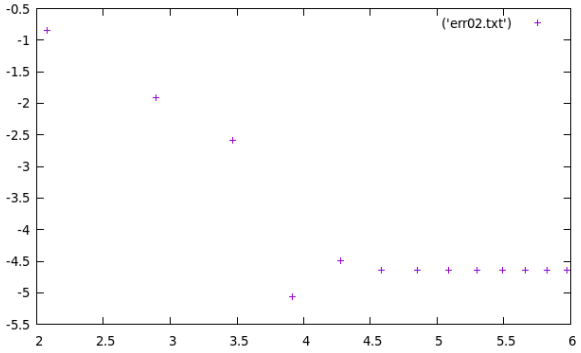


FIGURE 2 – Logarithme de l'erreur avec un maillage trop fin

Il faut ensuite appliquer une régression linéaire pour obtenir les pentes des courbes. Les coefficients de pentes sont accompagnés des coefficients de corrélation (notés R) qui attestent de la justesse de la régression linéaire. Avec un coefficient compris entre 0.95 et 1 on considère la régression acceptable. En guise d'exemple, voici en figure 3 la régression linéaire effectuée lorsqu'on utilise un maillage P1 sans la fonction *adaptmesh*.

Les ordres de convergence en distinguant le maillage P1 du maillage P2 sont répertoriés dans le tableau ci-dessous :

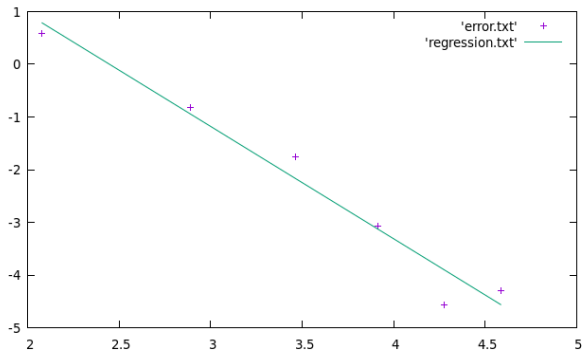


FIGURE 3 – Régression linéaire pour le cas P1, sans adaptmesh

Type de maillage	avec *adaptmesh*	sans *adaptmesh*
P1	$p = 5.556$ $R = 0.951$	$p = 2.133$ $R = 0.982$
P2	$p = 6.286$ $R = 0.967$	$p = 1.726$ $R = 0.934$

On remarque tout d'abord que l'ordre de convergence est plus élevé avec l'utilisation de *adaptmesh*. Cela signifie que pour un même nombre de triangles dans la maille, l'erreur sera plus faible avec l'adaptation de maillage que sans adaptation. La fonction *adaptmesh* améliore la précision du calcul en optimisant l'utilisation des triangles.

De plus, on pouvait s'attendre à ce que le maillage P2 soit plus précis que le maillage P1 donc que l'ordre de convergence du premier soit supérieur au second. C'est bien le cas lorsqu'on utilise *adaptmesh* ($p = 6.286 > 5.556$). Pour ce qui est de l'ordre de convergence du maillage P2 sans *adaptmesh*, il doit être erroné au vu de la faible valeur du coefficient de corrélation.

Ainsi, il est préférable d'utiliser un maillage P2 avec *adaptmesh* pour une précision accrue et un temps d'exécution de l'algorithme minimal.

3.3 Résolution d'un problème d'advection-diffusion

Considérons un récipient délimité par le domaine $\Omega = [-1, 1] \times [-1, 1]$, rempli d'un fluide en mouvement. A $t = 0$, on ajoute du sel à la concentration $u(t = 0, x) = u_0(x)$. $\mathbf{x}$ désigne le point $(x, y) \in \Omega$.

