

ARNAUD RIVET

Ingénieur MACS – Mathématiques Appliquées & Calcul Scientifique

Estimations *a posteriori* pour la modélisation des écoulements et des transferts de polluants en milieux poreux

Rapport de stage de fin d'études

Soutenance le 27 septembre 2012

Maître de stage : Marc BOURGEOIS
Ingénieur/Chercheur IRSN

Tuteur de stage : Emmanuel AUDUSSE
Maître de conférence Université Paris XIII - LAGA

REMERCIEMENTS

Je tiens à remercier, en tout premier lieu, Monsieur Marc Bourgeois, mon maître de stage, qui m'a fait confiance pour les tâches que j'avais à mener et m'a permis d'approfondir mes connaissances sur le sujet des Estimations *a posteriori* notamment, tout en me donnant l'opportunité d'exercer mes compétences en Mathématiques Appliquées et en Informatique dans le cadre dynamique de l'IRSN.

Un grand merci à Madame Delphine Pellegrini, ainsi qu'à toutes les personnes du bureau qu'elle dirige, le BERIS, pour leur accueil chaleureux au sein de l'équipe et leur enthousiasme.

Je remercie mes professeurs, Monsieur Omnes, Monsieur Lafitte, Monsieur Vohralik, pour avoir su me diriger et me donner de précieuses recommandations techniques tout au long du stage.

Merci aussi à mes voisins de bureau, Jean-François Barbier et Florian Mathon, pour l'accueil, la bonne humeur, les services rendus ainsi que les conseils et les remarques diverses sur l'IRSN ou autres, qu'ils ont eu la gentillesse de me prodiguer.

Et bien sûr un grand merci aux stagiaires du BERIS, Mario Caruso et Philippe Ung, pour l'esprit de camaraderie, mais aussi pour les remarques (pertinentes ou non), suggestions ou aide apportée au cours de ces derniers mois.

Sommaire

1	Introduction	3
2	L'IRSN, son rôle, ses missions	4
2.1	Création et missions	4
2.2	Organisation	5
2.3	<i>MELODIE</i> : outil de simulation numérique	6
3	L'écoulement permanent	8
3.1	Description du problème	8
3.1.1	Equation considérée	8
3.1.2	Estimateur : un théorème mathématique	9
3.1.3	Utilisation du théorème	10
3.2	Mise en place d'un maillage adaptatif	11
3.2.1	Algorithme	11
3.2.2	Raffinement et déraffinement de maillage	12
3.2.3	Calcul des estimateurs	14
3.3	Comparaisons avec raffinements uniformes de maillage	17
3.3.1	Comportement des erreurs	17
3.3.2	Nécessité de précision	27
3.3.3	Parallélisation de fonctions	30
4	Le transport de radionucléides	32
4.1	Description du problème	32
4.1.1	Equation considérée	32
4.1.2	Estimateur <i>a posteriori</i>	33
4.2	Maillage adaptatif	34
4.2.1	Algorithme	34
4.2.2	Détails de l'implémentation	35
4.3	Comparaisons avec raffinement uniforme de maillage	38
4.3.1	Utilisation d'une solution analytique	38
4.3.2	Cas de dépôt de colis dans un milieu hétérogène	40
5	Conclusion provisoire	41
6	Annexes	42
6.1	Annexe 1 : Organigramme de l'IRSN	42
6.2	Annexe 2 : Equations gérées par <i>MELODIE</i>	43
6.3	Annexe 3 : Théorie des éléments de référence	46
6.4	Annexe 4 : Changements apportés pour l'écoulement permanent	47
6.5	Annexe 5 : Changements apportés pour le transport	49
7	Bibliographie	52

1 Introduction

Le choix du nucléaire en France pour la production d'électricité a été fait à la suite des chocs pétroliers des années 70, dans le but d'accroître l'indépendance énergétique du pays (taux avoisinant les 50% en 2005 selon le CEA). Si le parc des centrales nucléaires de la France s'avère moindre que celui des Etats-Unis, respectivement second et premier avec 59 et 104 réacteurs, notre pays apparaît, avec une part de 76%, comme celui qui dépend le plus de ce mode de production d'électricité.

Ce dernier génère bien des problèmes, dont un des plus importants peut être la génération des déchets, et par conséquent leur gestion. En effet deux types de déchets existent, selon la durée de demi-vie radioactive : ceux de faible et moyenne activité, d'une part, et ceux de haute et moyenne activité à vie longue, d'autre part. C'est surtout la gestion de cette dernière catégorie qui est en cours d'élaboration. L'ANDRA (Agence Nationale pour la gestion des Déchets Radioactifs) est chargée d'étudier la conception et l'implantation d'un centre permettant de traiter ces déchets, via un stockage à grande profondeur. C'est dans ce cadre là notamment que l'IRSN (Institut de Radioprotection et de Sécurité Nucléaire) cherche à développer des outils en interne, afin de modéliser les couches géologiques et d'y simuler les systèmes hydrogéologiques ainsi que la migration éventuelle de polluants radioactifs, les radionucléides.

Cette modélisation, i.e. la discrétisation d'un volume généralement conséquent du sous-sol et l'approximation des équations s'y exerçant, peut engendrer un nombre très élevé de données, et ceci malgré les simplifications déjà réalisées. Les temps d'obtention des résultats pouvaient être démesurés et par conséquent limiter les calculs, réduisant ainsi la possibilité d'analyse de sensibilité du modèle à des variations de valeurs des différents paramètres. Des méthodes d'Estimateurs *a posteriori* ont donc été mises en place afin d'optimiser le maillage tout en garantissant une certaine précision sur les résultats.

Le sujet initial du stage consistait à étudier cette version du code, afin de déterminer si possible les règles de comportement vis-à-vis des nouveaux paramètres de calcul, dans le cadre du 2D. Les premiers tests ont montré des erreurs de programmation ainsi que quelques incompréhensions au niveau algorithmique ; celles-ci ont alors orienté le sujet vers la correction et la vérification des nouvelles fonctions introduites afin de valider la méthodologie mise en place. C'est pourquoi la réalisation de tests à partir de fonctions analytiques permettra la validation de cette méthode de raffinement par rapport à des raffinements uniformes, notamment grâce à la comparaison des erreurs calculées.

2 L'IRSN, son rôle, ses missions

2.1 Création et missions

Etablissement public à caractère industriel et commercial, c'est dans une démarche d'indépendance et de transparence que l'Institut de Radioprotection et de Sûreté Nucléaire (IRSN) voit le jour. Sa création remonte au 9 mai 2001 et son fonctionnement est précisé en 2002, lors de la fusion entre l'Institut de Protection et de Sûreté Nucléaire (IPSN), d'une part, et l'Office de Protection contre les Rayonnements Ionisants (OPRI), d'autre part. L'IRSN est placé sous la tutelle conjointe de cinq Ministères différents.

Disposant d'un contrat d'objectifs avec l'Etat, l'IRSN voit ses missions organisées en trois domaines principalement, que l'on peut détailler de la manière suivante :

– **Recherche et missions de service public :**

- Définition et mise en oeuvre de programmes de recherche, nationaux et internationaux, destinés à maintenir et développer les compétences nécessaires
- Formations et enseignements en radioprotection, sûreté et sécurité nucléaires, dispensés notamment aux professionnels de santé et aux personnes professionnellement exposées
- Veille permanente en matière de radioprotection grâce à la surveillance radiologique de l'environnement, à la gestion des données dosimétriques relatives aux travailleurs exposés aux rayonnements ionisants et à la gestion des sources radioactives
- Contribution à l'information du public sur les risques nucléaires et radiologiques (publications, Internet, expositions, colloques...)

– **Appui et concours technique et opérationnel aux pouvoirs publics et aux autorités :**

- Appui technique en matière de risques nucléaires et radiologiques pour les installations, les transports de substances radioactives, l'application des traités sur le contrôle des matières nucléaires et sensibles, la protection physique et la sécurité des applications industrielles et médicales
- Appui opérationnel en cas de crise ou de situation d'urgence radiologique proposé aux pouvoirs publics et aux autorités pour assurer la protection de la population, des travailleurs et de l'environnement, et pour rétablir la sécurité des installations

– **Prestations contractuelles d'expertise, de recherche et de mesure :**

- Réalisation d'expertises, recherches, analyses, mesures ou dosages pour des organismes publics ou privés, français ou non

2.2 Organisation

Pour faire face à cette diversité de missions, l'IRSN regroupait en 2011 1718 collaborateurs (54% d'hommes, 46% de femmes), présents sur 11 sites en France. 74% de ce personnel sont des ingénieurs, des chercheurs ou des cadres, les autres étant des techniciens et employés de support technique ou administratif. En effet 40% du budget de l'IRSN (293 millions d'euros de recettes, 282 millions de dépenses) sont consacrés à la recherche.

Hormis la Direction générale et les Directions fonctionnelles et de support (Ressources humaines, Communication...), la répartition des employés est réalisée sur 3 pôles (cf. *AN-NEXE 1 : Organigramme*) :

- Défense, sécurité et non-prolifération
- Sûreté nucléaire
- Radioprotection, environnement, déchets et crise

Chacun de ces pôles comporte plusieurs directions, comprenant chacune plusieurs services, qui sont composés eux-mêmes de plusieurs bureaux ou laboratoires. Le stage a donc eu lieu dans ce dernier pôle, "Pôle RadioProtection", dans la direction "Déchets-GEosphère" (PRP-DGE), dans le "Service d'Expertise des Déchets RAdioactifs et de la radioactivité Naturelle" (SEDRAN), et finalement au sein du "Bureau d'Expertise et de Recherche sur les Installations de Stockage" (BERIS). Ceci donne l'architecture suivante pour le stage : PRG-DGE/SEDRAN/BERIS.

Le BERIS est composé de onze personnes, hors secrétaires et stagiaires, dont le statut pour la majorité est Ingénieur-Chercheur. En effet hormis cette dernière catégorie, se trouve Mme Delphine Pellegrini, Chef de bureau, et deux chargés d'affaires.

C'est afin de pourvoir à sa mission de contrôle et de vérification que l'IRSN crée dans ses bureaux et ses laboratoires, ses propres outils. C'est notamment le cas au BERIS où le logiciel *MELODIE* est développé en interne.

2.3 MELODIE : outil de simulation numérique

Modèle d'Evaluation à LOng terme des Déchets Irradiants Enterrés

L'IRSN a créé et continue à développer le logiciel *MELODIE* afin de disposer d'un outil interne pour évaluer la sûreté à long terme d'un stockage de déchets radioactifs. Cet outil est utilisé notamment pour apprécier les performances des sites de stockage proposés par l'ANDRA, organisme chargé de la gestion des déchets radioactifs. Ainsi l'IRSN a pu, avec l'utilisation du logiciel entre autres, rédiger un avis sur le dossier "Argile" de l'ANDRA en 2005.

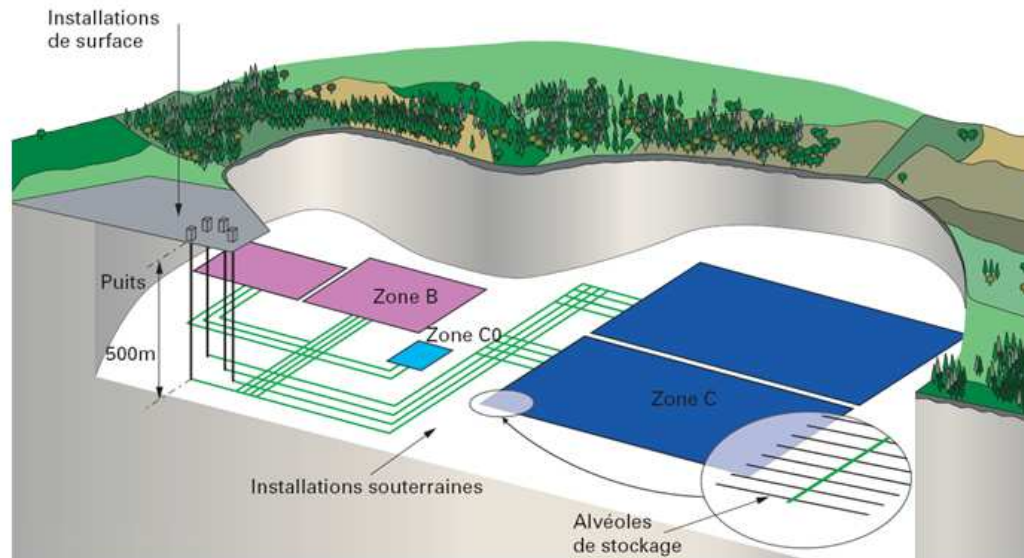


FIG. 1 – Schéma de principe d'une architecture de stockage (ANDRA, dossier Argile 2005)

Le modèle considéré dans *MELODIE* est formé d'un système couplé : une équation elliptique stationnaire pour l'écoulement et une équation de type diffusion-convection-réaction pour l'activité des radionucléides (cf. ANNEXE 2).

Pour la résolution de ces équations, *MELODIE* est constituée de plusieurs modules :

- **MELOMAIL & MELOBUILD** : à partir d'un fichier scénario initial, composé de commandes simples pour la réalisation du maillage du domaine à étudier ou encore pour la définition des paramètres à prendre en compte pour le calcul souhaité, ces modules permettent de retourner un fichier qui est ensuite pris en entrée par le code de calcul.
- **MELO** : code de calcul du logiciel, permettant la simulation des écoulements et/ou du transport de radionucléides. Ce sera sur les fonctions de ce programme, rédigées en FORTRAN (77-90), que la méthodologie des estimateurs *a posteriori* interviendra.
- **MELOVIEW** : module permettant la visualisation de données issues des fichiers résultats de *MELO*.

MELODIE permet donc une modélisation globale d'un site de stockage en tenant compte des principaux phénomènes physico-chimiques en action dans ce type d'ouvrage, sous une forme simplifiée, afin que puissent être traitées les grandes échelles, à la fois en espace et en temps lorsque ce dernier intervient.

L'échelle spatiale pour modéliser le relâchement de radionucléides en sous-sol entraîne un problème sur le volume des données à traiter. En effet le site de stockage représente déjà un domaine assez étendu, qui se compte en km^2 par exemple pour le dossier précédemment cité (coupe horizontale) et donc en millions de mailles. Mais en cas de migration des radionucléides, ces derniers ne restent pas cantonnés à l'installation de stockage et il faut donc représenter une partie conséquente du sous-sol afin de pouvoir avoir une vue globale de cette migration.

C'est pourquoi il est impératif d'avoir un maillage adaptatif, concentré aux endroits nécessaires et qui, comme son nom l'indique, s'adapte au comportement du transport des radionucléides.

Deux notions sont nécessaires au calcul de migration des radionucléides à travers un milieu saturé : les caractéristiques hydrologiques de l'eau dans un milieu poreux et le comportement d'un soluté dans ce même milieu. Cela signifie donc plusieurs équations et donc plusieurs estimateurs *a posteriori* à implémenter, ces derniers dépendant de l'équation considérée.

3 L'écoulement permanent

3.1 Description du problème

3.1.1 Equation considérée

Le premier problème traité sera l'équation de l'écoulement en milieu saturé, sans considération temporelle. En effet la solution de cette équation nous permet d'obtenir la vitesse de Darcy, nécessaire au calcul de transport de radionucléides. Or chaque équation possède des estimateurs qui lui sont propres. Une analyse est alors nécessaire afin de déterminer quel estimateur régissant le maillage, celui issu de l'équation d'écoulement ou de l'équation du transport, sera déterminant au temps t . Pour s'affranchir de cela, on considère donc dans un premier temps un écoulement en régime permanent, donné par le système suivant :

$$\left\{ \begin{array}{lll} -div(\mathbf{K}\nabla h) & = & f \quad \text{dans } \Omega \\ h & = & 0 \quad \text{sur } \Gamma_D \\ \mathbf{K}\nabla h \cdot \vec{n} & = & g \quad \text{sur } \Gamma_N \end{array} \right.$$

avec :

h : charge hydraulique

ρ : masse volumique de l'eau

g : accélération de la pesanteur

ν : viscosité dynamique de l'eau

$\mathbf{K}_r = \begin{pmatrix} a_{xx} & a_{xy} \\ a_{xy} & a_{yy} \end{pmatrix}$ perméabilité intrinsèque et $\mathbf{K} = \frac{\rho^*g}{\nu}\mathbf{K}_r$ tenseur de perméabilité

La vitesse de Darcy \vec{U} est alors obtenue par la relation :

$$\vec{U} = -\mathbf{K}\nabla h$$

Le fait de choisir une solution affine par triangle pour la charge hydraulique permet d'obtenir une vitesse de Darcy constante sur chaque maille.

On choisira de plus \mathbf{K}_r constant par élément (chaque composante constante) et notons alors \mathbf{K}^T ce tenseur constant sur le triangle T .

