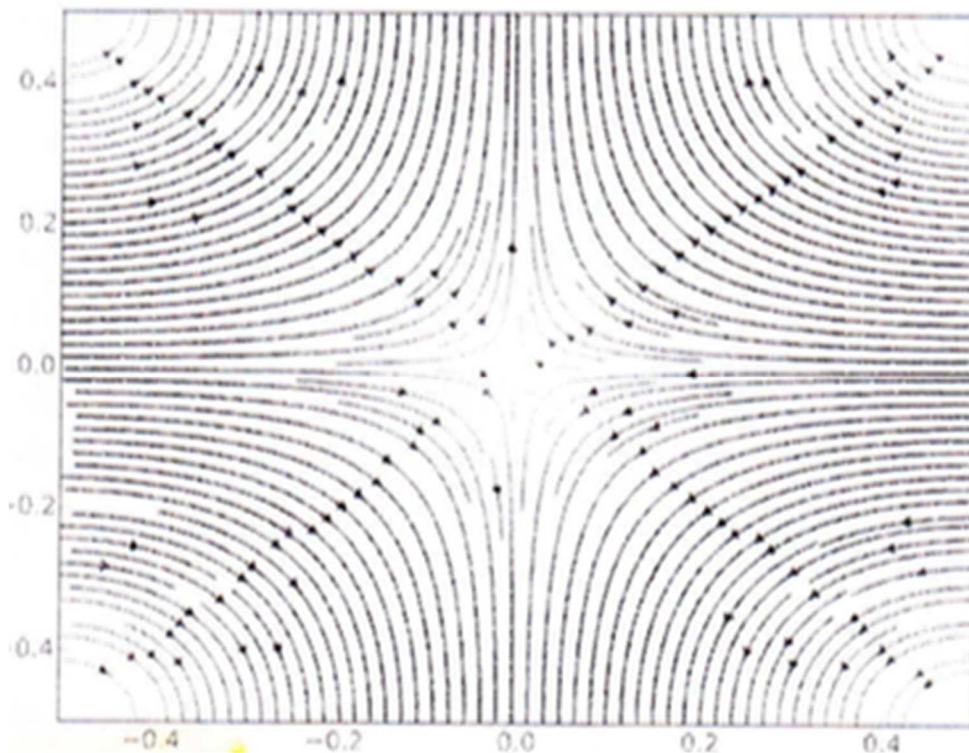


Bureau d'études Volumes finis : Conception d'un réacteur chimique à échelle réduite



L'objectif de ce bureau d'études est d'étudier la cinétique chimique lors de l'injection de deux réactifs au sein d'un réacteur chimique. Cette cinétique étant fortement influencée par la température, on se propose d'étudier le phénomène d'advection-diffusion de cette dernière par la méthode des volumes finis.



Sommaire

1) Cahier des charges et objectifs	4
2) Codage du pré-processeur	5
a) Utilisation d'un module	5
b) Lecture d'un fichier de données.....	5
c) Création du maillage	5
d) Calcul du champ de vitesse.....	7
e) Calcul du pas de temps	9
3) Noyau du calcul.....	9
a) Calcul des flux	9
b) Définition des conditions aux limites	11
c) Mise à jour de la température.....	12
4) Exploitation	14
a) Test d'advection pur	14
b) Test de diffusion pur	16
Conclusion	17

Tables des figures

Figure 1 : Principe de fonctionnement du réacteur chimique	4
Figure 2 : Création du maillage	7
Figure 3 : Tracé de la composante u de la vitesse	8
Figure 4 : Tracé de la composante v de la vitesse	8
Figure 5 : Calcul du signe pour la vitesse	11
Figure 6 : Boucle de mise à jour de la température	13
Figure 7 : Evolution du flux advectif pur (1)	14
Figure 8 : Evolution du flux advectif pur (2)	15
Figure 9 : Ajout d'un coefficient de diffusivité thermique pour le flux advectif pur	15
Figure 10 : Diffusion de la température	16

1) Cahier des charges et objectifs

Le but de ce bureau d'études est d'étudier l'évolution des champs de température liés aux phénomènes d'advection et de diffusion au sein du réacteur chimique. Pour ce faire, nous avons réalisé un programme codé en Fortran 95 devant répondre à certaines conditions :

- Modélisation par la méthode des volumes finis
- Maillage régulier en x et y
- Schéma amont pour le terme advectif
- Schéma centré pour le terme diffusif
- Intégration temporelle par la méthode d'Euler explicite
- Intégration des conditions aux limites

Fort de l'expérience acquise lors d'un projet Fortran réalisé au semestre 5, nous avons décidé d'articuler notre programme de manière à séparer tous les points du cahier des charges en sous-routines. Cette méthode nous a permis de garder un programme principal clair et relativement court qui sera décrit plus en détail dans la suite de ce rapport.