Comme on suppose l'écoulement incompressible, le problème considéré est donc la conservation de la masse du sel.

$$\begin{cases} \frac{\partial u}{\partial t} + c\nabla u - \varepsilon\Delta u = 0 \quad \text{sur} \quad]0, T[\times\Omega \\ u = 0 \quad \text{sur} \quad [0, T] \times \partial\Omega \\ u(0, x) = u_0(x) \end{cases} \tag{11}$$

Pour ce qui est du terme initial de la concentration de sel, on peut choisir de prendre une gaussienne : $u_0(x) = e^{-(x^2+y^2)}$. La célérité sera prise égale à $c = \begin{pmatrix} sin(t) \\ cos(t) \end{pmatrix}$

En ce qui concerne la partie programmation, l'algorithme est toujours le même. Ce qui change est l'ajout d'une condition de Dirichlet sur les bords ($u = 0$ sur les 4 bords de notre domaine Ω). Le terme de bord de la formulation variationnelle $-\int_{\partial\Omega} (\nabla u, \vec{n})\, v \; d\sigma$ ne peut plus être calculé directement puisqu'on ne connaît pas d'avance la solution.

Pour le bord 1, on aura $\nabla u.\overrightarrow{n_1} = \begin{pmatrix} \frac{\partial u}{\partial x}(x,y) \\ \frac{\partial u}{\partial y}(x,y) \end{pmatrix} . \begin{pmatrix} 0 \\ -1 \end{pmatrix} = -\frac{\partial u}{\partial y}(x,y)$, et on procède de la même manière pour les autres bords, avec $\overrightarrow{n_2} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \overrightarrow{n_3} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ et $\overrightarrow{n_4} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$.

Résultats de la simulation

On effectue une simulation en prenant pour temps final $T = 1$s et un pas de temps $\Delta t = 0.01$ s. Pour une meilleure précision, on utilise un maillage P2 avec la fonction *adaptmesh*.

Nous commençons par une simulation où le facteur de diffusion $\varepsilon = 1$. Les résultats sont donnés en figure 4.

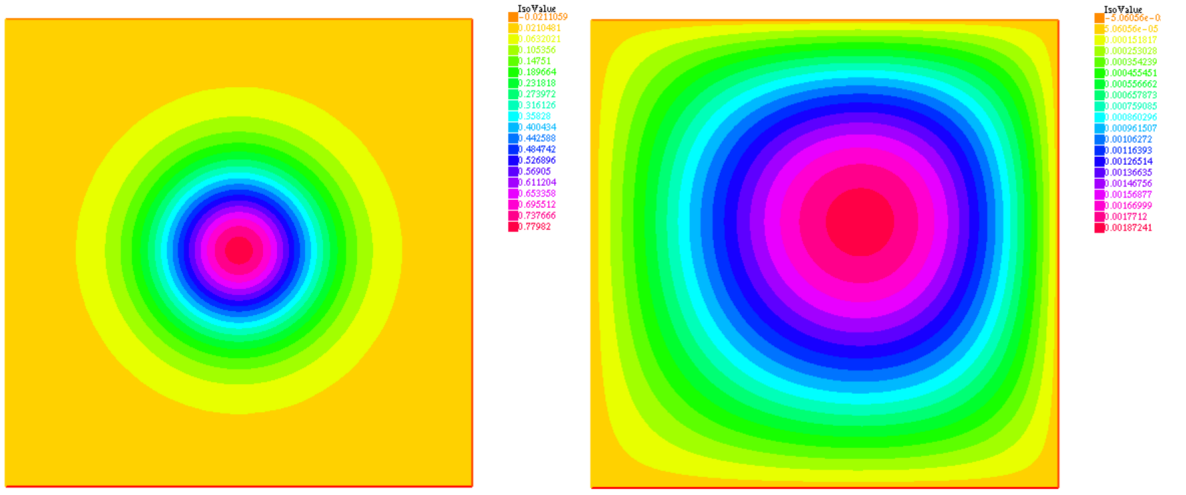


FIGURE 4 – Résultats à $t = 0$ s et $T = 1$ s pour $\varepsilon = 1$

On aperçoit bien ici le phénomène de diffusion qui fait tendre la concentration du sel vers une très petite valeur (quasiment 0) car il est diffusé uniformément dans toutes les directions. Le graphe au temps final est donc à peu près symétrique.

Maintenant réalisons la même simulation mais en prenant un coefficent de diffusion plus faible avec $\varepsilon = 0.1$ (figure 5).