3.1.2 Estimateur : un théorème mathématique

Définissons quelques variables :

h_T : grandeur caractéristique du triangle T (rayon du cercle inscrit ou diamètre de T)

h_E : longueur de l'arête E

\vec{n}_E : normale sortante à l'arête E (pour un triangle donné)

K_T : rayon spectral du tenseur de perméabilité sur l'élément T

$\mu_T = h_T * K_T^{-1/2}$

$\mu_E = h_E * (\max\{K_{T'}; \partial T' \ni E\})^{-1}$

En notant $||.||_T$ et $||.||_E$ les normes $L2$ sur T et E respectivement, et $[[.]].\vec{n}_E$ le saut sur l'arête E , on peut définir les grandeurs suivantes :

$$\begin{aligned} r_T(u_h) &= f + \operatorname{div}(\mathbf{K} \nabla u_h) \\ r_E(u_h) &= \begin{cases} -[\mathbf{K} \nabla u_h].\vec{n}_E & \text{si } E \in E_{h,int} \\ g - \mathbf{K} \nabla u_h.\vec{n}_E & \text{si } E \in E_{h,N} \\ 0 & \text{si } E \in E_{h,D} \end{cases} \end{aligned}$$

où $[[.]].\vec{n}_E$ est le saut normal à travers l'arête E .

Notons finalement $u_{h,T} := u_h|_T$

Le théorème mathématique, dont la démonstration peut être trouvée dans [1], nous assure l'inégalité suivante :

$$||u - u_h||_K \leq C\eta$$

où $||u||_K^2 = \int_{\Omega} (\mathbf{K} \nabla u, \nabla u)$ est la norme d'énergie de u , C une constante ne dépendant ni du tenseur ni du domaine, et où finalement l'estimateur d'erreur *a posteriori* η est défini par :

$$\eta = \left(\sum_{T \in T_h} \mu_T^2 ||r_T(u_h)||_T^2 + \sum_{E \in E_h} \mu_E ||r_E(u_h)||_E^2 \right)^{1/2}$$

Remarque : l'estimateur est construit à partir de la solution numérique u_h uniquement

Nous allons voir dans la partie suivante comment peut être utilisé ce théorème pour la mise en place d'un maillage adaptatif.

3.1.3 Utilisation du théorème

Le théorème précédent nous donne une majoration de l'erreur pour la norme d'énergie, majoration appelée "Estimateur *a posteriori*". Cet estimateur est construit à partir des données géométriques du maillage et de la solution numérique obtenue. De plus, lorsque le nombre de sommets devient assez élevé, cette construction a un coût négligeable par rapport à la résolution du système linéaire.

Mathématiquement ce n'est pas l'erreur qui nous intéresse mais l'erreur relative, qui est donnée simplement par :

$$\frac{\|u - u_h\|_K}{\|u\|_K} \text{ ou bien par } \frac{\|u - u_h\|_K}{\|u_h\|_K}$$

Si la première a l'avantage de se baser sur la solution exacte, et donc "meilleure" mais malheureusement inconnue *a priori*, la seconde prend en compte la solution numérique au dénominateur, dont la norme est calculable. C'est donc cette dernière expression de l'erreur relative qui est utilisable numériquement.

En effet si le théorème des estimations *a posteriori* nous garantit que l'erreur en norme K est bornée par η , donc que l'erreur relative est bornée par $\frac{\eta}{\|u_h\|_K}$, et si ce dernier terme est plus petit qu'une grandeur, appelons-la ϵ , alors *a fortiori* l'erreur relative sera elle-même plus petite que cet ϵ :

$$\frac{\|u - u_h\|_K}{\|u_h\|_K} \leq \frac{\eta}{\|u_h\|_K} \leq \epsilon \quad (1)$$

Cet ϵ représente la précision souhaitée. En effet si nous arrivons à construire un algorithme tel qu'au bout d'un certain nombre d'itérations on ait :

$$\eta \leq \epsilon * \|u_h\|_K$$

alors cela signifiera que la dernière inégalité (1) sera respectée, i.e. que la précision ϵ sera atteinte, quel que soit ϵ .

Nous appellerons $\epsilon * \|u_h\|_K$ la tolérance dans la suite de cette partie.

Grâce au raffinement de maillage, l'erreur et l'estimateur vont décroître. L'idée, comme on le verra dans la partie suivante avec l'algorithme du programme, sera alors de sélectionner les éléments qu'on souhaite raffiner dans le but de faire décroître l'estimateur, afin qu'il passe sous la tolérance voulue et donc de garantir la précision souhaitée pour l'erreur relative.

3.2 Mise en place d'un maillage adaptatif

3.2.1 Algorithme

Nous allons décrire l'algorithme mis en place pour le maillage adaptatif. Nous prendrons un nombre d'itérations maximum, fixé par l'utilisateur, afin de garantir la terminaison de l'algorithme. En effet si la précision souhaitée est trop petite, on peut mettre beaucoup de temps pour atteindre le critère d'arrêt, ou encore la déformation des triangles lors du raffinement peut altérer le bon déroulement du processus. Voici une description des variables, lues sur le fichier *Refine_perm.data* pris en entrée :

nb_raft_lim : nombre d'itérations maximum de la procédure
Tolmax : précision souhaitée sur la norme d'énergie
nloop : fréquence d'appel à la procédure de déraffinement

Notons que le choix des coefficients $coeff_1$ et $coeff_2$, utilisés respectivement pour le raffinement et le déraffinement, méritent une étude plus approfondie afin d'optimiser la procédure. Nous prendrons pour les premiers tests $coeff_1 = 0.25$. Quant à $coeff_2$, il ne sera pas utilisé puisque nous ne ferons pas de déraffinement immédiatement. En effet nous voulons nous assurer dans un premier temps que le raffinement seul nous permet une décroissance de l'erreur.

Entrées : *nb_raft_lim, Tolmax, nloop*

Sortie : *données relatives au maillage et résultats*

POUR *ilevel = 1 à nb_raft_lim*

```
.   Assemblage et résolution du système linéaire
.   Tolérance = Tolmax *  $\|u_h\|_K$ 
.   Calcul des estimateurs  $\eta_T$  sur chaque élément  $T$  et  $\eta_{\max} = \max\{\eta_T; T \in T_h\}$ 
.   Estimateur global :  $\eta = \sqrt{\sum_T \eta_T}$ 
.   SI  $\eta \leq \textit{Tolérance}$  ALORS
.       .   FIN : la précision souhaitée est garantie
.   SINON
.       .   On raffine les éléments  $T$  tel que  $\eta_T > coeff_1 * \eta_{\max}$ 
.       .   On adapte les conditions aux limites et les valeurs (interpolation linéaire)
.       .   SI ilevel est un multiple de nloop ALORS
.           .   On déraffine les éléments  $T$  tel que  $\eta_T < coeff_2 * \eta_{\max}$ 
.       .   FIN SI
.   FIN SI
FIN POUR
```

Enregistrement des résultats

Remarque : on peut prendre en compte d'autres variables, telles que *i_choice* (= 0 ou 1) et *nmu_ini* qui permettent de fixer une limite au déraffinement (on ne peut pas détruire les noeuds dont le numéro est inférieur ou égal à *nmu_ini*), *iTyp_ref* qui indique le type de raffinement à utiliser, ou encore *surf_limit* qui impose une surface maximale pour les éléments du maillage.

3.2.2 Raffinement et déraffinement de maillage

Le maillage a un rôle clé dans le bon déroulement du calcul. En effet il est important que les mailles vérifient certains critères pour assurer un bon comportement de la solution. En 2D le critère à respecter est celui de Delaunay : *soit T un triangle du maillage, alors seuls les sommets formant T sont contenus à l'intérieur du cercle circonscrit à T* (illustration ci-dessous).

Afin que ce critère soit respecté (le plus possible), une fonction est mise en place. Elle se charge d'intervertir, si besoin, l'arête commune entre deux triangles voisins (*FIG. 2*).

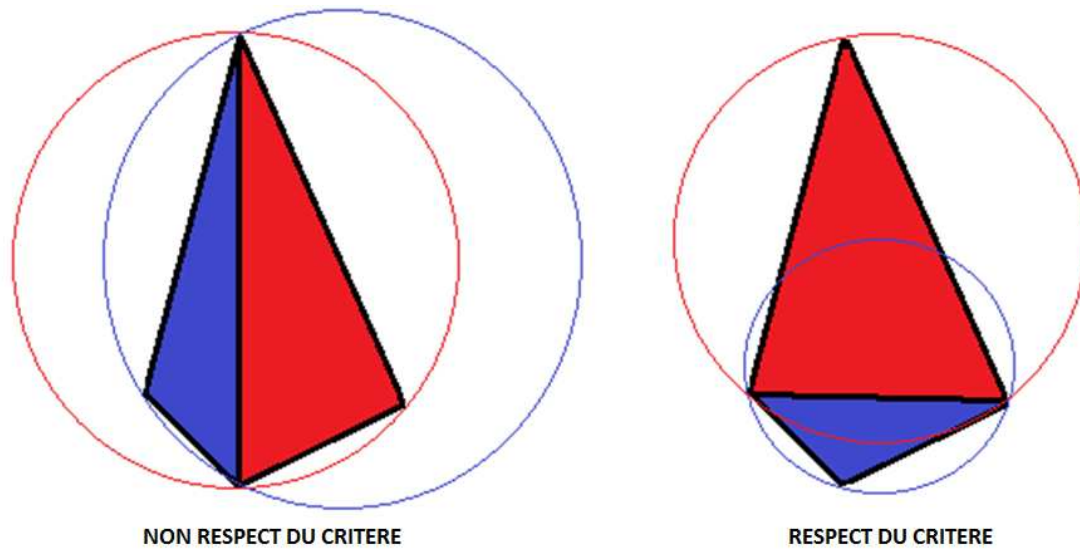


FIG. 2 – Réaffectation d'arête en vue de vérifier le critère de Delaunay

Le raffinement de maillage, tout comme le déraffinement, est assez délicat, puisqu'il s'agit d'une modification d'une géométrie qui, *a priori*, correspondait bien. Un raffinement localisé engendre forcément plus de problèmes puisque la géométrie initiale des triangles ne peut pas être gardée dans le cadre des maillages conformes.

Plusieurs méthodes de raffinement ont été testées, et voici une présentation de celle qui sera utilisée dans le cadre de *MELODIE*.

Si un triangle a besoin d'être raffiné, celui-ci sera séparé en 3 en joignant le centre de gravité aux sommets. Cette méthode a été choisie de par sa simplicité de mise en place, malgré l'engendrement de déformations assez importantes du triangle. Une fonction, voir ci-dessous, permettra de bouger les arêtes si besoin afin que le critère de Delaunay soit respecté pour un maximum de triangles.

Si le triangle à raffiner présente une arête au bord, on peut séparer le nouveau triangle du bord en créant un nouveau sommet au milieu de l'arête en question et en reliant ce nouveau sommet avec le centre de gravité (*FIG 3*). Cela permet d'avoir une géométrie plus adaptée au critère de Delaunay : les triangles sont en effet moins déformés.

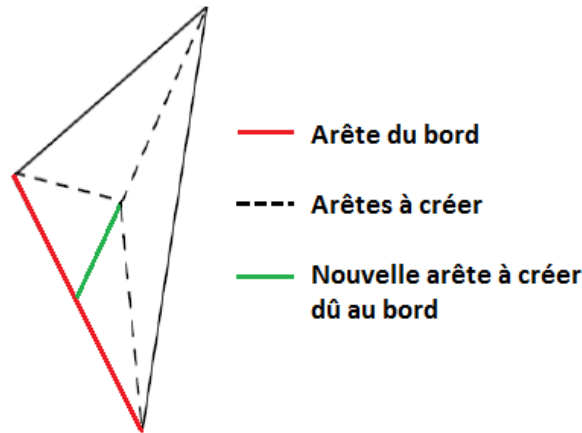


FIG. 3 – Raffinement d'un triangle au bord

Les valeurs nécessaires aux points créés sont obtenus par interpolation linéaire des 2 (pour le point créé au bord) ou des 3 (pour le sommet créé au centre de gravité) sommets déjà existant.

Pour déraffiner un triangle, il va falloir trouver l'ensemble de ses voisins (triangles ayant un sommet et/ou une arête en commun). On créera ensuite un noeud au centre de gravité du triangle et on joindra les noeuds qui étaient reliés à un des noeuds du triangle, au centre de gravité. On détruira finalement les noeuds du triangle, à moins qu'un des noeuds ne fasse parti du bord, auquel cas il sera relié au centre de gravité.

Comme pour le raffinement, les valeurs au centre de gravité seront obtenues par interpolation linéaire des valeurs aux sommets et si le déraffinement entraîne l'apparition d'un triangle au bord, ce dernier pourra être redécoupé en 2 triangles.

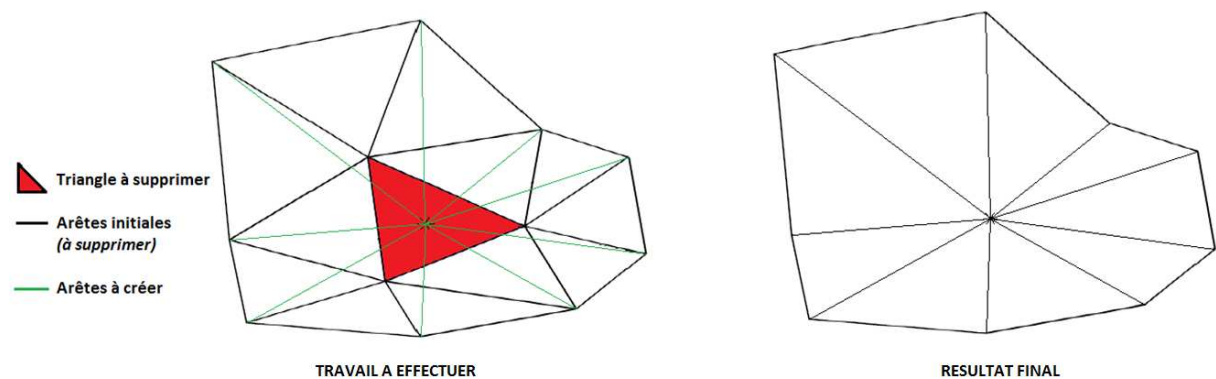


FIG. 4 – Méthode de déraffinement

3.2.3 Calcul des estimateurs

Fonction *metric.f* : il s'agit de la fonction qui calcule l'estimateur *a posteriori*. Le calcul de la norme K de la solution numérique, donc de la tolérance, et les calculs des erreurs dans le cadre des solutions analytiques, sont effectués eux aussi dans cette fonction.

- Séparation des calculs de r_T et r_E :

Une première chose possible pour améliorer la vitesse du code est de faire les calculs de chaque estimateur, r_T et r_E , dans des boucles séparées (boucles sur les triangles). En effet pour obtenir r_E , on doit parcourir chaque élément T et pour chacun de ses voisins T' , on doit effectuer le calcul de $\mathbf{K}^{T'} \nabla u_{h,T'}$ afin de déterminer le saut à travers les arêtes de T . On calculera donc n fois $\mathbf{K}^T \nabla u_{h,T}$ ($n = 1, 2$ ou 3 , n étant le nombre de voisins de T).

On peut s'affranchir de cela en gardant dans un tableau les composantes de $\mathbf{K}^T \nabla u_{h,T}$ pour chaque élément, lors d'une première boucle où l'on calcule les r_T .

Dans l'implémentation, ce tableau est appelé *Kgradh* et est par conséquent de taille $2 \times \text{nbre d'éléments}$.

- Estimateurs de résidu r_T :

$$\sum_{T \in T_h} \mu_T^2 \|r_T(u_h)\|_T^2 = \sum_{T \in T_h} \mu_T \int_T [f + \text{div}(\mathbf{K} \nabla u_h)]^2$$

La solution étant prise d'ordre 1, i.e. un plan sur chaque triangle, $f + \text{div}(\mathbf{K} \nabla u_h) := f$. Donc la contribution sur chaque triangle T du résidu est réduit à l'expression

$$\mu_T^2 \int_T f^2$$

Si f est connue analytiquement, ce terme peut être calculé en même temps que $\int_{C_i} f$ comme on le verra par la suite **PAGE A METTRE POUR SUIVRE**. En effet la précision de ces intégrales est primordiale pour la précision de la solution numérique. En revanche si f est seulement connue aux sommets du maillage, l'intégrale sur T de ce terme au carré peut être approchée à partir des 3 sommets par la formule de Hammer, exacte pour un polynôme d'ordre 2. En notant d_1 , d_2 et d_3 les moyennes du résidu sur chaque arête du triangle T (en l'occurrence les moyennes de f , puisque le terme en divergence est nul), et A_T son aire, on a la formule suivante :

$$\int_T [f + \text{div}(\mathbf{K} \nabla u_h)]^2 = \int_T f^2 = A_T * (d_1^2 + d_2^2 + d_3^2)/3$$

De plus, on peut prendre pour h_T le rayon du cercle inscrit, facilement calculable par la formule :

$$r = \frac{2 * A_T}{p_T} \text{ avec } p_T \text{ est le périmètre de } T \text{ et } A_T \text{ son aire}$$

ou alors le diamètre du triangle : la plus grande distance entre 2 points du triangle, i.e. la plus longue arête du triangle.

– Estimateurs de saut r_E :

On boucle sur les éléments et pour chaque élément on teste pour chaque arête afin de déterminer si on a la présence d'un voisin (tableau *nvois*). Le calcul de $\|r_E(u_h)\|_E$ est alors très simple si on considère la solution u_h affine par élément, car dans ce cas $\mathbf{K}^T \nabla u_{h,T}$ est un vecteur constant sur T , donc *a fortiori* sur l'arête du triangle.