Avant de nous lancer dans le codage pur de ce projet, nous avons réalisé des séances en salle afin de discrétiser l'équation du transport de la température :

$$\frac{\partial T}{\partial t} + \nabla \cdot (\vec{U}T) - \nabla \cdot (\alpha \vec{\nabla} T) = 0$$

Ensuite, nous avons mis nos idées par écrit en rédigeant un algorithme de résolution de ce problème. Cette approche a permis une bonne compréhension du sujet ainsi qu'un fil conducteur quant aux différentes étapes régissant ce projet.

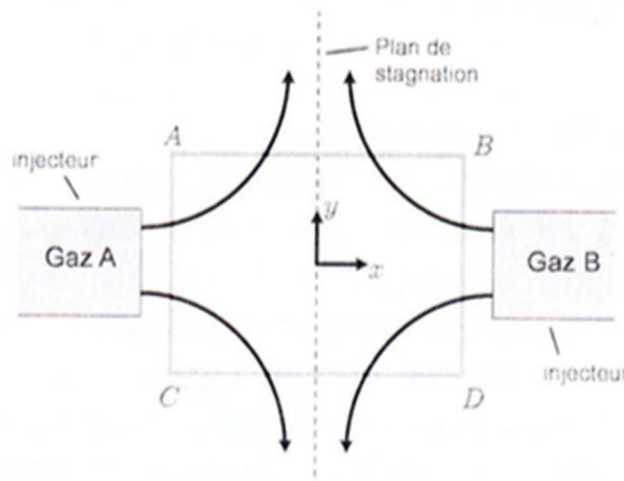


Figure 1 : Principe de fonctionnement du réacteur chimique

2) Codage du pré-processeur

a) Utilisation d'un module

Dans l'optique de garder un programme principal le plus propre possible, nous avons fait le choix d'utiliser un module. Ce dernier permet de généraliser la déclaration de toutes les variables qui vont être utilisées au cours de ce projet, qu'importe que ce soit dans le programme principal ou dans une subroutine il nous suffira alors d'entrer la commande « USE module_VF » et les instructions qui suivront seront capables de comprendre à quelle variable nous faisons appel.

b) Lecture d'un fichier de données

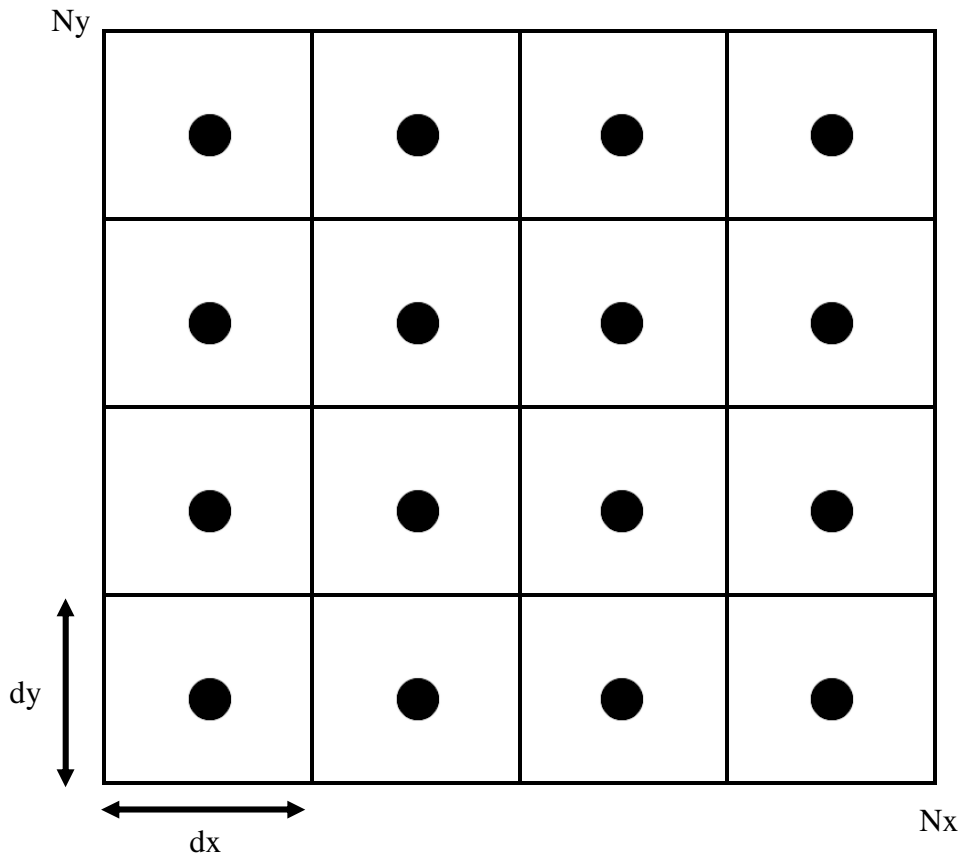
Dans un projet complexe comme celui-ci, nous aurons besoin de certaines constantes dans les calculs sur lesquels l'utilisateur du programme pourra avoir accès facilement. C'est dans cet objectif que nous avons créé un fichier texte « data.txt » contenant toutes les constantes. Si l'utilisateur en voit le besoin, il peut modifier la valeur de ces paramètres dans le fichier texte et la modification sera prise en compte dans tout le programme sans qu'il ait besoin de s'en soucier.