Cette fois-ci la diffusion n'est pas très efficace, la partie supérieure du domaine ayant une concentration bien supérieure en sel que la partie basse. Le mouvement du fluide transporte la matière plus efficacement qu'elle ne se diffuse.

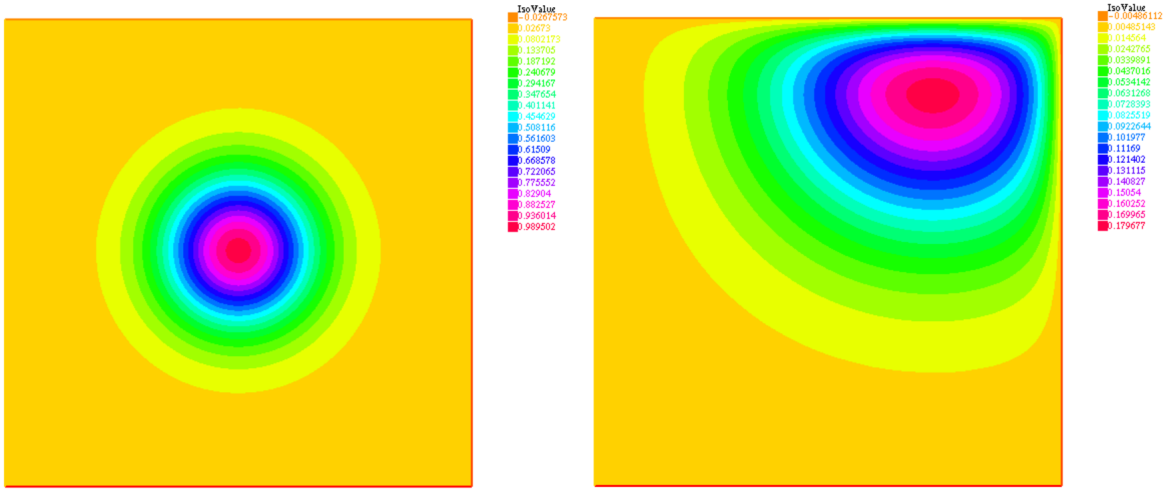


FIGURE 5 – Résultats à $t = 0$ s et $T = 1$ s pour $\varepsilon = 0.1$

4 Conclusion

Dans ce projet, nous avons pu apprendre à utiliser $FreeFem++$ dans le cadre de la résolution d'une EDP d'advection diffusion. Nous avons pu comparer les différents modèles permettant la résolution de cette EDP : la maillage en P1 et P2 ainsi que les adaptatifs.

Nous avons donc comparé les erreurs en fonction des différents types de modèles. Sans surprise, l'erreur la plus faible apparaît avec la maillage en **P2 adaptatif**.

Le maillage le plus discrétisé est celui en P2 et la fonction *adaptmesh* permet de le rendre plus optimisé en augmentant le nombre de triangles là où il y a de fortes variations de la solution.

5 Bibliographie

Références

[1] Frederic HECHT, *FreeFEM Documentation*, 2020,
https ://doc.freefem.org/introduction/index.html

[2] François ALOUGES, *Eléments finis en dimension N >=2*, 2014,
http ://www.cmap.polytechnique.fr/ alouges/cours9.pdf

[3] Mehmet ERSOY, *Eléments Finis : applications (FreeFem++)*, 2020,
http ://ersoy.univ-tln.fr/documents/teaching/CM_freefem.pdf

6 Annexes

6.1 Méthode des caractéristiques de Galerkin dans le cas d'une équation de transport

Considérons le problème de transport suivant :

$$\begin{cases} \partial_t u(t,x) + c\partial_x u(t,x) = 0 \quad \forall x \in \mathbb{R},\ \forall t > 0 \\ u(0,x) = u_0(x) \end{cases}$$

avec $c > 0$.