Dans le cas d'un voisin présent sur l'arête, on s'assure d'abord que l'indice du voisin T' est bien supérieur à T . En effet le parcours des éléments ayant été fait par ordre croissant, si $T' < T$ alors le calcul des sauts aura déjà été fait précédemment, à l'élément T' . On garde l'idée de mettre la moitié de la contribution du saut dans chaque élément.

On a alors la formule suivante dans le cas d'une arête interne (i.e présence d'un voisin) :

$$\begin{aligned}
 \mu_E \|r_E(u_h)\|_E^2 &= \mu_E \int_E r_E(u_h)^2 \\
 &= \mu_E \int_E (-[\mathbf{K} \nabla u_h] \cdot \vec{n}_E)^2 \\
 &= \mu_E \int_E ((\mathbf{K}^T \nabla u_{h,T} - \mathbf{K}^{T'} \nabla u_{h,T'}) \cdot \vec{n}_E)^2 \\
 &= \mu_E * h_E * ((\mathbf{K}^T \nabla u_{h,T} - \mathbf{K}^{T'} \nabla u_{h,T'}) \cdot \vec{n}_E)^2
 \end{aligned}$$

Bien sûr dans le cas d'une arête au bord, la formule est similaire :

$$\mu_E \|r_E(u_h)\|_E^2 = \begin{cases} \mu_E * h_E * (g - \mathbf{K}^T \nabla u_{h,T} \cdot \vec{n}_E)^2 & \text{si } E \in E_{h,N} \\ 0 & \text{si } E \in E_{h,D} \end{cases}$$

– Tolérance :

La tolérance est définie par :

$$\text{Tol} = \epsilon * \|u_h\|_K$$

où ϵ est la précision souhaitée, lue dans 'Refine_perm.data' (*Tolmax*).

Le calcul de la norme d'énergie (au carré) peut se faire dans la même boucle que le calcul de r_T . En effet tous les éléments nécessaires au calcul de la norme sont constants par élément :

$$\begin{aligned}
||u_h||_K^2 &= \int_{\Omega} (\mathbf{K} \nabla u_h, \nabla u_h)_{\Omega} \\
&= \sum_{T \in T_h} \int_T (\mathbf{K} \nabla u_h, \nabla u_h)_T \\
&= \sum_{T \in T_h} \int_T (\mathbf{K}^T \nabla u_{h,T}) \cdot (\nabla u_{h,T}) \\
&= \sum_{T \in T_h} A_T * (\mathbf{K}^T \nabla u_{h,T}) \cdot (\nabla u_{h,T})
\end{aligned}$$

On transformera ensuite cette norme d'énergie en tolérance, en prenant la racine de la norme et en la multipliant cette dernière par la précision.

Cette tolérance, comme nous avons pu le voir précédemment, nous servira à définir notre critère d'arrêt lié à la précision souhaitée.

La précision atteinte par rapport à la précision souhaitée sera à étudier lors des cas-tests suivants réalisés à partir de solutions analytiques.

– Tenseur :

On peut garder dans un tableau le tenseur qui est constant par élément. On évite ainsi de devoir recalculer les 3 termes propres à chaque triangle, comme ce qui avait été mis en place pour le tenseur de perméabilité dans le cadre de l'équation du transport.

On peut alors calculer le rayon spectral K_T de \mathbf{K}^T et le garder lui aussi dans un tableau, appelé *Kspec* dans le code. On aura en effet besoin d'avoir accès aux rayons spectraux des 2 triangles séparant une arête interne afin de pouvoir prendre le maximum pour le calcul de μ_E .

3.3 Comparaisons avec raffinements uniformes de maillage

3.3.1 Comportement des erreurs

Quelques tests de Benchmark ont déjà été réalisés pour le logiciel *MELODIE* et sa validation. Nous allons nous inspirer de ces derniers afin de comparer les résultats de précision obtenue grâce aux raffinements issus des estimateurs *a posteriori*, par rapport à des raffinements de maillage uniformes utilisés pour le Benchmark.

Les tests sont tous réalisés sur le carré unité $\Omega = [0, 1] \times [0, 1]$. Le tenseur de perméabilité est pris comme $\mathbf{K} = \begin{pmatrix} 1.5 & 0.5 \\ 0.5 & 1.5 \end{pmatrix}$. Nous allons calculer en plus de l'erreur relative d'énergie $e_K = \frac{\|(u-u_h)\|_K}{\|u_h\|_K}$ et l'erreur relative L^2 $e_{L^2} = \frac{\|u-u_h\|_{L^2}}{\|u\|_{L^2}}$. En effet la première nous permet de nous assurer que la précision est bien respectée, tandis que la seconde nous permet une comparaison avec les valeurs des tests du Benchmark notamment. Les résultats attendus sont les suivants : une décroissance d'ordre 1 pour l'erreur relative en norme K et un ordre 2 pour l'erreur relative en norme L^2 .

– 1^{er} test :

$$u(x, y) = 16x(1-x)y(1-y)$$

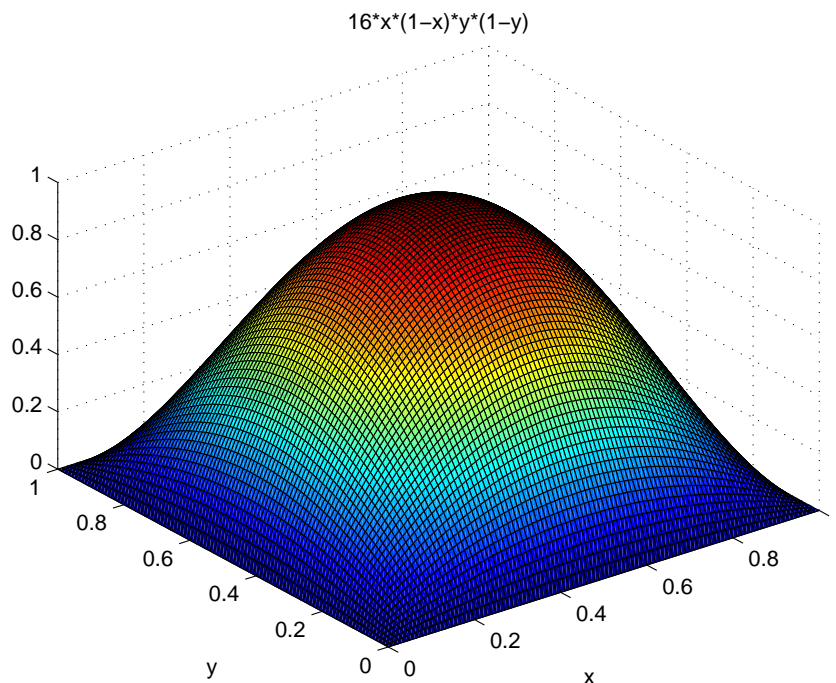


FIG. 5 – Solution analytique

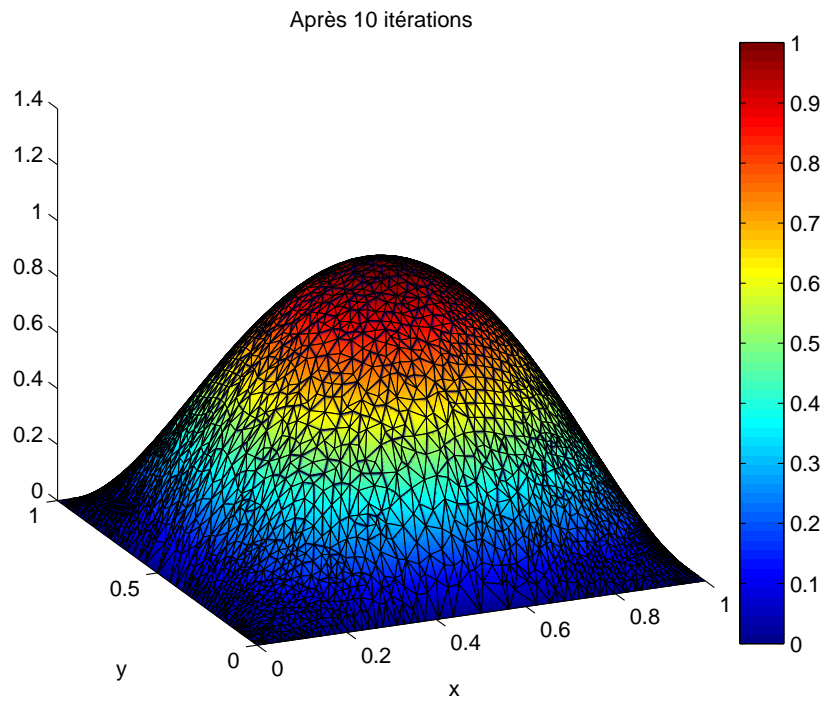


FIG. 6 – Solution numérique obtenue

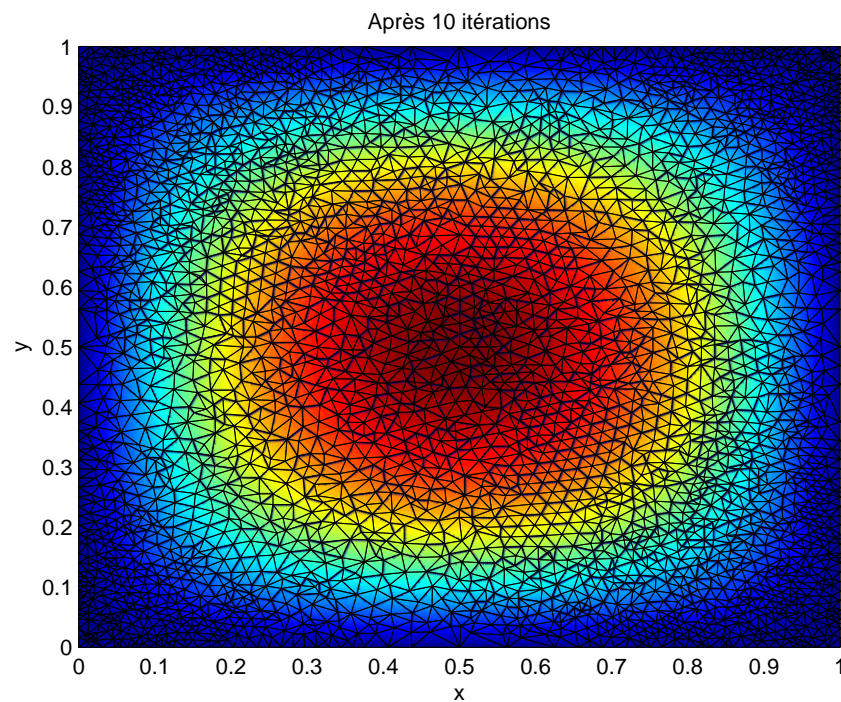


FIG. 7 – Maillage obtenu

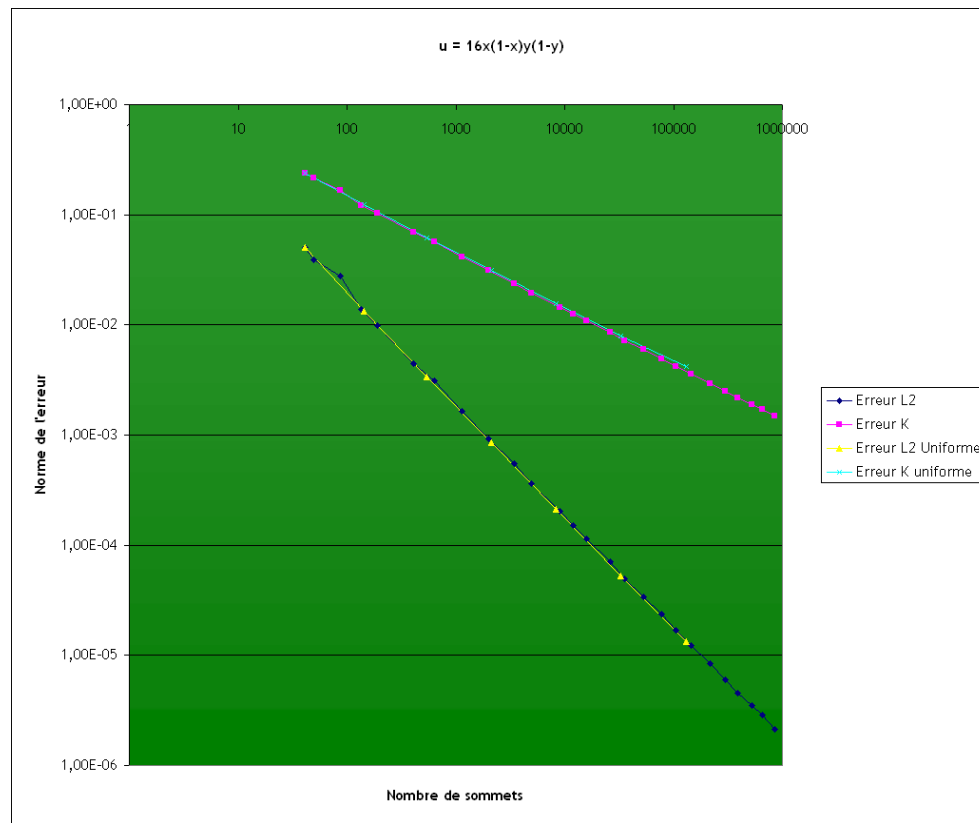


FIG. 8 – Convergence des normes

On remarque sur la *FIG. 8* que l'ordre de l'erreur e_{L^2} est retrouvé par rapport à un raffinement uniforme de maillage, même si les valeurs de cette norme sont légèrement plus faibles pour un maillage uniforme, à nombre de noeuds égal. Cela peut venir du fait que l'estimateur est construit par rapport à la norme d'énergie $||\cdot||_K$ et non pas par rapport à la norme L^2 , ou encore par le fait que la fonction est assez régulière, et que donc un maillage uniforme l'approchera forcément mieux.

A ce sujet on peut aussi remarquer grâce à la *FIG. 7* que le maillage obtenu est assez uniforme sur le domaine Ω .

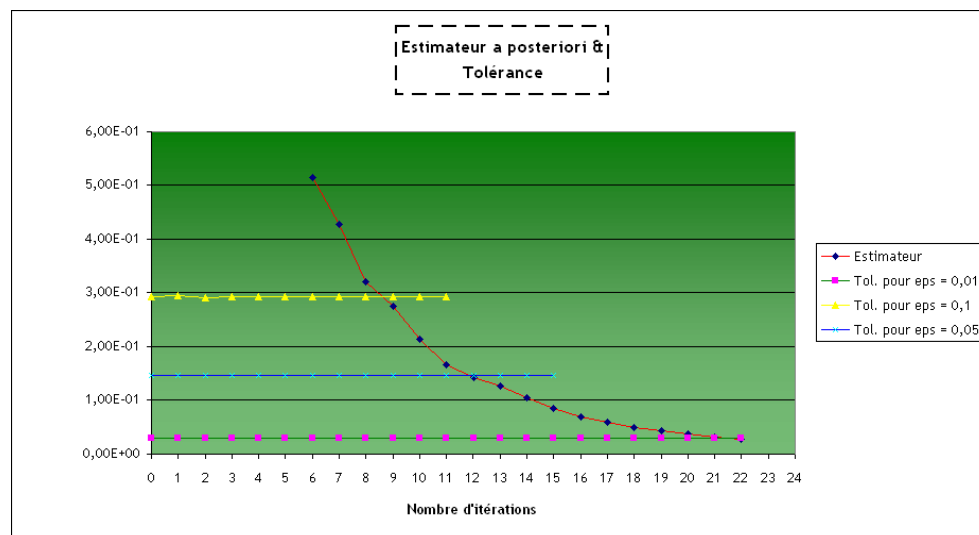


FIG. 9 – Comparaison des précisions

La FIG. 9 permet d'illustrer le comportement de l'estimateur avec le raffinement : il décroît toujours. La valeur de $\|u_h\|_K$ se stabilisant avec le raffinement, la tolérance apparait comme un palier que l'estimateur arrive, de par son comportement, à atteindre (rappel : $\text{Tol} = \epsilon * \|u_h\|_K$).

On peut alors essayer de voir quelle est la précision réelle obtenue par rapport à la précision fixée. Cette dernière est obtenue par la division de l'estimateur par la norme d'énergie à chaque itération. On remarque que ce rapport est à peu près constant : $3.0e-01$ environ (cette constante sera retrouvée pour les tests suivants).