Pour intégrer ces données à notre programme principal, nous avons créé une subroutine permettant la lecture de ce fichier et l'affectation des valeurs qu'il contient aux variables qu'il vise. La subroutine ouvre le fichier texte puis lit ligne par ligne les valeurs qu'il contient. Elle les affecte directement aux variables que l'on a défini.

Si par ailleurs il était nécessaire d'ajouter un paramètre non existant à ce jour dans le fichier texte, il faudrait ajouter une ligne dans la subroutine « lecture_fichier » afin qu'elle lise cette valeur et l'affecte à la variable qui vient d'être créée.

c) Création du maillage

La première étape de programmation a été la création du maillage dans lequel seront réalisés tous les calculs de flux, de température, de vitesse, etc... Nous avons fait le choix de réaliser un maillage uniforme avec le nombre de nœuds N_x selon l'axe x et N_y selon l'axe y.



Comme on le voit sur le schéma ci-dessus, nous avons réalisé un quadrillage du réacteur chimique à échelle réduite en définissant un nombre de nœud N_x/N_y (à choisir dans le fichier data.txt) ainsi qu'un pas dx/dy calculer de la manière suivante :

$$dx = \frac{(x_2 - x_1)}{N_x - 1}$$

On y voit également les centres des volumes représentés par des points dans le schéma précédent qui sont au nombre de N_x-1 et N_y-1 . La discrétisation de type volumes finis permet de modéliser tout type de géométrie plus ou moins complexe de manière précise. Cependant, dans notre cas, nous avons conservé un maillage uniforme tout le long de l'étude par manque de temps. Le maillage a été paramétré entre $-L/2$ et $L/2$ selon x et y de manière à être toujours symétrique.

Afin de réaliser ce maillage, nous avons dans un premier temps créé deux vecteurs X et Y dans lesquels se trouvent les valeurs issues d'une boucle itérative balayant le domaine $-L/2$ à $L/2$ avec un pas dx (ou dy selon l'autre axe). La taille du vecteur pouvant varier au cours de l'utilisation du programme ou selon les besoins de l'utilisateur, nous avons utilisé les fonctions d'allocation de

tableaux dynamique (« ALLOCATBLE ») nous permettant non pas de fixer la taille du tableau lors de la création de celui-ci mais uniquement après avoir lu le fichier de données et donc de connaître le nombre de nœuds qui allaient composer le maillage. Nous avons également utilisé cette méthode pour tous les autres vecteurs/tableaux de notre programme.

Afin de tracer le maillage et avoir un premier aperçu, nous avons récupéré une subroutine nommé « VTSWriter2D.f90 » permettant la sortie des résultats au format *Paraview*. Par soucis de compatibilité, cette subroutine n'acceptait en entrée que des tableaux pour décrire les X et Y, c'est pourquoi nous avons créé des matrices tab_x/tab_y de tailles Nx x Nx (et Ny x Ny) dans lesquels nous avons simplement injecté à chaque ligne le vecteur X et Y précédemment définis.

On obtient le résultat suivant pour un choix de Nx = Ny = 10 :

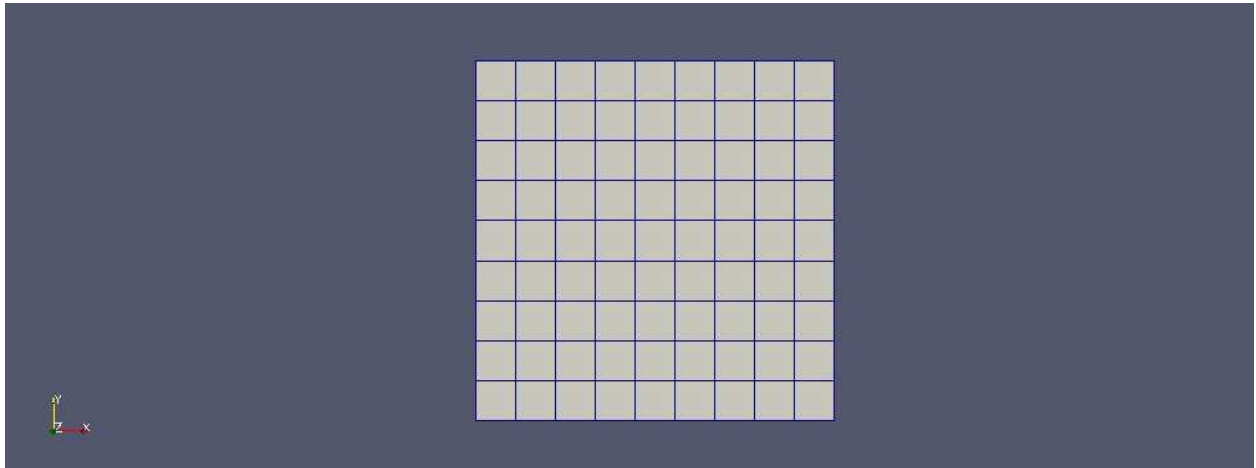


Figure 2 : Création du maillage

d) Calcul du champ de vitesse

Une fois le maillage créé, nous avons calculé le champ de vitesse dans ce domaine grâce à la formule suivante :

$$\vec{U} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{cases} A \cos\left(\pi\left(\frac{x}{L} - \frac{1}{2}\right)\right) \sin\left(\pi\left(\frac{y}{L} - \frac{1}{2}\right)\right) \\ -A \sin\left(\pi\left(\frac{x}{L} - \frac{1}{2}\right)\right) \cos\left(\pi\left(\frac{y}{L} - \frac{1}{2}\right)\right) \end{cases}$$