Trouvons les courbes caractéristiques telles que :

$$\begin{cases} X'(s) = c, \quad 0 \leqslant s < t \\ X(t) = x \end{cases}$$

On a donc :

$$\frac{d}{ds}u(s,X(s)) = \frac{\partial u}{\partial t}(s,X(s)) + X'(s)\frac{\partial u}{\partial x}(s,X(s)) = 0$$

$$u(t,X(t)) = u(t,x) = u(s,X(s))$$

Or cette dernière égalité est vraie pour tout $0 \leqslant s < t$ et donc en particulier elle est vraie pour $s = 0$:

$$u(t,x) = u(0,X(0)) = u_0(X(0)) = u_0(x - ct)$$

La solution du problème est donc $u(t,x) = u_0(x - ct)$. Ainsi, dans le cas plus général du problème de transport suivant :

$$\partial_t u + c\nabla u = 0 \Leftrightarrow \frac{Du}{Dt} = 0$$

on peut écrire

$$\frac{Du}{Ds}(s,X(s)) = \partial_t u(s,X(s)) + X'(s)\nabla u(s,X(s))$$

où X vérifie $\begin{cases} X'(s) = c, \qquad t_n \leqslant s \leqslant t_{n+1} \\ X(t_{n+1}) = x \end{cases}$

ce qui peut s'approximer ainsi :

$$\frac{Du}{Ds}(s,X(s)) \simeq \frac{u(t_{n+1},X(t_{n+1})) - u(t_n,X(t_n))}{\delta t} = \frac{u(t_{n+1},x) - u \circ X^n}{\delta t} = \frac{u^{n+1} - u \circ X^n}{\delta t}$$

6.2 Code FreeFem

```
//Constant of the square domain
real x0=-1,x1=1;
real y0=-1,y1=1;
real a=1;

// Number of nodes on each boundaries of the square defined below
int m = 50;

// Number of total triangle
real nbtri = 0;

//Maillage
mesh Th=square(m,m,[x0+(x1-x0)*x,y0+(y1-y0)*y]);

// Execution time
real Tf  = 0.5,
// time step
     dt  = 0.01,
// viscosity coefficient
     eps = 0.1;

fespace Vh(Th,P2);

Vh phi = exp(-a*(x^2+y^2));

for(real t=0;t<Tf;t=t+dt){
    cout << " Time = " << t <<endl;

    // P2 Finite element space

    //phi will denote the unknown of the equation and here the initial data
    Vh phiex = exp(-a*(x^2+y^2+t)),

    //w is the test function
     w,

    //Constants of convection
    c1=sin(t), c2=cos(t),

    //f the source term
    f = -a*exp(-a*(x^2+y^2+t))-2*c1*a*x*exp(-a*(x^2+y^2+t))-2*c2*a*y*exp(-a*(x^2+y^2+t))
    -eps*4*a*exp(-a*(x^2+y^2+t))*(a*x^2+a*y^2-1);

    // Physical coefs
    verbosity=0;
    Th=adaptmesh(Th,phi);
    verbosity=1;

    //Solving
    Vh phiold=phi;
```

```
    Vh cc=convect([c1,c2],-dt,phi);
    solve AdvDiff(phi,w)
        =
          int2d(Th)(phi*w/dt +eps*(dy(phi)*dy(w))+eps*(dx(phi)*dx(w)))
        - int2d(Th)(f*w+cc/dt*w)
        - int1d(Th,1) (-2*a*exp(-a*(x^2+1+t))*w)
        - int1d(Th,2) (-2*a*exp(-a*(y^2+1+t))*w)
        - int1d(Th,3) (-2*a*exp(-a*(x^2+1+t))*w)
        - int1d(Th,4)(-2*a*exp(-a*(y^2+1+t))*w);

    plot(Th,phi,dim=3,fill=1,value=1); // Plot phi, to save just add ps="NameOfFigure.eps"

    Vh err = phi-phiex; // Error
    real L2error = sqrt(int2d(Th)(err^2));
    nbtri=nbtri+Th.nt;
    cout<< "L2error " << " = " << L2error << " nt " << Th.nt<< endl;

    };

cout<<"Triangle moyen " <<nbtri*dt/Tf<<"\n"; //Getting information
```