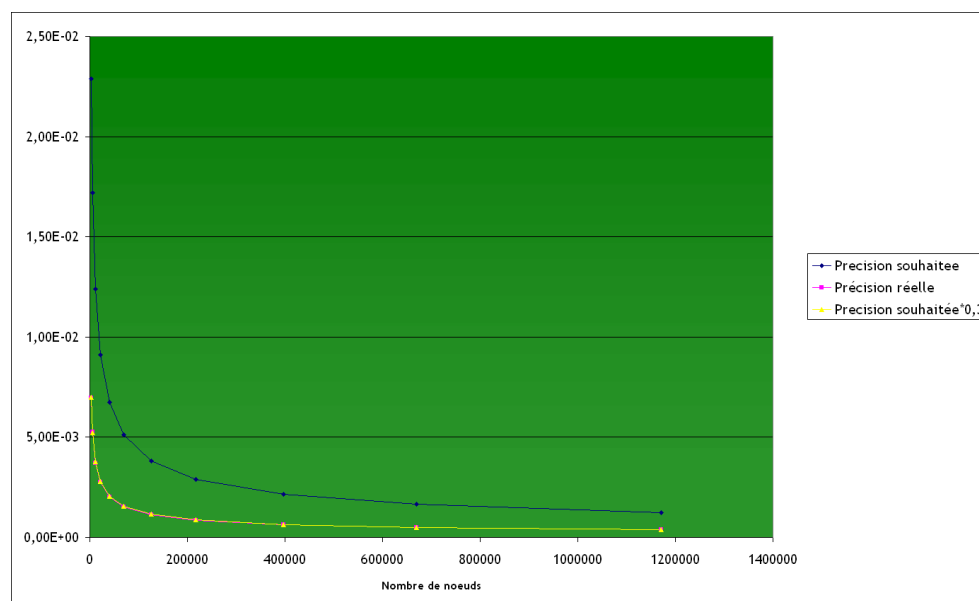


FIG. 10 – Comparaison des précisions

– 2^{me} test

$$u(x, y) = \sin[(1-x)(1-y)] + (1-x)^3(1-y)^2$$

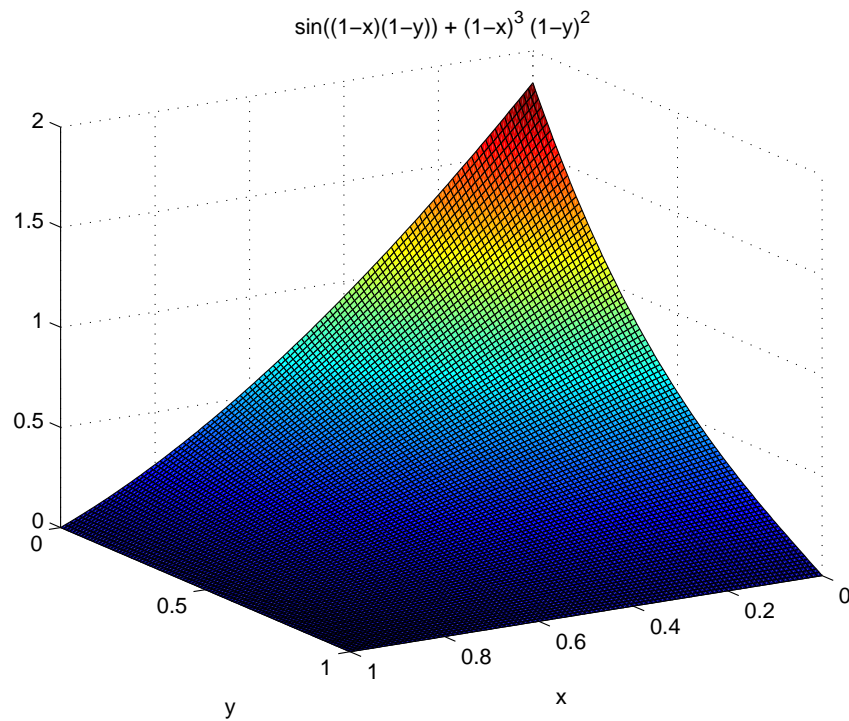


FIG. 11 – Solution analytique

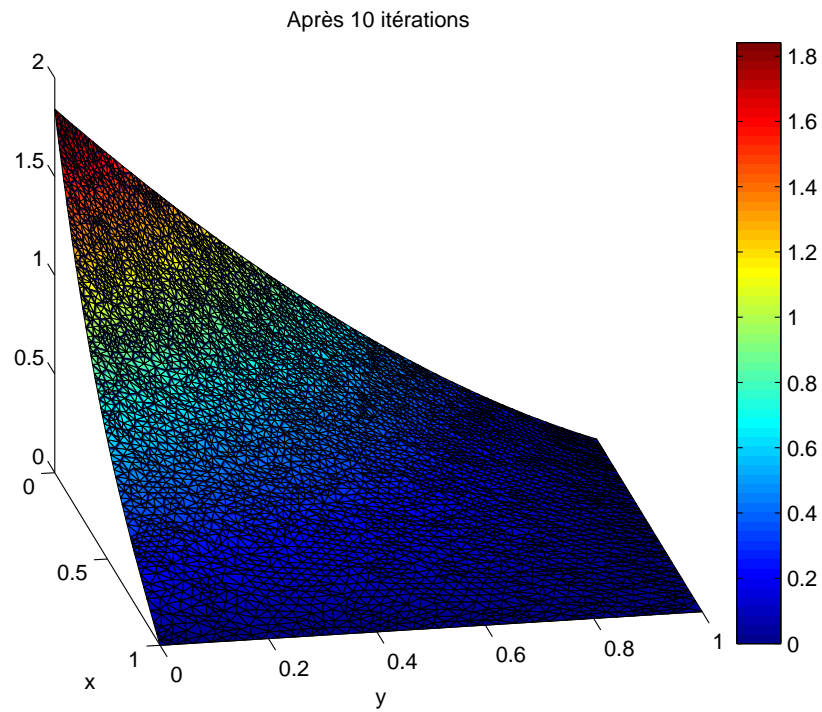


FIG. 12 – Solution numérique obtenue

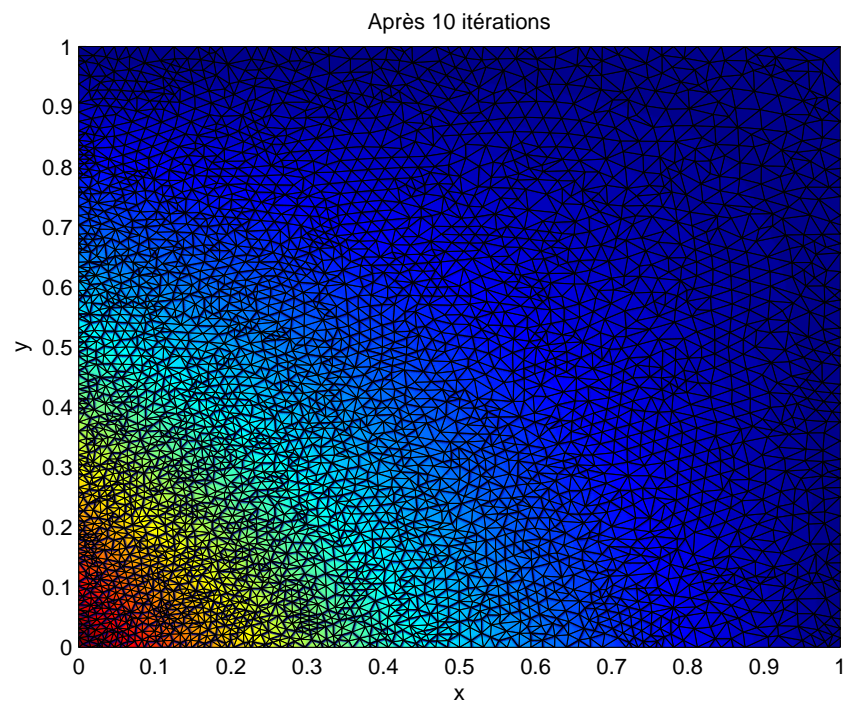


FIG. 13 – Maillage obtenu

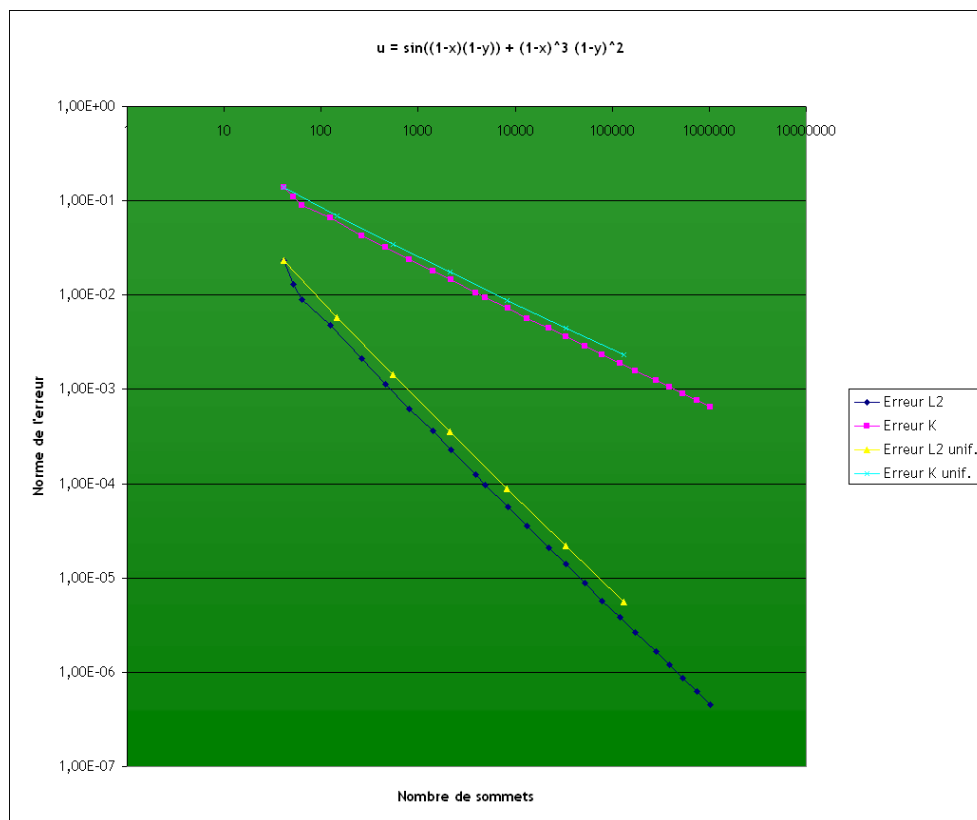


FIG. 14 – Convergence des normes

Sur le résultat de ce test (*FIG. 11*), si on retrouve globalement les ordres de convergence des normes du test précédent, il semblerait que le maillage adaptatif permet une légère amélioration par rapport à un raffinement uniforme. En effet les deux erreurs relatives obtenues par un raffinement adaptatif sont légèrement plus faibles que leurs homologues uniformes. Ceci semble possible grâce aux premières itérations qui permettent d'avoir un ordre plus élevé que celui attendu. Une étude de l'obtention d'ordres plus élevés peut être très intéressante, notamment par la mise en place de plusieurs tests sur le raffinement. On peut par exemple penser à évaluer le comportement selon le pourcentage d'éléments raffinés.

Le maillage obtenu (*FIG. 13*) est très intéressant : il semble être adapté à la géométrie de la fonction puisqu'il est raffiné autour de l'origine, là où la pente de la fonction est la plus forte, alors qu'autour du point (1,1) les éléments sont bien plus grands (pente proche de zéro dans cette région). Le maillage adaptatif semble convenir pour une fonction qui possèderaient de fortes variations. Nous allons donc nous intéresser à une fonction présentant cette caractéristique dans le cas-test suivant.

– 3^{me} test

$$u(x, y) = \exp\left(-200\left(x - \frac{1}{2}\right)^2 - 200\left(y - \frac{1}{2}\right)^2\right)$$

Cette fonction a été choisie par rapport au fait qu'elle soit très localisée dans le domaine Ω . On espère ainsi pouvoir avoir des résultats bien meilleurs en terme de précision avec un raffinement adaptatif de maillage par rapport à un maillage uniforme sur l'ensemble du domaine.