A étant un paramètre caractéristique de la vitesse maximale du gaz (A=0,05 m/s).

Le calcul des vitesses se fait en chaque centre des volumes et c'est pourquoi nous avons créé de nouveaux tableaux tab_xc/tab_yc qui sont calculés dans une boucle de 1 à N_{x-1} tel que :

$$\begin{cases} tab_{xc}(i,j) = tab_x(i,j) + \frac{dx}{2} \\ tab_{yc}(i,j) = tab_y(i,j) + \frac{dy}{2} \end{cases}$$

De cette manière, les « x » et « y » que l'on voit dans la formule de calcul de la vitesse correspondent dans notre cas à ces tableaux de centre. On obtient finalement les profils de vitesse suivants pour les composantes u et v :

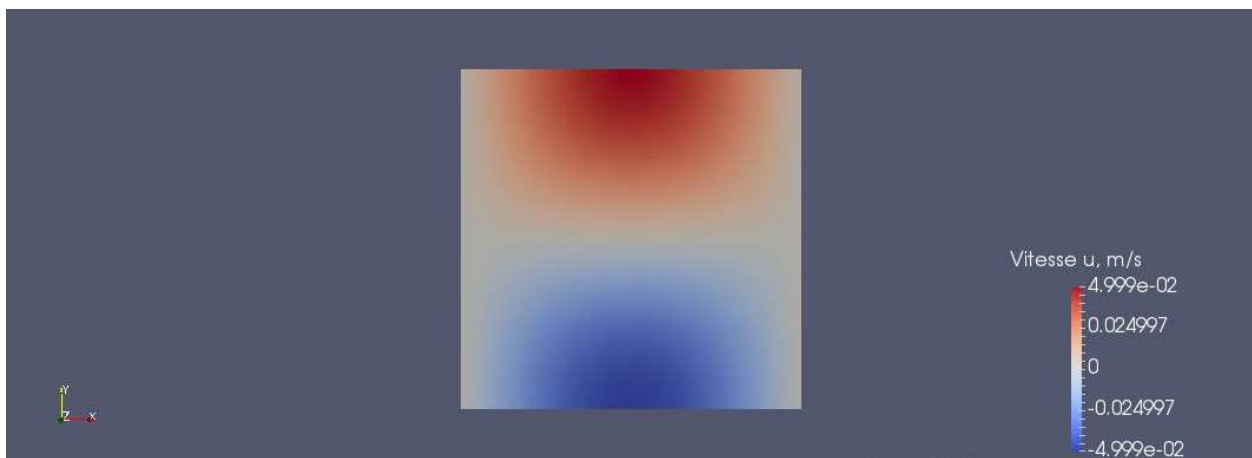


Figure 3 : Tracé de la composante u de la vitesse

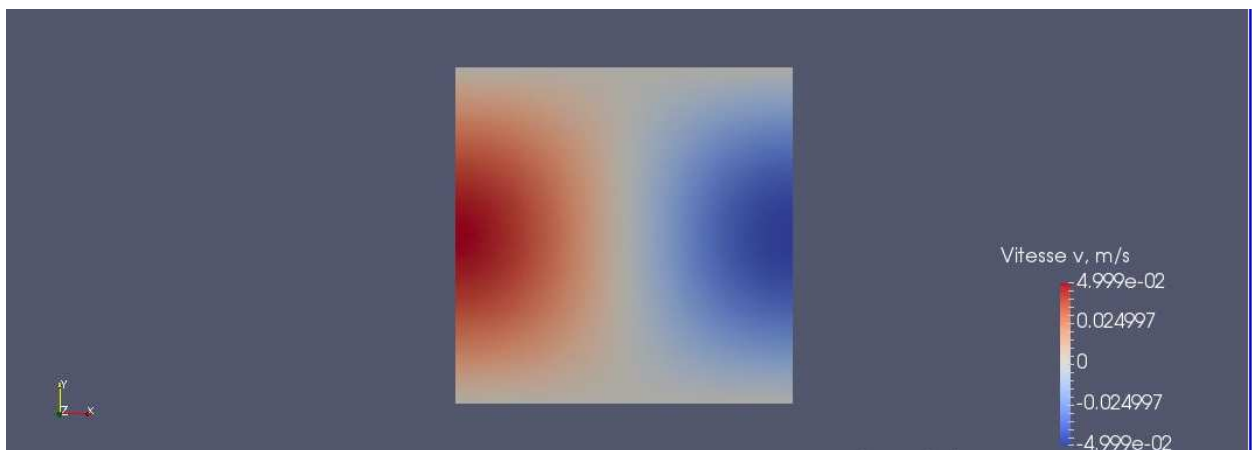


Figure 4 : Tracé de la composante v de la vitesse

e) Calcul du pas de temps

La dernière étape dans notre pré-approche du problème est le calcul du pas de temps qui nous servira lors de la mise à jour de la température en fonction du temps. L'expression de ce dernier est de la forme :