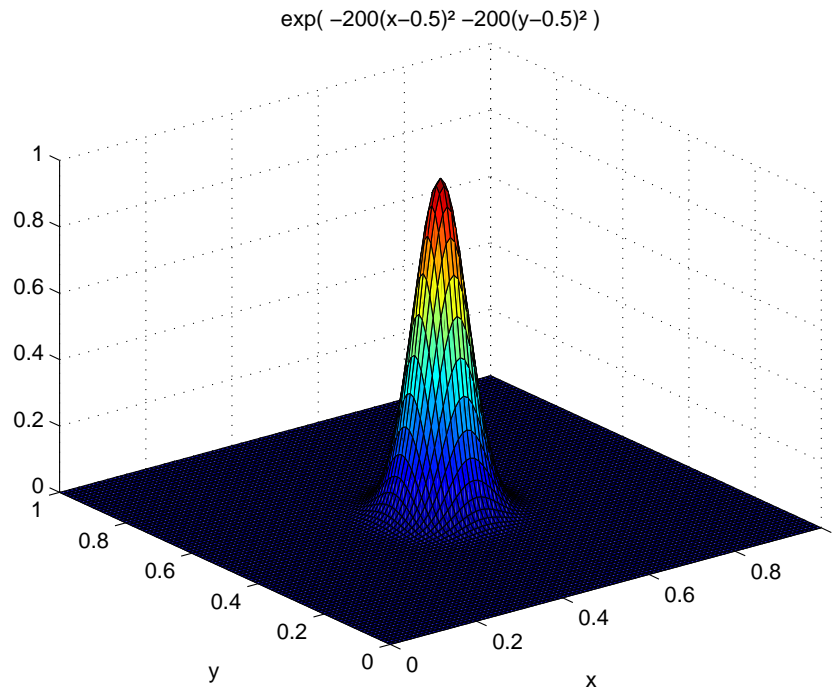


FIG. 15 – Solution analytique

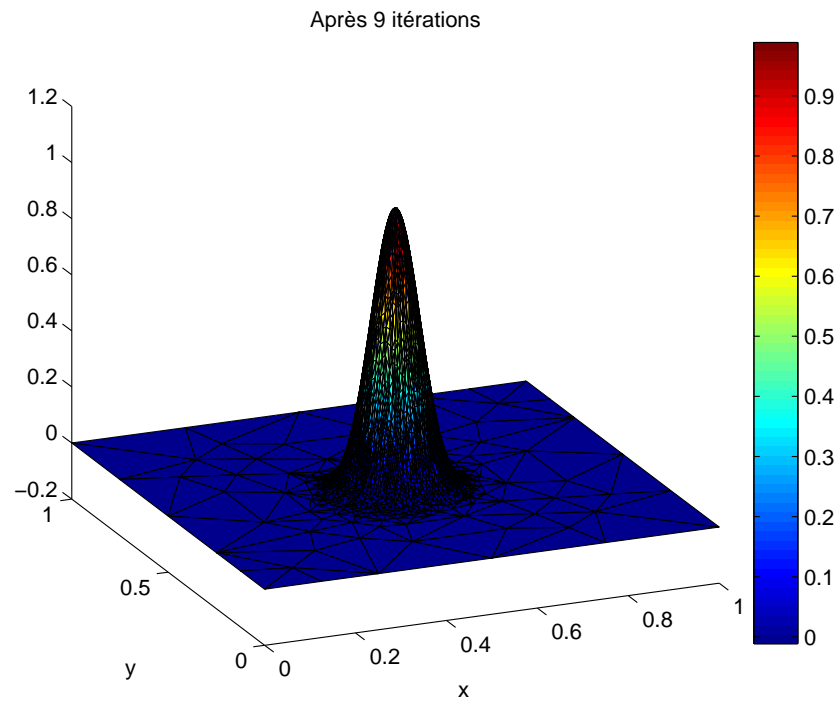


FIG. 16 – Solution numérique obtenue

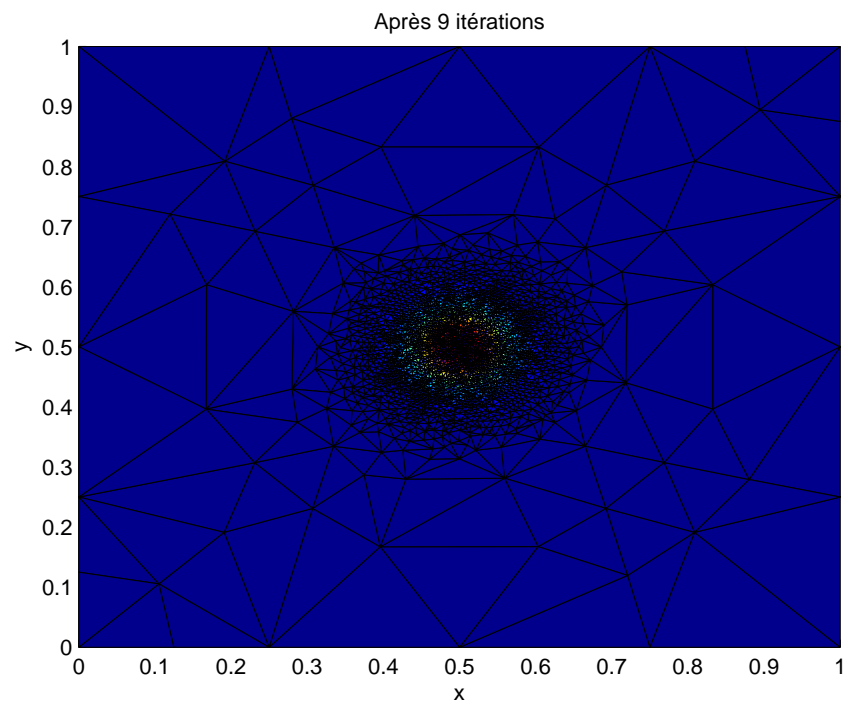


FIG. 17 – Maillage obtenu

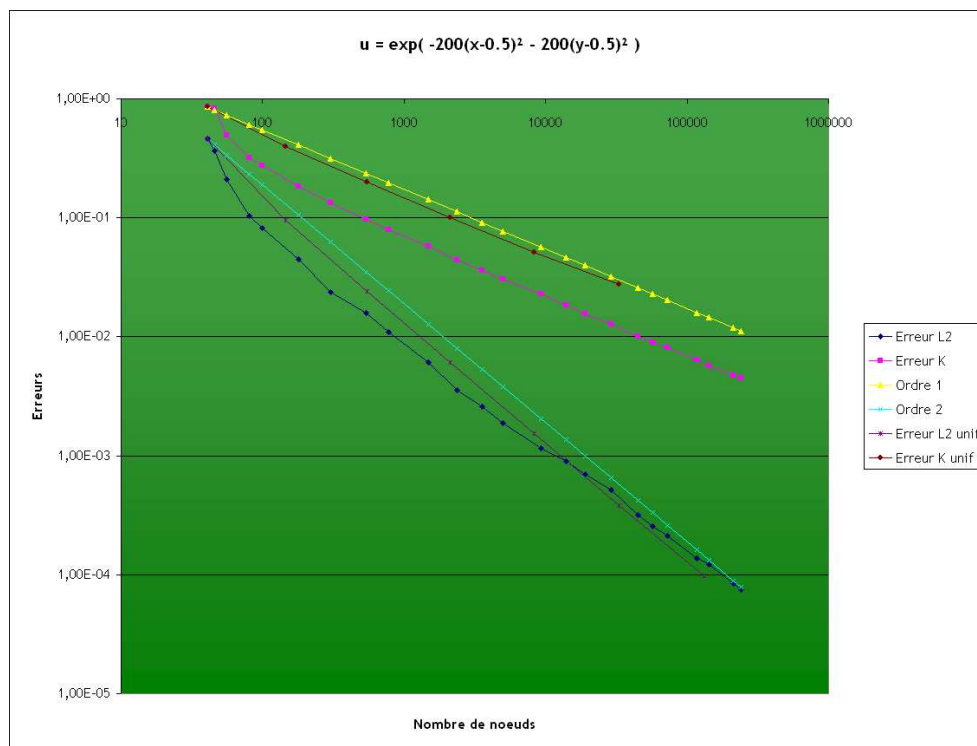


FIG. 18 – Convergence des normes

On peut remarquer que le maillage obtenu *FIG. 17* est tel qu'on l'attendait : centré sur le point $(0.5, 0.5)$, là où la fonction atteint son maximum, et restreint globalement au support de la fonction (domaine sur lequel la fonction est non nulle).

On aperçoit de plus un très bon comportement des erreurs, à la fois en norme d'énergie et en norme $L2$. Elles sont en effet toujours régulières, et inférieures, à nombre de noeuds égal, à leur homologue respective issu d'un raffinement de maillage uniforme.

La tendance que possède cette dernière courbe pour l'erreur $L2$, i.e. de se rapprocher de la courbe étalon d'ordre 2, est explicable par la manière dont les erreurs ont été calculées. En effet il a fallu approcher ces valeurs et le niveau de précision dans l'approximation des intégrales a varié avec le nombre de noeuds.

3.3.2 Nécessité de précision

Pour obtenir les résultats ci-dessus, il a fallu pouvoir calculer très précisément les intégrales pour le système linéaire. En effet une approximation grossière du second membre, $\int_{C_i} f$, ne permet pas de retrouver les ordres de décroissance. Le comportement de l'erreur relative peut être extrêmement irrégulier.

Afin d'évaluer assez précisément cette intégrale de f sur C_i (C_i étant le volume de contrôle associé au noeud i , cf. ANNEXE 2) peut être décomposée comme l'intégrale de f sur tous les $Q_{i,T} = C_i \cap T$. Les $Q_{i,T}$ seront alors eux-mêmes décomposés en 2 triangles, en considérant une "arête virtuelle" entre le sommet i et le centre de gravité de T . Aisément, une fonction récursive peut être utilisée afin d'approcher la valeur de l'intégrale sur ces triangles formant $Q_{i,T}$. Si l'aire du triangle est assez petite (comparaison avec une valeur rentrée par l'utilisateur : l'aire-palier), on effectue une quadrature d'ordre 1. En revanche si l'aire est supérieure à cette aire-palier, on applique la fonction récursive sur chacun des sous-triangles (cf. FIG. 19). Ainsi l'algorithme d'une telle fonction se présente ainsi :

x : tableau des abscisses des sommets du triangle
 y : tableau des ordonnées des sommets du triangle
 i_f : valeur de l'intégrale de f (initialisée à 0 avant le 1er appel)

Entrées : x, y, i_f

Sauvegarde dans des variables tampons des coordonnées

Calcul de l'aire du triangle A_T **SI** $A_T < \text{aire-palier}$

. Evaluation de f aux 3 sommets $\rightarrow f_1, f_2, f_3$

. Quadrature (ordre 1) pour l'intégrale : $i_f = i_f + \frac{A_T}{3} * (f_1 + f_2 + f_3)$

SINON

. Calcul des coordonnées des milieux des arêtes

. Appel récursif de la fonction sur chaque sous-triangle

. Réattribution des bonnes coordonnées pour x et y

FIN SI

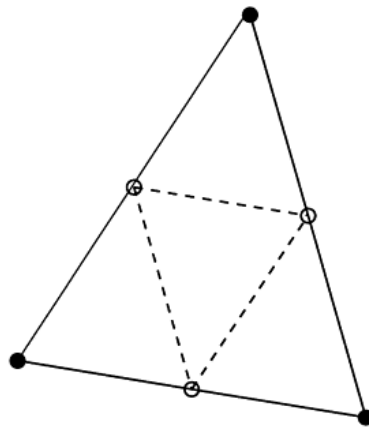


FIG. 19 – Si l'aire du triangle n'est pas assez petite, on le "découpe" en 4

Procédure : cette fonction sera utilisée en parcourant l'ensemble des triangles composant le maillage du domaine. On séparera le triangle courant en 3 quadrilatères, soit en 6 sous-triangles sur lesquels la fonction récursive sera appelée. On précise que la dernière étape de la fonction récursive, à savoir la réattribution des bonnes coordonnées, est indispensable si à la suite de la fonction récursive, un accès aux tableaux x et y doit être fait.

Après la définition de cet outil, nous pouvons illustrer l'influence de l'approximation de f , et donc l'approximation de son intégrale, sur l'erreur relative $L2$.

Prenons la fonction $u(x, y) = \exp(-200(x - \frac{1}{2})^2 - 200(y - \frac{1}{2})^2)$. Nous allons considérer les approximations suivantes :

- f est constante sur chaque volume de contrôle C_i
- f est d'ordre 1 sur chaque triangle composant le quadrilatère $Q_{i,T}$
- Utilisation d'une aire-palier égale à 10^{-4}
- Utilisation d'une aire-palier égale à 10^{-8}

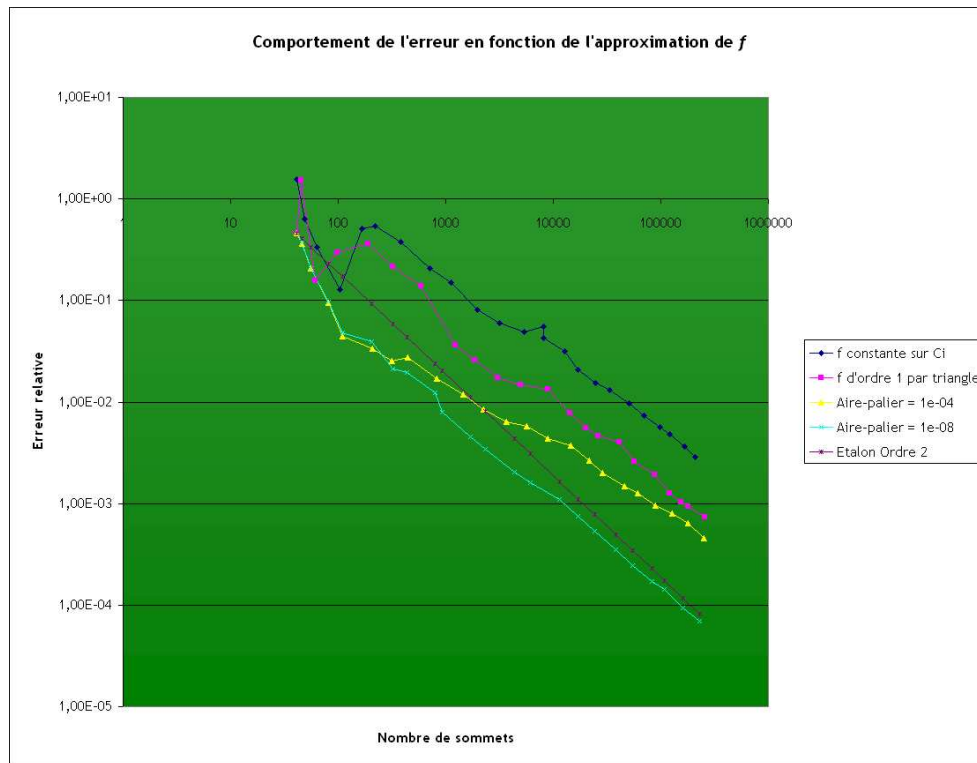


FIG. 20 – Influence de l'approximation de f

La FIG. 20 nous permet de nous rendre compte de l'importance de l'approximation de l'intégrale de f sur la précision du résultat.

La première méthode qui consiste à considérer que f est constante sur le volume de contrôle C_i (méthode initialement implémentée dans le code) entraîne un comportement irrégulier de l'erreur relative. Cette dernière est même supérieure à 1 au début et augmente à plusieurs reprises alors qu'on raffine. On peut même rajouter qu'une mauvaise approximation amène à des valeurs négatives de la solution numérique, qui doit représenter une exponentielle. Cela est un élément important à prendre en compte, notamment pour la simulation du transport de radionucléides où on peut avoir des activités négatives. Une meilleure ap-

proximation du terme source peut améliorer le code de ce point de vue là.

Avec l'amélioration de l'approximation de l'intégrale de f , on constate une amélioration aussi du comportement de l'erreur : en choisissant une aire-palier de 10^{-8} , l'erreur relative de la solution numérique décroît toujours aux cours des raffinements.

Bien évidemment une telle approximation n'est possible que si le second membre est connu analytiquement. Si jamais ce n'est pas le cas et que f est connu seulement aux noeuds du maillage, il peut être alors intéressant de considérer le fait que f soit affine par triangle plutôt que constante par volume de contrôle. En effet c'est l'approximation simple la plus forte qu'on puisse faire, et qui ne nécessite pas de grandes modifications au niveau de l'implémentation, pour un résultat nettement meilleur.

On peut aussi regarder l'influence de l'aire-palier sur la précision du calcul. Pour cela nous allons considérer la fonction $u(x, y) = 16x(1 - x)y(1 - y)$ et calculer l'erreur relative entre l'intégrale de f sur le domaine $[0, 1] \times [0, 1]$ et la somme cumulée des intégrales de f sur les volumes de contrôle C_i par la fonction récursive. La FIG. 21 illustre bien le comportement de l'erreur, qui décroît de manière assez linéaire avec l'aire-palier.

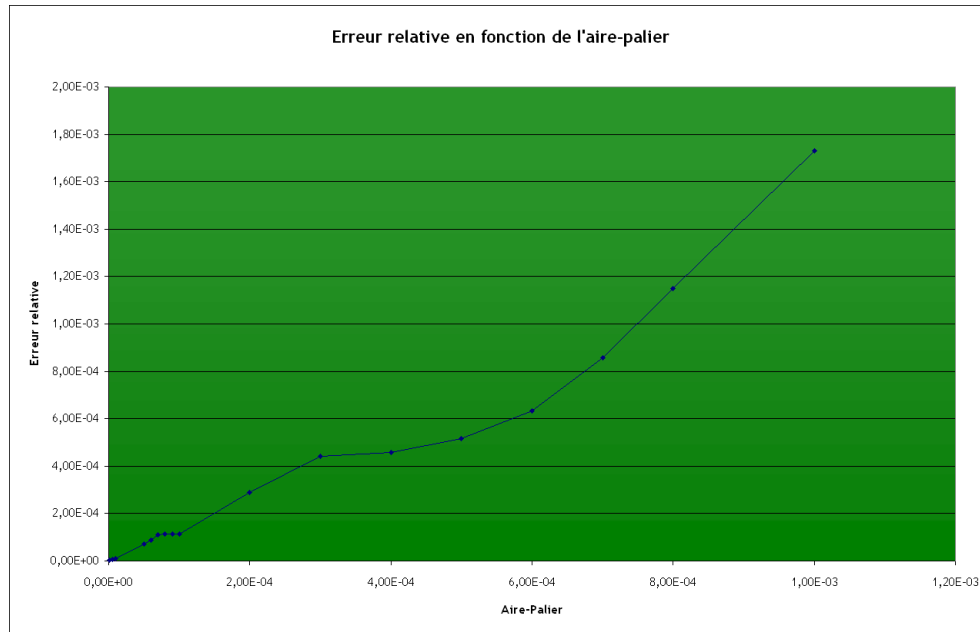


FIG. 21 – Influence de l'aire-palier

C'est cette même fonction récursive qui est utilisée pour le calcul des erreurs. Adaptée, elle prend alors en argumant les valeurs de la solution numérique aux sommets.

L'aire permettant d'obtenir les courbes dans la partie précédente a été prise égale à 10^{-8} . Cela signifie qu'il faut un nombre conséquent de calculs pour d'une part l'approximation de \int_{C_i} avant la résolution du système linéaire, et d'autre part pour le calcul des erreurs. La machine sur laquelle le stage a été effectué, et donc les tests, disposait de 4 coeurs et 8 processeurs. Cela a encouragé à paralléliser ces opérations.

3.3.3 Parallélisation de fonctions

Les caractéristiques techniques de la machine, le langage de programmation, le calcul sur des triangles de manière indépendante : tous les éléments étaient réunis pour développer un calcul de ces intégrales, $\int f$ et calcul des erreurs K et $L2$, grâce à *OpenMP* en parallélisant l'utilisation de cette fonction récursive définie précédemment.

Pour s'apercevoir des gains de temps, nous allons considérer les 3 fonctions précédemment utilisées et nous allons tester les temps mis par rapport au nombre de *threads* (processus calculant indépendamment une partie du code).

$$- u(x, y) = 16x(1 - x)y(1 - y)$$

Pour vérifier la validité de la méthode, nous allons aussi calculer l'erreur relative obtenue entre le calcul analytique de $\int_{\Omega} f$ et la valeur cumulée des intégrales approchées, toujours avec une aire-palier de 10^{-8}). Voici les résultats obtenus :

Nbr Threads	Tps CPU (s)	Tps d'exécution (s)	Err relative
1	55.4	55.2	1.07 e-12
8	59.4	11.4	2.44 e-13
16	58.4	7.6	2.00 e-13
32	58.5	7.3	2.00 e-13
40	58.4	7.3	2.00 e-13
64	58.3	7.3	1.33 e-13
128	58.3	7.3	1.33 e-13

$$- u(x, y) = \sin((1 - x)(1 - y)) + (1 - x)^3(1 - y)^2$$

Nbr Threads	Tps CPU (s)	Tps d'exécution (s)
1	230.0	229.9
8	241.9	45.5
16	242.3	31.0
32	243.9	30.8
40	242.2	30.8
64	242.3	30.4
128	242.2	30.4

$$- u(x, y) = \exp(-200(1 - x)^2 - 200(1 - y)^2)$$

Nbr Threads	Tps CPU (s)	Tps d'exécution (s)
1	199.4	200.1
8	209.6	39.6
16	210.0	27.0
32	210.1	26.7
40	210.2	26.7
64	210.3	26.5
128	210.2	26.5

On remarque globalement le même comportement quelle que soit la fonction utilisée : un palier apparaît à partir de 32 *threads* utilisés et on obtient un même coefficient du gain de temps, soit 7,5 environ. Ce coefficient est bien sûr à mettre en rapport avec le nombre de processeurs, 8 dans notre cas. On peut aussi noter qu'il vaut mieux choisir un nombre de *threads* supérieur au nombre de processeurs, et non pas égal comme on pourrait le penser initialement. Concernant l'erreur pour la première fonction, un comportement étrange apparaît : plus le nombre de *threads* est élevé, plus l'erreur est petite. On peut penser que cela est dû au fait que les erreurs sur l'approximation numérique se cumulent moins. Quoi qu'il en soit, avec une aire-palier de 10^{-8} , l'erreur commise sur l'approximation de $\int f$ est négligeable.

Cette utilisation du calcul parallèle sera indispensable pour le transport : en effet non seulement le maillage changera, mais le temps interviendra lui aussi. Il va falloir à de nombreuses reprises évaluer $\int_{C_i} f$. La parallélisation va permettre d'obtenir les résultats bien plus rapidement.

4 Le transport de radionucléides

4.1 Description du problème

4.1.1 Equation considérée

Le transport d'un soluté non réactif en milieu poreux peut être modélisé, sur l'intervalle en temps $]0, T[$ et sur le domaine Ω , par l'équation de convection-réaction-diffusion suivante :

$$\begin{cases} \frac{\partial a}{\partial t} - \text{div}(\mathbf{D}\nabla a - \vec{u}a) + \lambda a = f & \text{dans } \Omega \times]0, T[\\ a = 0 & \text{sur } \Gamma \times]0, T[\\ a(\bullet, 0) = a_0(\bullet) & \text{dans } \Omega \end{cases}$$

avec :

a : activité du soluté

\vec{u} : vitesse de Darcy, obtenue grâce à la résolution de l'équation d'écoulement

λ : constante de décroissance radioactive

f : terme source

$\mathbf{D} = \alpha|\vec{u}| + \omega d * I_2$: tenseur de dispersion

d : coefficient de diffusion du soluté

ω : porosité du milieu

$\alpha = \begin{pmatrix} \alpha_L & 0 \\ 0 & \alpha_T \end{pmatrix}$: tenseur de dispersivité (longitudinale et transversale par rapport à \vec{u})

Pour le cadre traité, afin que l'on puisse appliquer le théorème des estimations *a posteriori* (3) de la partie suivante, les hypothèses faites sur le modèle sont les suivantes :

(H1) Les grandeurs autres que l'activité sont considérées comme constantes en temps

(H2) Implication immédiate de H1, $\mathbf{D} = \mathbf{D}(x)$ et $\vec{u} = \vec{u}(x)$

(H3) \mathbf{D} est symétrique défini positif

(H4) $f \in L^2(0, T; \Omega)$

(H5) $a_0 \in H^1(\Omega)$

Nous choisirons d'appliquer un schéma implicite lors de la discrétisation en temps. En effet s'il ne possède pas la précision de Crank-Nicholson, il s'affranchit de toute CFL. L'intervalle de temps $]0, T]$ sera discrétisé et les $(t_i)_{i=0..N}$ formeront cette discrétisation. On notera $\tau_i = t_i - t_{i-1} \forall i \in 1..N$. Le code de calcul retournera alors les différents u_h^i , approximation affine par triangle de la solution au temps t_i .

La solution numérique $u_{h,\tau}$ sera construite à partir des u_h^i par linéarité en temps, i.e. à l'instant t tel que $t_{i-1} < t < t_i$, la fonction $u_{h,\tau}$ sera définie par la relation suivante :

$$u_{h,\tau}(t) = u_h^i - \frac{t_i - t}{\tau_i}(u_h^i - u_h^{i-1}) \quad (2)$$

ou bien, de manière équivalente, par

$$u_{h,\tau}(t) = u_h^{i-1} - \frac{t - t_{i-1}}{\tau_i}(u_h^i - u_h^{i-1})$$

4.1.2 Estimateur *a posteriori*

Les estimateurs présentés dans cette partie sont issus du rapport final de mon prédécesseur (réf. [1]), basé sur les travaux [2] et [3].

On définit le saut à travers une arête E par : $[\bullet.\vec{n}_E]$ et la norme d'énergie $[[\bullet]]$ par :

$$[[u]]^2(t) = \|u(t)\|_{L^2(\Omega)}^2 + \int_0^t (\|\mathbf{D}^{1/2}\nabla u(s)\|_{L^2(\Omega)}^2 + \lambda\|u(s)\|_{L^2(\Omega)}^2)ds$$

On a alors les quantités suivantes (adaptées au schéma implicite), et le théorème des estimations *a posteriori* lié à l'équation considérée :

– Estimateur spatial :

Définissons les estimateurs résiduel et de sauts inter-éléments par :

$$R_h^n|_Q = f_h^n - \frac{a_h^n - a_h^{n-1}}{\tau_n} + \text{div}(\mathbf{D}_h^n \nabla a_h^n - \vec{u}^n a_h^n)$$

$$r_h^n|_E = [\mathbf{D}_h^n \nabla a_h^n . \vec{n}_E]$$

$$z_h^n|_\gamma = \vec{u}^{n-1} . \vec{n}_\gamma (a_h^{n-1}(x_i) - a_h^{n-1}(x_j))$$

Les indicateurs locaux de l'erreur en espace sont alors définis par :

$$(\eta_R^n)^2 = \sum_{V \in \mathcal{V}} \sum_{Q \in V} \alpha_Q^2 \|R_h^n\|_{L^2(Q)}^2$$

$$(\eta_r^n)^2 = D_{\min}^{-1/2} \sum_{E \in \mathcal{E}} \alpha_E \|r_h^n\|_{L^2(E)}^2$$

$$(\eta_z^n)^2 = D_{\min}^{-1/2} \sum_{\gamma \in \Gamma_h^n} \alpha_\gamma \|z_h^n\|_{L^2(\gamma)}^2$$

où $\alpha_S = \min(h_S D_{\min}^{-1/2}, \lambda^{-1/2})$ pour $S = Q, E$ ou γ .

L'estimateur spatial est finalement donné par :

$$(\eta^n)^2 = (\eta_R^n)^2 + (\eta_r^n)^2 + (\eta_z^n)^2$$

– Estimateur temporel :

$$(\theta^n)^2 = \frac{\tau_n}{3} (\|(\mathbf{D}_h^n)^{1/2} \nabla (a_h^n - a_h^{n-1})\|_{L^2(\Omega)}^2 + \|\text{div}(\vec{u}^n (a_h^n - a_h^{n-1}))\|_{L^2(\Omega)}^2 + \|\lambda^{1/2} (a_h^n - a_h^{n-1})\|_{L^2(\Omega)}^2)$$

– Théorème :

$$[[u - u_{h,\tau}]](t_n) \leq C * \left(\sum_{m=1}^n \{(\eta^m)^2 \tau_m + (\theta^m)^2\} \right)^{1/2} \quad (3)$$

4.2 Maillage adaptatif

4.2.1 Algorithme

Les premiers tests effectués dans le cadre du stage, les questions de mon maître de stage sur le rôle de la tolérance notamment, et la lecture de la documentation du travail de mon prédécesseur, m'ont amené à entrer dans le code afin de vérifier l'implémentation des diverses fonctions ainsi que l'algorithme. Légèrement modifié par rapport au précédent, voici l'algorithme, adapté de [3], qui est dorénavant utilisé :

N_{sp}/N_{tm}		<i>nombre maximum de raffinements en espace/temps</i>
t_{n-1}	$= t_0$	<i>temps antérieur au calcul</i>
t_n	$= t_{n-1} + \tau$	<i>temps du calcul</i>
Tol_h^n	$= 0$	<i>tolérance spatiale, définition page suivante</i>
Tol_t^n	$= 0$	<i>tolérance temporelle, définition page suivante</i>
$(\eta^n)^2$	$= Tol_h^n + 1$	<i>estimateur spatial</i>
n_{sp}	$= 0$	<i>nombre de raffinements en espace effectués</i>
n_{tm}	$= 0$	<i>nombre de raffinements en temps effectués</i>

TANT QUE $t_n < t_N$

. *Estimateur spatial*

. **TANT QUE** $n_{sp} < N_{sp}$ et que $(\eta^n)^2 > Tol_h^n$

. . Résolution du système linéaire $\rightarrow u_{h,\tau}(t_n)$

. . Calcul de la tolérance spatiale Tol_h^n et de la tolérance temporelle Tol_t^n

. . (*cf. page suivante pour la définition et le détail des calculs des tolérances*)

. . Calcul de l'estimateur spatial η^n

. . **SI** $\eta^n > Tol_h^n$

. . . Stratégie de raffinement en espace, $n_{sp} = n_{sp} + 1$

. . **FIN SI**

. **FIN TANT QUE**

. *Estimateur temporel*

. Calcul de l'estimateur temporel θ^n

. **SI** $n_{tm} < N_{tm}$ et que $\theta^n > Tol_t^n$

. . Raffinement en temps :

. . $\tau = \tau/2$

. . $n_{tm} = n_{tm} + 1$

. **SINON SI** $\theta^n \leq Tol_t^n$

. . On passe alors au pas de temps suivant avec une accélération :

. . $\tau = 2 * \tau$

. . $t_{n-1} = t_n$

. . $n_{tm} = n_{tm} - 1$

. **SINON**

. . *nombre maximal de raffinements temporels atteint*

. . On passe au pas de temps suivant par défaut : pas d'accélération

. . $t_{n-1} = t_n$

. **FIN SI**

. $t_n = t_{n-1} + \tau$

FIN TANT QUE

4.2.2 Détails de l'implémentation

Nous verrons dans cette partie l'implémentation des différents termes nécessaires à la mise en place du raffinement adaptatif.

– Définition et calcul des tolérances :

Comme il a été précisé par l'équation (2), $u_{h,\tau}$ est linéaire en temps et définie entre 2 instants de discrétisation t_{i-1} et t_i . Ainsi, les fonctions $v_1 := u_{h,\tau}^2$ et $v_2 := (\mathbf{D}_h \nabla u_{h,\tau}, \nabla u_{h,\tau})$ sont d'ordre 2 en temps. On peut expliciter ces 2 fonctions $\forall t \in [t_{i-1}, t_i]$ (en tenant compte du fait que le tenseur discrétisé \mathbf{D}_h est symétrique) :

$$\begin{aligned} v_1(t) &= (u_h^i)^2 + \left(\frac{t_i - t}{\tau_i} (u_h^i - u_h^{i-1}) \right)^2 - 2 \frac{t_i - t}{\tau_i} (u_h^i - u_h^{i-1}) \\ v_2(t) &= (\mathbf{D}_h \nabla u_h^i, \nabla u_h^i) - 2 \frac{t_i - t}{\tau_i} (\mathbf{D}_h \nabla u_h^i, \nabla (u_h^i - u_h^{i-1})) \\ &\quad + \left(\frac{t_i - t}{\tau_i} \right)^2 (\mathbf{D}_h \nabla (u_h^i - u_h^{i-1}), \nabla (u_h^i - u_h^{i-1})) \end{aligned}$$

On intègre chacune de ces deux fonctions sur $[t_{i-1}, t_i]$ et on obtient les résultats suivants :

$$\begin{aligned} \int_{t_{i-1}}^{t_i} v_1(t) dt &= \frac{t_i - t_{i-1}}{3} ((u_h^i)^2 + (u_h^{i-1})^2 + u_h^i * u_h^{i-1}) \\ \int_{t_{i-1}}^{t_i} v_2(t) dt &= \frac{t_i - t_{i-1}}{3} [(\mathbf{D}_h \nabla u_h^i, \nabla u_h^i) + (\mathbf{D}_h \nabla u_h^{i-1}, \nabla u_h^{i-1}) + (\mathbf{D}_h \nabla u_h^i, \nabla u_h^{i-1})] \end{aligned}$$

On utilisera ces derniers résultats pour le calcul de la tolérance. En effet on a :

$$\begin{aligned} [[u_{h,\tau}]^2(t_n) - \|u_{h,\tau}(t_n)\|_{L^2(\Omega)}^2] &= \int_0^{t_n} \left(\|\mathbf{D}_h^{1/2} \nabla u_{h,\tau}(s)\|_{L^2(\Omega)}^2 + \lambda \|u_{h,\tau}(s)\|_{L^2(\Omega)}^2 \right) ds \\ &= I(t_n) \end{aligned}$$

Décomposons alors $I(t_n)$:

$$\begin{aligned} I(t_n) &= \sum_{m=1}^n \int_{t_{m-1}}^{t_m} \left(\|\mathbf{D}_h^{1/2} \nabla u_{h,\tau}(s)\|_{L^2(\Omega)}^2 + \lambda \|u_{h,\tau}(s)\|_{L^2(\Omega)}^2 \right) ds \\ &= \sum_{m=1}^n \int_{\Omega} \int_{t_{m-1}}^{t_m} (v_2(s) + \lambda v_1(s)) ds \\ &= \sum_{m=1}^n \frac{\tau_m}{3} \{ (\|u_h^m\|_{L^2(\Omega)}^2 + \|u_h^{m-1}\|_{L^2(\Omega)}^2 + \int_{\Omega} u_h^m u_h^{m-1}) \lambda \\ &\quad + \|u_h^m\|_K^2 + \|u_h^{m-1}\|_K^2 + \int_{\Omega} (\mathbf{D}_h \nabla u_h^m, \nabla u_h^{m-1}) \} \\ &= \sum_{m=1}^n \alpha_m \end{aligned}$$

A partir d'une précision souhaitée ϵ , on peut définir la tolérance temporelle Tol_t^m et la tolérance spatiale Tol_h^m , pour tout temps t_m de la discrétisation temporelle, par :

$$Tol_t^m = \epsilon^2 * \frac{\alpha_m}{2}$$

$$Tol_h^m = \epsilon^2 * \frac{\alpha_m}{2\tau_m}$$

Si on arrive à avoir $(\eta^m)^2 \leq Tol_h^m$ et $(\theta^m)^2 \leq Tol_t^m$, alors $\forall m \in 1..n$:

$$\begin{aligned} (\eta^m)^2 * \tau_m + (\theta^m)^2 &\leq \epsilon^2 * \left(\frac{\alpha_m}{2\tau_m} \tau_m + \frac{\alpha_m}{2} \right) \\ &\leq \epsilon^2 * \alpha_m \end{aligned}$$

Donc en sommant :

$$\begin{aligned} \sum_{m=1}^n ((\eta^m)^2 * \tau_m + (\theta^m)^2) &\leq \epsilon^2 \sum_{m=1}^n \alpha_m \\ &\leq \epsilon^2 * I(t_n) \\ &\leq \epsilon^2 * I(t_n) + \epsilon^2 \|u_{h,\tau}\|_{L^2\Omega}^2 \\ &\leq \epsilon^2 * [[u_{h,\tau}]]^2(t_n) \end{aligned}$$

Et en utilisant finalement le théorème (3) :

$$[[u - u_{h,\tau}]]^2(t_n) \leq C \left(\sum_{m=1}^n ((\eta^m)^2 * \tau_m + (\theta^m)^2) \right) \leq C * \epsilon^2 * [[u_{h,\tau}]]^2(t_n)$$

ou en d'autres termes :

$$\frac{[[u - u_{h,\tau}]](t_n)}{[[u_{h,\tau}]](t_n)} \leq C * \epsilon$$

A chaque pas de temps t_m , il faudra calculer $\int_{\Omega} u_h^m u_h^{m-1}$, $\int_{\Omega} (\mathbf{D}_h \nabla u_h^m, \nabla u_h^{m-1})$, $\|u_h^m\|_{L^2(\Omega)}^2$ et $\|\mathbf{D}_h \nabla u_h^m, \nabla u_h^m\|_{L^2(\Omega)}^2$. On gardera en mémoire ces 2 dernières normes au carré afin d'éviter leur calcul au pas de temps suivant.

Si seul α_m est utile pour définir les tolérances spatiale et temporelle à un temps t_m , on gardera néanmoins dans le cadre des solutions analytiques la quantité $\sum \alpha_m$, qu'on transmettra au pas de temps suivant afin de pouvoir calculer l'erreur relative en norme d'énergie à tous les temps.

Les intégrales sur Ω précédemment citées seront bien entendu calculées à partir des intégrales sur les triangles composant le maillage, les deux fonctions, u_h^m et u_h^{m-1} , étant affines par sur chacun de ces éléments.

Dans la définition des tolérances, le coefficient pris de 1/2 est arbitraire. Il permet de pouvoir équilibrer les contributions liées à l'estimateur spatial et à l'estimateur temporel.

– Estimateurs temporel et spatiaux

Les différences rencontrées et donc les changements apportés au code initial sont visibles en annexe (ANNEXE 5). Une comparaison détaillée entre ce qui avait été programmé et la correction apportée, sur le calcul de l'estimateur résiduel et la théorie des éléments de référence notamment, y a été réalisée.

L'implémentation des estimateurs spatiaux est similaire à ce qui a pu être décrit pour l'écoulement permanent, à la différence près qu'il est nécessaire d'avoir les valeurs du temps précédent.

Concernant l'estimateur temporel, on y faisait appel que selon une fréquence donnée. Si nous voulons respecter le critère d'arrêt tel qu'il est décrit dans la description de l'algorithme, par rapport à la précision souhaitée, cet estimateur temporel doit être calculé à chaque pas de temps. C'est lui qui permet de décider si le raffinement en temps doit avoir lieu.

Le calcul de cet estimateur est réalisé dans "ConcErrTime2.f". Les modifications apportées à cette fonction étaient légères : il s'agissait principalement d'un nettoyage du code et de regrouper les éléments déjà calculés et donc gardés en mémoire (vitesse de Darcy, tenseur...) plutôt que de refaire les calculs.

Finalement on peut rappeler la parallélisation effectuée, via *OpenMP*, pour le calcul des estimateurs, de manière similaire à ce qui a pu être fait dans le cadre de l'écoulement permanent.