$$dt = \min \left\{ \left(\frac{|u|}{dx * CFL} + \frac{|v|}{dy * CFL} + \frac{\alpha}{r} \left(\frac{1}{dx^2} + \frac{1}{dy^2} \right) \right) \right\}^{-1}$$

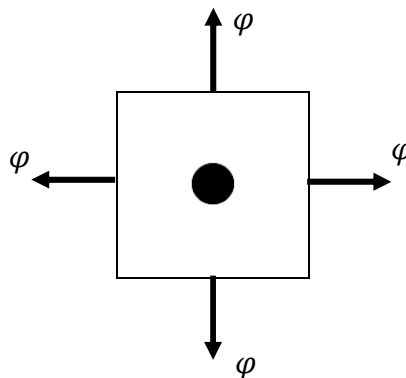
Les composantes u et v étant des tableaux, nous avons organisé le calcul de ce pas de temps un peu différemment de la formule ci-dessus. Nous avons tout d'abord rempli un tableau « dt_loc » de chacune des valeurs de « dt » en fonction de u et v . C'est seulement une fois le tableau créé que nous avons demandé à trouver la plus petite valeur de celui-ci et de l'affecter à « dt ».

On remarque la constante « CFL » dans le calcul qui représente le nombre de Courant et « r » le nombre de Fourier que l'on a respectivement fixés égaux à 1 et 0,5. Ces deux constantes sont des conditions de stabilité dans le cas de calcul par la méthode des volumes finis.

3) Noyau du calcul

a) Calcul des flux

Si nous observons le cas d'un seul volume du maillage, nous remarquons que les flux autour de ce volume sont organisés de la manière suivante :



C'est donc un calcul qui se réalise en chaque nœud et non en chaque centre comme c'est le cas pour les vitesses et les températures. La direction des flux varie en fonction des vitesses aux centres des volumes de deux cases adjacentes.

Les flux advectifs et diffusifs dans le réacteur chimique sont définis de la manière suivante :

$$\begin{cases} \varphi_{adv} = \vec{u} * \rho C_p T \\ \varphi_{diff} = -k \nabla T \end{cases}$$

En partant de l'équation du transport de la température décrite en introduction, on obtient une équation de la forme :

$$\frac{d}{dt}(Vol_{i,j} \bar{T}_{i,j}) = \int (\vec{U}T) \cdot \vec{n} dS + \int \alpha(\nabla T) \cdot \vec{n} dS$$

Cette intégrale sur la surface laisse apparaître une distinction entre les flux advectifs en x et en y (de même pour les flux diffusifs) que nous noterons désormais flux NORD pour les flux selon x et flux EST pour les flux selon y. On obtient finalement une équation du type :

$$\begin{aligned} \frac{d}{dt}(Vol_{i,j} \bar{T}_{i,j}) = & \varphi_{adv,n,ij} dx + \varphi_{adv,s,ij} dx + \varphi_{adv,o,ij} dy + \varphi_{adv,e,ij} dy \\ & + \varphi_{diff,n,ij} dx + \varphi_{diff,s,ij} dx + \varphi_{diff,o,ij} dy + \varphi_{diff,e,ij} dy \end{aligned}$$

Et $\begin{cases} \varphi_{adv,n,ij} = -\varphi_{adv,s,ij+1} \\ \varphi_{adv,e,ij} = -\varphi_{adv,o,ij+1} \end{cases}$ (de même pour les flux diffusifs)

On obtient alors le schéma amont suivant pour l'advection :

$$\varphi_{adv,n,ij} = \begin{cases} U_{i,j} \bar{T}_{i,j} & \text{si } \vec{U}_{i,j} \cdot \vec{n}_n > 0 \\ U_{i,j+1} \bar{T}_{i,j+1} & \text{si } \vec{U}_{i,j} \cdot \vec{n}_n < 0 \end{cases}$$

$$\varphi_{adv,e,ij} = \begin{cases} U_{i,j} \bar{T}_{i,j} & \text{si } \vec{U}_{i,j} \cdot \vec{n}_e > 0 \\ U_{i+1,j} \bar{T}_{i+1,j} & \text{si } \vec{U}_{i,j} \cdot \vec{n}_e < 0 \end{cases}$$