4.3 Comparaisons avec raffinement uniforme de maillage

CETTE PARTIE EST EN COURS DE TRAVAIL A LA REMISE DE CETTE 1ERE VERSION DE RAPPORT

4.3.1 Utilisation d'une solution analytique

Nous allons nous placer sur le domaine $[0, 30] \times [0, 20]$. Nous utiliserons la fonction suivante pour la solution de l'équation de transport :

$$u(x, y, t) = \exp\left(-\frac{1}{4}(x - 10 - t)^2 - \frac{1}{2}(y - 10)^2\right)$$

Parce qu'il s'agit d'un dôme toujours positif, cette fonction parait en effet particulièrement adaptée à la modélisation d'une activité radioactive (dépôt de colis + migration d'une quantité de polluant).

IMAGE A METTRE

L'erreur sera calculée en norme d'énergie. Pour cela il faudra calculer l'intégrale suivante :

$$\begin{aligned} I_e(t_n) &= \int_0^{t_n} (\|(u - u_{h,\tau})(s)\|_K + \|(u - u_{h,\tau})(s)\|_{L^2(\Omega)}) ds \\ &= \sum_{m=1}^n \int_{t_{m-1}}^{t_m} (\|(u - u_{h,\tau})(s)\|_K + \|(u - u_{h,\tau})(s)\|_{L^2(\Omega)}) ds \\ &= \sum_{m=1}^n I_{e,m} \end{aligned}$$

Contrairement à ce qui avait été fait pour définir et calculer les tolérances, un calcul exact est difficilement envisageable pour les $I_{e,m}$. On va donc approcher ces intégrales en discrétisant $[t_{m-1}, t_m]$, par exemple par les $(s_{m,i})_{i=0..M}$, où $s_{m,0} = t_{m-1}$ et $s_{m,M} = t_m$. On calculera alors pour chaque $s_{m,i}$ ($i \in 1..M$) les contributions $u(s_{m,i}) - u_{h,\tau}(s_{m,i}) = u(s_{m,i}) - u_h^m + \frac{t_m - s_{m,i}}{\tau_m} (u_h^m - u_h^{m-1})$, ainsi que les gradients respectifs, aux sommets du triangle T considéré. On effectuera la formule de quadrature simple pour avoir les normes K et $L2$ sur le triangle T (même fonction utilisée que pour le calcul des normes d'erreur dans le cas permanent, cf. p27, avec découpage du triangle en 4 si l'aire est trop grande). On multipliera chacune des normes par le pas de cette nouvelle discrétisation en temps, $\frac{t_m - t_{m-1}}{M}$ et on les additionnera, ce qui correspond à une approximation par la méthode des rectangles à gauche.

$$\begin{aligned}
I_e(t_n) &= \sum_{m=1}^n I_{e,m} \\
&= \sum_{m=1}^n \sum_{i=1}^M (s_{m,i} - s_{m,i-1}) (|(u - u_{h,\tau})(s_{m,i})|_K + \|(u - u_{h,\tau})(s_{m,i})\|_{L^2(\Omega)}) \\
&= \sum_{m=1}^n \sum_{i=1}^M \frac{\tau_m}{M} \sum_T (|(u - u_{h,\tau})(s_{m,i})|_{K(T)} + \|(u - u_{h,\tau})(s_{m,i})\|_{L^2(T)})
\end{aligned}$$

Les résultats ici sont mitigés.

En effet si l'estimateur temporel a un bon comportement (décroît lorsque le nombre de noeuds augmente, et donc devient à chaque temps inférieur à sa tolérance), l'estimateur temporel reste en revanche dans tous les cas inférieur à sa tolérance : on a donc toujours une accélération du pas de temps, mais jamais de raffinement en temps. Cela conduit à avoir un pas de temps final très grand, qui peut expliquer le fait que la précision fixée n'est pas atteinte au dernier pas de temps (alors qu'elle l'était à tous les pas précédents).

On peut aussi remarquer que la constante C du théorème (3) dépend du temps.

4.3.2 Cas de dépôt de colis dans un milieu hétérogène

Pour ce cas "réel", dans le sens où nous allons simuler un relâchement de radionucléides d'après les données de ces derniers et non plus utiliser une solution analytique, nous allons considérer des milieux ne possédant pas les mêmes propriétés physiques et étudier le comportement obtenu.

Nous n'allons plus pouvoir calculer la norme de l'erreur, puisque nous n'utiliserons pas de solution analytique. La précision obtenue par rapport à un raffinement de maillage uniforme sera donc évaluée de la sorte : nous allons considérer, à un temps donné, les valeurs des différents estimateurs issus d'un maillage adaptatif et nous allons chercher pour quel niveau de raffinement de maillage uniforme ces valeurs sont retrouvées. Nous comparerons alors le nombre de mailles nécessaires et nous pourrons déterminer ainsi le gain, ou la perte (les tests n'ayant pas encore été lancés à l'heure de remise de cette version du rapport), de "précision" obtenu(e) par l'utilisation de la méthode de raffinement adaptative.

5 Conclusion provisoire

Grâce à ce stage au sein de l'IRSN, j'ai pu approfondir mes connaissances en analyse numérique et me familiariser avec un langage de programmation très présent dans les codes de calcul industriels : FORTRAN. J'ai eu l'opportunité de mettre en oeuvre des algorithmes permettant un raffinement de maillages adaptatifs, basés sur les théorèmes mathématiques des estimations *a posteriori*, pour un logiciel utilisé dans un cadre professionnel. Ce stage aura donc été pour moi un premier aperçu du monde du travail, et plus particulièrement du domaine du développement de codes de calcul scientifiques, dans lequel je compte exercer et développer mes compétences dans ma carrière à venir.

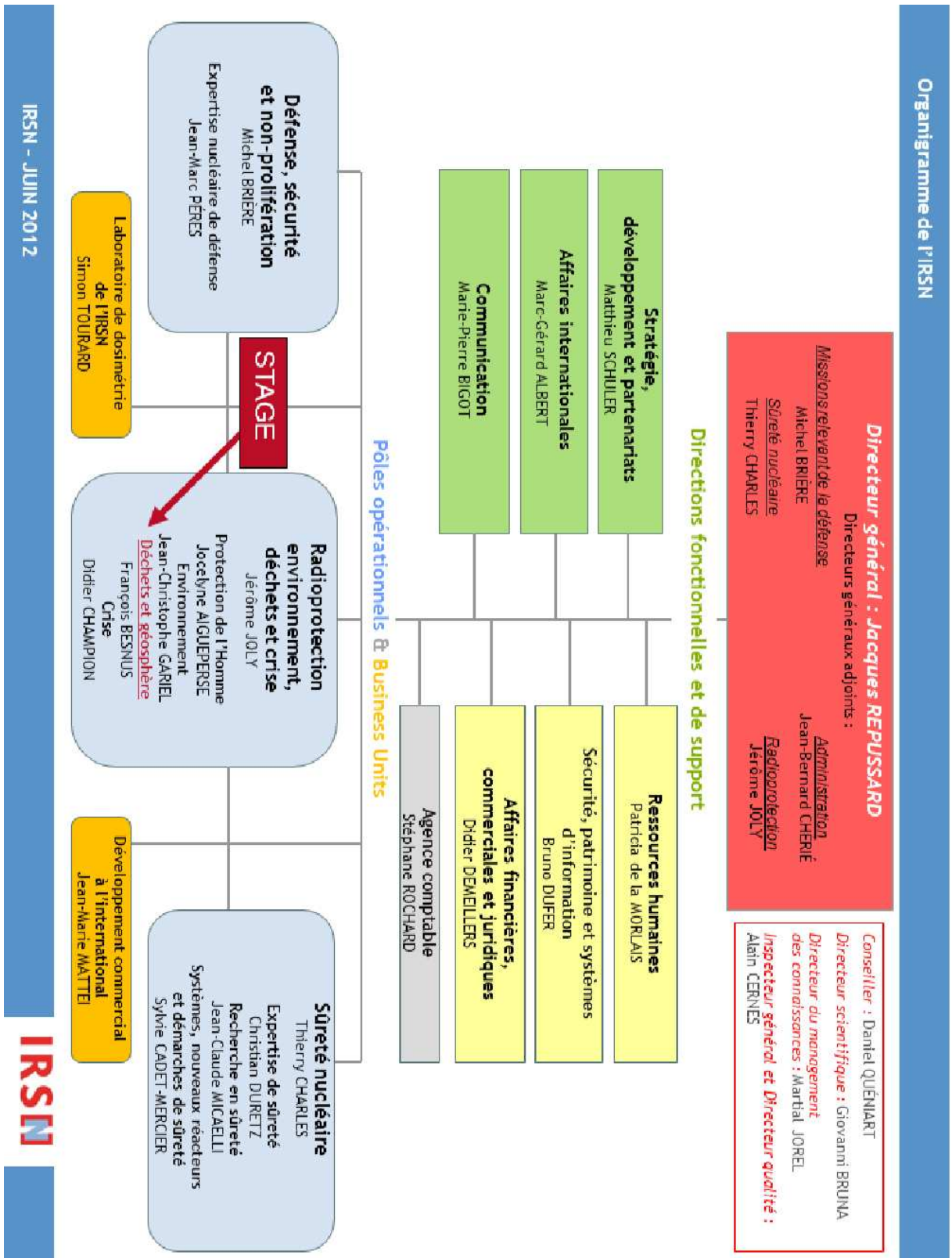
Si le sujet du stage initial, qui concernait l'étude des comportements du code et la recherche des paramètres optimaux d'utilisation de celui-ci, n'a pas, ou très peu, été abordé et par conséquent reste un problème entier, la correction des algorithmes et des fonctions mises en place par mon prédécesseur était nécessaire en vue de laisser un code de calcul fonctionnel, correct, propre et, je l'espère, compréhensible. J'ai en effet pu voir que la documentation était précieuse et avait besoin d'être précise, afin que les personnes qui prennent la suite d'un travail effectué puissent comprendre le plus rapidement et le plus clairement possible ce qui a été fait.

Les résultats obtenus par la mise en place de solutions analytiques ont été, globalement, satisfaisants et ont illustré le gain en précision sur l'erreur relative notamment, que permettait cette nouvelle méthode de raffinement mise en place, par rapport à des raffinements uniformes menés sur le maillage.

Plusieurs aspects peuvent être traités pour le travail à effectuer à la suite du stage. Tout d'abord les premiers résultats obtenus dans le cadre de l'équation de transport avec solution analytique ne sont pas entièrement satisfaisants. Il pourrait être intéressant de revoir la démonstration du théorème des estimations *a posteriori* liées à cette équation, voire d'en implémenter de nouvelles, comme par exemple celles données par [4]. Ensuite le sujet initial du stage pourrait être à nouveau évoqué. En effet si le gain en précision a pu être démontré - dans le cadre de l'équation d'écoulement pour le moment - celui en temps reste à étudier. La méthode des estimations *a posteriori* multipliant les assemblages de systèmes linéaires et leur résolution, il faut impérativement essayer de limiter ces derniers afin que le temps de calcul ne soit pas démesuré. Outre la mise en place d'une résolution de système linéaire par utilisation de calculs parallèles, on pourrait, et c'était l'axe de départ du stage, effectuer une analyse de sensibilité des paramètres, par exemple à l'aide d'indices de Sobol.

6 Annexes

6.1 Annexe 1 : Organigramme de l'IRSN



6.2 Annexe 2 : Equations gérées par *MELODIE*

L'écoulement permanent

L'équation générale de l'écoulement (4) a pour origine la loi de Darcy (5) et la loi de conservation de masse. Elle s'écrit de la manière suivante :

$$\text{div}(\mathbf{K}\nabla h) = S_s \frac{\partial h}{\partial t} + q \quad (4)$$

où les quantités sont définies par :

- S_s : coefficient d'emmagasinement
- h : charge hydraulique
- q : débit d'eau injectée ou prélevée par m^3
- \mathbf{K} : tenseur de perméabilité

Et \vec{U} la vitesse de Darcy est donnée par :

$$\vec{U} = -\mathbf{K}\nabla h \quad (5)$$

Le transport

Le transport d'un soluté dans un milieu poreux est décrit par l'équation de transport (6) qui relie la concentration du soluté et les différents phénomènes physiques liés au polluant :

$$\text{div}((\mathbf{D} - \omega d \mathbf{I})\nabla c_i - \vec{u} c_i) = \omega' R_i \frac{\partial c_i}{\partial t} + \lambda_i R_i \omega' c_i - \lambda_j R_j \omega' c_j \frac{M_i}{M_j} \quad (6)$$

avec :

- \vec{u} : vitesse de Darcy
- \mathbf{D} : tenseur de dispersion
- d : coefficient de diffusion
- \mathbf{I} : matrice identité correspondant à la dimension traitée
- ω : porosité totale
- ω' : porosité cinématique
- c_k : concentration volumique de l'élément k ($k = i$ ou j)
- R_k : coefficients de retard de l'élément k
- λ_k : constante de décroissance radioactive de l'élément k
- M_k : masse de l'élément k

On pourra transformer cette dernière équation du transport (6) en considérant non plus la concentration c d'un soluté mais son activité massique A définie par $A = \lambda \frac{N}{\rho M} c$ (avec ρ masse volumique de l'eau et N le nombre d'Avogadro). L'activité massique correspond au nombre de désintégration par seconde de l'élément par unité de masse.

La résolution des équations

La méthode de résolution des équations utilisée dans le cadre du stage aura été la méthode FVFE (*Finite Volumes Finite Elements*). Cette méthode nécessite notamment la création d'un maillage dual. Ce dernier sera obtenu en joignant le centre de gravité de chaque triangle avec les milieux des arêtes le composant (on peut aussi choisir de considérer le point de concours des médiatrices plutôt que le centre de gravité). On crée ainsi un volume de contrôle C_P pour chaque noeud P du maillage (cf. FIG. 22).

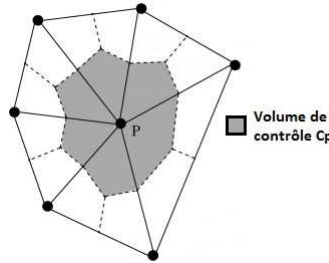


FIG. 22 – Représentation du maillage dual et d'un volume de contrôle

Considérons l'équation de transport satisfaite par un soluté non réactif :

$$\text{div}(\mathbf{D}\nabla A - \vec{u}A) = \omega \frac{\partial A}{\partial t} + q \quad (7)$$

On appelle N_j les fonctions de base utilisées dans le développement de Galerkin et définies de la manière suivante :

$$\begin{cases} N_j(x_j) &= 1 \\ N_j(x_i) &= 0 \\ N_j(x) &\text{linéaire entre } x_j \text{ et } x_i \end{cases}$$

Alors en utilisant un développement de Galerkin et en intégrant sur C_i l'équation de transport (7), on obtient pour chaque volume C_i et après intégration par parties :

$$\int_{\partial C_i} \mathbf{D} \left(\sum_j \nabla N_j A_j \right) \cdot \vec{n} d\sigma - \int_{\partial C_i} \left(\vec{u} \sum_j \nabla N_j A_j \right) \cdot \vec{n} d\sigma = \int_{C_i} \omega \frac{\partial \sum_j N_j A_j}{\partial t} dx + \int_{C_i} q dx$$

où \vec{n} représente la normale sortante au contour ∂C_i .

L'approximation du terme temporel est faite à l'aide d'un schéma d'Euler :

$$\int_{C_i} \frac{\partial A}{\partial t} dx \simeq \sum_{T:i \in T} \sum_j \frac{A^{n+1} - A^n}{t_{n+1} - t_n} \int_{C_i \cap T} N_j dx$$

L'approximation du flux diffusif se fera sur un schéma implicite :

$$\int_{\partial C_i} \mathbf{D} \left(\sum_j \nabla N_j A_j \right) \cdot \vec{n} d\sigma \simeq \sum_{T:i \in T} \int_{C_i \cap T} \mathbf{D}_T \sum_j \nabla N_j A_j^{n+1} \cdot \vec{n} d\sigma$$

où \mathbf{D}_T est une approximation du tenseur de diffusion-dispersion sur T .

Le traitement du flux convectif sera effectué sur des schémas d'approximation de type Godunov semi-implicite ou implicite :

$$\int_{\partial C_i} \left(\vec{u} \sum_j \nabla N_j A_j \right) \cdot \vec{n} d\sigma \simeq$$

6.3 Annexe 3 : Théorie des éléments de référence

Le calcul de l'estimateur résiduel (estimateur spatial) se fait sur les $Q = C_i \cap T$, qui sont des quadrangles. Une méthode pour calculer la norme L^2 sur ces quadrangles peut être de passer par des éléments de référence, et en l'occurrence le carré \hat{Q} formé par les points $\hat{A}_1(-1, -1)$, $\hat{A}_2(1, -1)$, $\hat{A}_3(1, 1)$ et $\hat{A}_4(-1, 1)$.

La transformation F_Q permettant de passer de \hat{Q} à Q est définie par :

$$F_Q \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \text{ et } x = \sum_{i=1}^4 x_i \hat{\phi}_i(\hat{x}, \hat{y}), y = \sum_{i=1}^4 y_i \hat{\phi}_i(\hat{x}, \hat{y})$$

avec

$$\begin{aligned} \hat{\phi}_1(\hat{x}, \hat{y}) &= \frac{1}{4}(1 - \hat{x})(1 - \hat{y}), \\ \hat{\phi}_2(\hat{x}, \hat{y}) &= \frac{1}{4}(1 + \hat{x})(1 - \hat{y}), \\ \hat{\phi}_3(\hat{x}, \hat{y}) &= \frac{1}{4}(1 + \hat{x})(1 + \hat{y}), \\ \hat{\phi}_4(\hat{x}, \hat{y}) &= \frac{1}{4}(1 - \hat{x})(1 + \hat{y}) \end{aligned}$$

et où les x_i, y_i sont les coordonnées des A_i , points du quadrangle Q .

La matrice jacobienne de F_Q est alors donnée par :

$$B_Q = \begin{pmatrix} \frac{\partial x}{\partial \hat{x}} & \frac{\partial x}{\partial \hat{y}} \\ \frac{\partial y}{\partial \hat{x}} & \frac{\partial y}{\partial \hat{y}} \end{pmatrix}$$

Il est suffisant d'avoir un déterminant non nul pour B_Q afin que F_Q soit inversible. Cette condition est notamment satisfaite si Q est convexe, ce qui sera toujours notre cas. On a alors :

$$|\det(B_Q)(\hat{x}, \hat{y})| = \frac{1}{16} | \{ (x_2 - x_1)(1 - \hat{y}) + (x_3 - x_4)(1 + \hat{y}) \} * \{ (y_4 - y_1)(1 - \hat{x}) + (y_3 - y_2)(1 + \hat{x}) \} - \{ (y_2 - y_1)(1 - \hat{y}) + (y_3 - y_4)(1 + \hat{y}) \} * \{ (x_4 - x_1)(1 - \hat{x}) + (x_3 - x_2)(1 + \hat{x}) \} |$$

En posant $\hat{u} := u(F_Q)$, on a $\forall X \in Q, \forall \hat{X} \in \hat{Q}$:

$$u(X) = u(F_Q(\hat{X})) = \hat{u}(\hat{X})$$

Donc en particulier $\hat{u}(\hat{A}_i) = u(A_i)$ pour $i = 1..4$

La méthode des éléments de référence nous permet d'avoir, avec $\hat{f} := \hat{u} \sqrt{|\det(B_Q)|}$:

$$\begin{aligned} \int_Q u^2(X) dX &= \int_{F_Q(\hat{Q})} u^2(X) dX \\ &= \int_{\hat{Q}} u^2(F_Q(\hat{X})) |\det(B_Q)(\hat{X})| d\hat{X} \\ &= \int_{\hat{Q}} \hat{u}^2(\hat{X}) |\det(B_Q)(\hat{X})| d\hat{X} \\ &= \int_{\hat{Q}} \hat{f}^2(\hat{X}) d\hat{X} \end{aligned}$$

La quadrature qu'on devait réaliser sur Q peut alors être réalisée sur \hat{Q} à l'aide des valeurs de \hat{f} évaluées aux sommets \hat{A}_i pour $i = 1..4$ (il reste donc à évaluer le déterminant de la Jacobienne puisque les valeurs de \hat{u} aux \hat{A}_i sont les mêmes que celles de u aux A_i).

6.4 Annexe 4 : Changements apportés pour l'écoulement permanent

- Comme on l'a déjà vu, les contributions données par r_T et r_E peuvent être calculées dans 2 boucles différentes : on évite ainsi de devoir refaire pour chaque triangle les calculs des quantités (tenseur, aire, gradient,...) propres à ses voisins.
- L'estimateur résiduel $r_T(u_h) = f + \text{div}(\mathbf{K}\nabla u_h) = f$ ne peut pas être tiré du système linéaire. En effet le système linéaire représente l'intégrale de l'équation, donc $\int_{C_i} f$, alors que nous avons besoin de la norme $L2$ de r_T , donc $\int_T f^2$.
- Les calculs des estimateurs résiduel et des sauts ne sont pas bien réalisés. Même si f n'est connu que ponctuellement, i.e. que le calcul de l'intégrale n'est pas possible exactement et qu'il faut par conséquent utiliser une formule de quadrature, les calculs implémentés ne sont pas bons. Le descriptif des calculs et des formules de quadrature corrects peut être trouvé dans la partie sur l'implémentation du calcul de l'estimateur (Page à préciser).

Ce qui avait été rentré pour l'approximation de la norme $L2$ de r_T (outre le fait que r_T n'est pas le résidu du système linéaire) est de la forme :

$$\|r_T\|_{L^2(T)}^2 = \int r_T^2 \simeq \left(\frac{A_T}{3}\right)^2 * (r_1 + r_2 + r_3)^2 \quad (8)$$

Cette approximation est fausse. On peut le constater en prenant une fonction constante : la norme $L2$ (au carré) d'une fonction constante $r_T := 1$ est donc égale à l'aire du triangle A_T , tandis que par la formule A indiquer, cela donnerait A_T^2 . On utilisera la quadrature des points milieux, ou formule de Hammer, donnée page A INDIQUER et exacte pour un polynôme d'ordre 2.

Concernant la formule utilisée pour le calcul de la norme $L2$ de r_E (1.249 à 254), elle s'avère elle aussi inexacte. $xgradh$ représente la première composante du vecteur $\mathbf{K}^T \nabla u_{h,T}$ tandis que $ygradh$ représente la seconde. La formule à appliquer est alors la différence entre les vecteurs $\mathbf{K}^T \nabla u_{h,T}$ et $\mathbf{K}^{T'} \nabla u_{h,T'}$ qui retourne donc un vecteur, dont on calcule le produit scalaire avec la normale, entrante ou sortante, cela n'a pas d'importance puisque finalement on prend le carré de ce produit scalaire. On doit donc avoir :

$$((xgradh - xgradh') * xnorm + (ygradh - ygradh') * ynorm)^2$$

Autrement dit il s'agit du carré du produit scalaire (carré de somme) et non pas la somme des termes du produit scalaire au carré.

- On remarquera aussi qu'il n'est pas nécessaire de garder pour chaque élément les 3 $\mathbf{K}^{T'} \nabla u_{h,T'}$ (6 composantes donc), mais qu'on peut garder dans un tableau les deux composantes de $\mathbf{K}^T \nabla u_{h,T}$ pour chaque triangle T . On repèrera quels sont les voisins de chaque triangle T grâce au tableau des voisins *nvois*. On gardera l'idée de mettre la moitié de la contribution à l'élément ainsi qu'à son voisin. Par conséquent on aura besoin de savoir si le calcul a déjà été réalisé afin de ne pas le refaire. Cela est possible en considérant le fait que les triangles sont traités par ordre croissant (boucle sur les indices des triangles), donc si le voisin d'un triangle a un numéro d'identification plus petit que le sien, le calcul aura déjà été fait.

- De manière similaire à ce qui a été fait pour l'équation de transport, où les composantes du tenseur de diffusion-dispersion est gardé en mémoire pour chaque triangle, la sauvegarde du tenseur de perméabilité doit être réalisée afin de ne pas avoir à recalculer les trois composantes par triangle à chaque fois.
- Le "diamètre" du triangle h_T tel qu'il est calculé (*diam* dans *metric.f* (1.244) : distance entre le centre de gravité et le premier sommet du triangle) peut paraître comme non rigoureux : il dépend en effet du premier sommet entré dans la liste des sommets de chaque triangle. Puisque ce terme doit apparaître au numérateur dans l'expression de μ_T , il est plus avisé de prendre le diamètre du triangle (i.e. la plus grande longueur d'arête). S'il avait été au dénominateur, le rayon du cercle circonscrit au triangle aurait été plus approprié.
- Finalement il n'est pas utile de considérer un vecteur *sqestimator* qui a les contributions en racine carré de chaque estimateur (résiduel ou des sauts). En effet toutes les considérations qu'on peut faire sur *sqestimator* sont possibles sur *estimator*, et l'estimateur global η est bien la somme des contributions directement (sans calcul de la racine carré).
- La fonction "relative_err.f" n'est pas du tout adaptée, ne serait-ce que par son nom. Cette fonction doit retourner la norme K de la solution numérique $\|u_h\|_K$, et pas du tout l'erreur relative qui est $\frac{\|u - u_h\|_K}{\|u_h\|_K}$. De plus, tous les éléments présents au calcul de cette norme sont présents dans la fonction qui calcule l'estimateur ("metric.f"). Finalement cette norme n'est utile que pour calculer la tolérance qui est, rappelons-le, le critère d'arrêt du code lorsque l'estimateur lui est inférieur. Donc nous pouvons nous servir de la tolérance qui sera la somme cumulée des $\|u_h\|_{K,T}^2$ (norme K au carré de la solution numérique sur le triangle T). Il suffira, une fois les boucles finies (pour le calcul de r_T et r_E), de prendre sa racine carré et de la multiplier par la précision souhaitée ϵ , pour avoir la tolérance et ainsi la comparer avec l'estimateur.

6.5 Annexe 5 : Changements apportés pour le transport

Estimateur temporel

- A ne pas oublier : la division par 3 de l'intervalle de temps ainsi que la multiplication par la racine de la constante de décroissance radioactive pour le terme de réaction "Treac".
- Normalement il n'est pas nécessaire de garder pour cet estimateur temporel un tableau avec les contributions locales. En effet seule la valeur finale (somme cumulée des contributions) sera nécessaire pour l'algorithme : on ne raffine pas les triangles en se basant sur cet estimateur, mais on détermine le pas de temps (pas de temps suivant ou refaire le calcul mais pour un pas de temps plus court). En revanche, il peut être intéressant de voir ce que cela donnerait si on se basait sur cet estimateur pour le raffinement en espace.

Estimateurs spatiaux

- Une fois encore l'estimateur résiduel n'est pas le résidu du système linéaire. Il faut donc faire appel aux données directement.
- La valeur au centre de gravité du résidu (appelé, à tort, *secmgr* dans la fonction "ConcErrSpace.f") peut être bien plus simplement calculée (pas besoin des *fctbase-centre*) par la moyenne pondérée des valeurs aux 3 sommets :

$$secmgr = \frac{1}{3} \sum_{i=1}^3 secm(n(i))$$

En effet on a seulement besoin d'une approximation linéaire puisque la solution numérique, elle, est simplement affine par élément.

- Les surfaces duales (surface des Q_i) valent le tiers de l'aire du triangle, calcul immédiat donc et qu'on peut mettre en facteur (entré une seule fois pour chaque triangle).
- Les 3 contributions dans *Tresidual* ne doivent pas être mises au carré car *secmTri* contient déjà les 2nd membres au carré.

Comme pour l'estimateur des sauts dans le cas de l'écoulement permanent la formule de quadrature n'est pas bonne (problème entre somme des carrés et carré des sommes), et son calcul peut se faire dans une nouvelle boucle. On aura en effet enregistré les grandeurs nécessaires de chaque triangle plutôt que de faire le calcul pour chacun des voisins.

Finalement le α_Q du théorème n'est pas l'aire du quadrilatère. Concernant h_Q , puisque ce dernier est au numérateur, on prendra le diamètre de Q , soit la plus grande des diagonales de Q . S'il avait été au dénominateur, on aurait pu décomposer Q en deux triangles, et prendre pour h_q le rayon du cercle inscrit à un de ces triangles par exemple.

- Mauvaise utilisation de la théorie des éléments de référence.
L'estimateur résiduel η_R^n se fait sur les quadrangles $Q = C_i \cap T$. Plusieurs méthodes peuvent être utilisées afin d'approcher la norme $L2$ de R_h^n sur Q , comme on le verra

par la suite.

La contribution de l'estimateur résiduel sur le triangle T sera donnée par la somme des contributions sur chaque Q_i ($i = 1..3$) composant T . On doit évaluer la contribution sur chaque Q_i car cette dernière est composée d'un poids, α_{Q_i} , propre à chaque quadrangle.

La méthode précédemment mise en place pour le calcul de cette norme sur Q est une méthode par élément de référence (cf. ANNEXE 3). Cette méthode ne présente que peu d'intérêt : en effet même si on se ramène au carré de référence \hat{Q} , une approximation de l'intégrale sur ce nouvel élément va être nécessaire, en plus de l'évaluation du déterminant de la Jacobienne B_Q aux sommets de ce carré de référence. Autant réaliser alors une formule de quadrature sur le quadrangle Q .

Il s'est avéré de plus que cette méthode est mal implémentée dans le code, et qu'une fois corrigée, la précision est moindre que les 2 méthodes qui suivent.

Concernant la quadrature directe, on peut choisir de découper Q en 2 triangles T_1, T_2 et d'effectuer une approximation des intégrales sur chaque triangle T_1, T_2 par la méthode des points milieux, et finalement sommer ces 2 valeurs. Cette manière de faire nécessite l'approximation (calculée par la moyenne) du résidu aux points milieux des arêtes des triangles M_i , au centre de gravité du triangle et aux points milieux des arêtes de chaque triangle T_1, T_2 (encore une fois moyenne des extrémités).

On peut finalement décider de réaliser une formule de quadrature sur le quadrangle Q , réalisée encore aux points milieux de Q . On évite ainsi l'approximation du résidu entre le centre de gravité et chacun des sommets du triangle T , pour une très légère perte de précision.

En effet une vérification de toutes ces formules a été réalisée. Prenons le cas-test suivant :

Considérons $u(x, y) := x + y$ et calculons $\int_T u^2 = \sum_{i=1}^3 \int_{Q_i} u^2$ où T est le triangle

défini par les sommets $(0;0)$, $(1;0)$ et $(1;1)$. Les valeurs respectives en chacun des sommets étant alors de 0, 1 et 2. Le calcul analytique de cette intégrale nous donne

$\int_T u^2 = \frac{7}{12} = 0.583333...$. Notons dans le tableau suivant les résultats renvoyés

par les différentes méthodes. Le résultat brut est renvoyé sur la première ligne et le pourcentage d'erreur par rapport à la solution analytique sur la seconde ligne. En première colonne figure les valeurs renvoyées par la méthode implémentée dans le code, dans la seconde apparaissent les résultats de la nouvelle méthode par éléments de référence, suivis par les quadratures en séparant tout d'abord Q en deux triangles, puis finalement en appliquant directement la quadrature sur Q :

	Elmt réf. 1	Elmt réf. corrigé	$T_1 + T_2$	Q
Résultat	0.056071	0.573831	0.583333	0.583333
Erreur (%)	-90.6488	-1.6289	0.0000	0.0000

La résolution de l'équation du transport sur un domaine maillé $\Omega = [0, 30] \times [0, 20]$ et sur plusieurs cas de fonctions-test (solution analytique connue) montrera que les valeurs obtenues sur chaque triangle par les deux méthodes de quadrature,

appliquées soit directement à Q , soit à chacun des triangles de Q , sont les mêmes à 10^{-14} près. On peut donc considérer que seules les erreurs numériques sont responsables de l'écart.

Quant à la méthode par éléments de référence corrigée, les valeurs restent comprises entre 1 et 2% des valeurs retournées par les méthodes de quadrature. La précision semble donc être moindre, avec plus de calculs nécessaires.

En conclusion on choisira la première méthode de quadrature (sur Q) pour calculer l'estimateur résiduel, car elle est légèrement plus rapide que la seconde, avec une précision tout-à-fait similaire.

Tolérances

Le calcul des tolérances se ferait dans "relative_err.f" (qui porte encore mal son nom, puisqu'il s'agit ici du calcul de la norme d'énergie de la solution numérique). On peut encore une fois intégrer ce calcul dans celui de l'estimateur spatial global : tous les éléments nécessaires apparaissent dans la fonction "ConcErrSpace2(_moi).f".

On peut remarquer que les intégrales en temps sur les termes u_h^2 et $(\mathbf{D}\nabla u_h, \nabla u_h)$ sont approchées (méthode des rectangles à gauche) alors que le calcul de ces intégrales peut être fait de manière exacte (voir le détail de la tolérance **PAGE A METTRE**). Cela entraîne alors le passage en argument des valeurs du temps antérieur, ainsi que la somme cumulée des α_m pour pouvoir calculer l'erreur relative dans le cadre des solutions analytiques.

7 Bibliographie

[1] B. AMAZIANE, M. BOURGEOIS, M. EL FATINI. Recherche d'un estimateur d'Erreur a Posteriori pour le logiciel MELODIE en 2D et 3D et Application à l'Adaptation de maillage. *Rapport d'avancement n°1*, 2009 ; *Rapport de recherche n°2*, 2010 ; *Rapport de recherche final*, 2011

[2] B. AMAZIANE, A. BERGAM, M. EL OSSMANI & Z.MGHAZLI. *A posteriori* estimators for vertex centred finite volume discretization of a convection-diffusion-reaction equation arising in flow porous media. *International Journal for Numerical Methods in Fluids*, 2009 ; 59 :259-284

[3] R. VERFURTH. Robust *a posteriori* error estimates for nonstationary convection-diffusion equations. *SIAM, J. Numer. Anal.* 43 :1783-1802, 2005

[4] Danielle HILHORST, Martin VOHRALIK. A posteriori error estimates for combined finite volume - finite element discretizations of reactive transport equations on nonmatching grids, *Comput. Methods Appl. Mech. Engrg.* (2010), doi : 10.1016/j.cma.2010.08.017

[5] Notice théorique du code de calcul MELO version 5.0 *Rapport DSU/SSIAD n°10*