Ainsi que les schémas centrés suivants pour la diffusion :

$$\begin{cases} \varphi_{diff,n,ij} \\ \varphi_{diff,e,ij} \end{cases} = \begin{cases} \alpha \frac{dT}{dy} = \alpha \frac{T_{i,j+1} - T_{i,j}}{Y_{i,j+1} - Y_{i,j}} \\ \alpha \frac{dT}{dx} = \alpha \frac{T_{i+1,j} - T_{i,j}}{X_{i+1,j} - X_{i,j}} \end{cases}$$

Pour rappel, la direction des flux change en fonction de la vitesse des deux cases adjacentes. C'est pourquoi, dans notre boucle de calcul, un test logique nous indique quelle est la vitesse à prendre en compte.

```
DO i=2,Nx-1
  DO j=1,Ny-1
    shift = int(-0.5*(1+sign(1., 0.5*(U(i-1,j)+U(i,j))))
    flux_adv_X(i,j) = U(i+shift,j)*T(i+shift,j)*dy
  END DO
END DO
```

Figure 5 : Calcul du signe pour la vitesse

Nous avons déclaré un entier « shift » qui est le résultat de notre test logique. Ce test consiste à réaliser une moyenne entre les vitesses de deux cases l'une à côté de l'autre et d'affecter le signe du résultat à 1. Si la moyenne est négative on obtient « shift = 0 » et donc on calcule le flux avec la vitesse et la température en (i,j). Cependant, si la moyenne est positive on obtient « shift = -1 » et donc on calcule le flux avec les vitesses et températures en (i-1, j). Ici, nous avons l'exemple pour le calcul des flux selon l'axe x mais le même test régit les flux selon y.

b) Définition des conditions aux limites

Les calculs décrits jusqu'à présent étaient définis sur le « centre du réacteur chimique » sans prendre en compte les extrémités déterminées par les conditions aux limites, tant pour les températures que pour les flux advectifs/diffusifs.

Pour commencer, les températures aux extrémités droite et gauche de notre réacteur sont définies par un profil gaussien tel que $T_A(y)$ les températures à gauche et $T_B(y)$ les températures à droite sont de la forme :

$$\begin{cases} T_A(y) = (T_a - T_0) e^{-\frac{y^2}{2\sigma_A^2}} + T_0 \\ T_B(y) = (T_b - T_0) e^{-\frac{y^2}{2\sigma_B^2}} + T_0 \end{cases}$$

Ces températures interviennent ensuite dans le calcul des flux aux extrémités de notre maillage. En effet, nous exprimons aussi des conditions aux limites pour le calcul des flux advectifs et diffusifs. Nous nous intéressons toujours aux extrémités gauche et droite pour le moment.

Pour les flux advectifs, nous définissons les flux au point $[1, j]$ (extrémité gauche) comme étant le produit entre la vitesse au point et le profil de température gaussien T_A . En ce qui concerne le point $[N_x, j]$ (extrémité droite), nous avons fixé le fait que ce soit un flux sortant et c'est pourquoi nous calculons sa valeur avec la vitesse et la température du point en N_{x-1} .

En revanche, pour les flux diffusifs, les deux extrémités prennent en compte les profils de température T_A et T_B tel que :

$$\begin{cases} \varphi_{diff,x}(1,j) = -\alpha \frac{T_A - T(1,j)}{\frac{dx}{2}} * dy \\ \varphi_{diff,x}(N_x,j) = -\alpha \frac{T(N_x-1,j) - T_B}{\frac{dx}{2}} * dy \end{cases}$$

Nous pouvons dans notre cas nous permettre de simplement diviser par $dx/2$ car nous sommes en présence d'un maillage uniforme.

Intéressons-nous désormais aux conditions aux limites aux extrémités hautes et basses de notre maillage. Ces dernières sont plus évidentes car elles sont égales à la valeur du flux précédent (valable pour le flux diffusif et advectif). Par exemple, pour le flux diffusif en $(i,1)$, nous fixons la valeur telle que :

$$\varphi_{adv,y}(i,1) = \varphi_{adv,y}(i,2)$$

Nous obtenons donc un total de 8 conditions aux limites, 4 sur les flux advectifs et 4 sur les flux diffusifs.

c) Mise à jour de la température

Dans cette partie, on s'intéresse à expliquer la façon dont est mise à jour la température en fonction du temps d'observation. On définit une évolution de la température grâce au schéma d'Euler explicite tel que :

$$T_{i,j}^{n+1} = T_{i,j}^n + \frac{dt}{dx*dy} [\varphi_{adv,y}(i,j) - \varphi_{adv,y}(i,j+1) + \varphi_{adv,x}(i,j) - \varphi_{adv,x}(i+1,j) + \varphi_{diff,y}(i,j) - \varphi_{diff,y}(i,j+1) + \varphi_{diff,x}(i,j) - \varphi_{diff,x}(i+1,j)]$$

Comme on le voit avec les indices (i, j), on réalise dans une subroutine nommée « maj_temp » une double boucle sur i et sur j afin de balayer les matrices de flux et de températures. Dans un second temps, on souhaite réaliser une boucle sur le temps et celle-ci se fait dans notre programme principal en lui-même. Pour ce faire, nous devons définir un temps final (inscrit dans le fichier de données) ainsi qu'un nombre d'itérations Nt calculé comme étant le rapport du temps final sur le pas de temps déterminé précédemment. On réalise alors une boucle de 1 à Nt comportant l'appel de plusieurs subroutines comme on peut le voir ci-dessous :

```
DO k=1,Nt
    CALL flux_adv
    CALL flux_diff
    CALL maj_temp
    CALL VTSWriter(k*dt,k,Nx,Ny,tab_x,tab_y,T,U,V,"int")
END DO
```

Figure 6 : Boucle de mise à jour de la température

On peut apercevoir l'appel de la subroutine « VTSWriter » qui comporte un certain nombre de variables d'entrée que l'on va décrire ici :

- Le temps
- L'étape : c'est le numéro qui sera attribué à notre fichier de données en sortie
- Nombre de nœuds Nx dans la direction x
- Nombre de nœuds Ny dans la direction y
- Tableau des abscisses des nœuds
- Tableau des ordonnées des nœuds
- Champ de température
- Composante de la vitesse u
- Composante de la vitesse v
- Variable de type chaîne de caractère variant si c'est la première ou la dernière fois que l'on fait appel à la subroutine, ou bien si c'est une étape intermédiaire.

4) Exploitation

a) Test d'advection pur

Afin de vérifier et de valider ce que nous avons programmé jusqu'à présent, nous réalisons des tests simples en supprimant des termes des équations comme pour l'exemple qui va suivre et qui représente l'advection pure de la température, sans aucune prise en compte de la diffusion.

Pour réaliser cela, nous simplifions notre programme en mettant en commentaire l'appel de la subroutine « flux_diff » dans le programme principal ainsi que les termes de flux diffusif dans le calcul de la mise à jour de la température. Ensuite, on fixe les composantes de la vitesse u et v respectivement à 1 et 0 ainsi que le coefficient de diffusivité thermique α à 0.

De cette manière, nous obtenons les résultats attendus suivants :

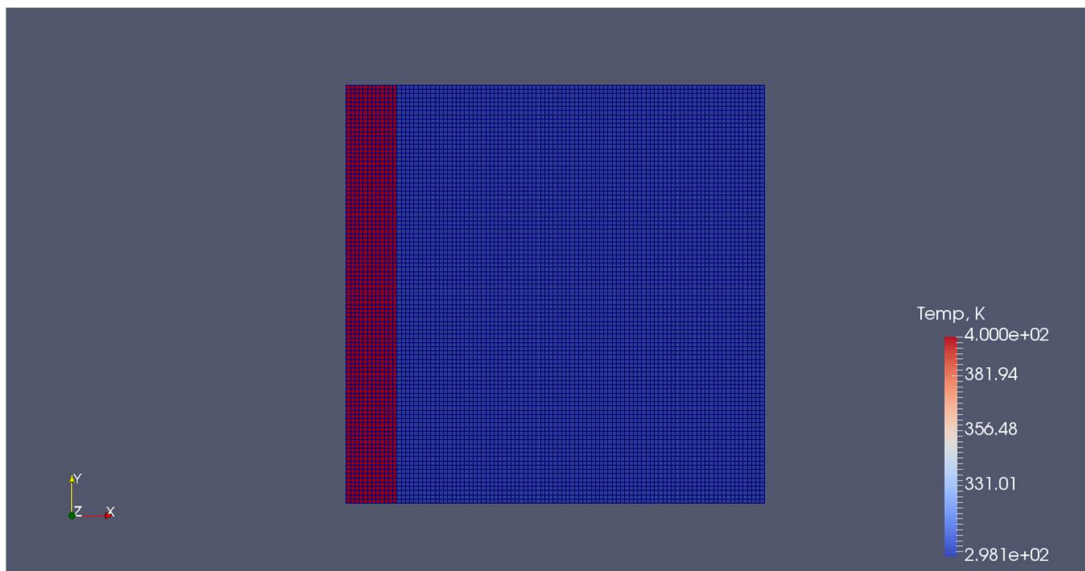


Figure 7 : Evolution du flux advectif pur (1)

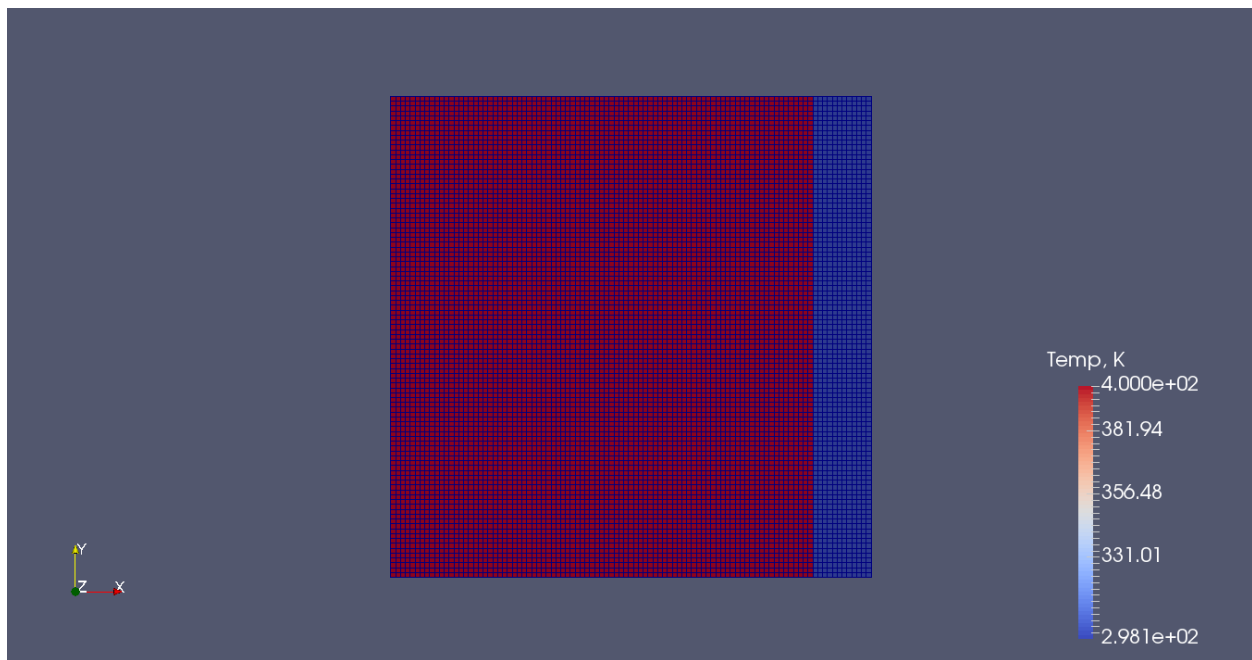


Figure 8 : Evolution du flux advectif pur (2)

Si nous n'avons pas fixé le coefficient de diffusivité thermique à 0, nous aurions pu observer quelques effets intervenir dans le transport de température comme on peut le voir sur l'image ci-dessous :

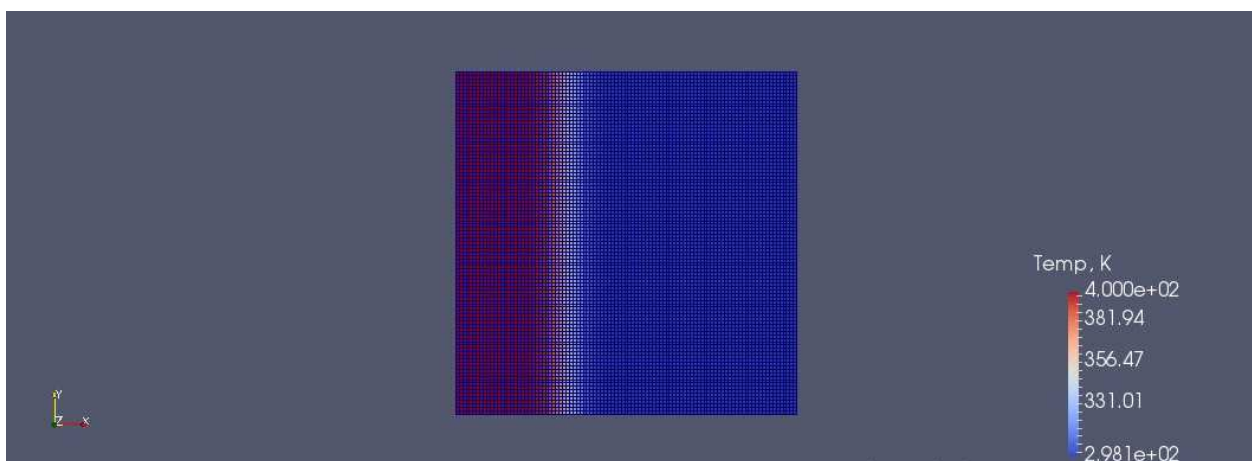


Figure 9 : Ajout d'un coefficient de diffusivité thermique pour le flux advectif pur

b) Test de diffusion pur

De la même manière, on s'intéresse à tester notre code afin de visualiser uniquement l'effet de diffusion de la température dans notre réacteur et on obtient le résultat suivant :



Figure 10 : Diffusion de la température

Dans cette configuration, nous avons « réactivé » le calcul des flux diffusifs ainsi que l'appel de la subroutine le permettant dans la boucle temporelle. Nous avons également fixé la valeur du coefficient de diffusivité thermique à $2,0e-5$ et du vecteur vitesse à 0. De cette manière, nous avons pu supprimer tous les effets liés au phénomène de l'advection observés précédemment.

Malheureusement, par manque de temps, nous n'avons pas eu l'occasion d'aborder tous les points de vue proposés dans le sujet de ce bureau d'études. Cependant, cette première approche nous a déjà permis de comprendre un grand nombre de choses sur le phénomène de diffusion et l'impact qu'il a dans le transport de la température au sein d'une modélisation comme celle du réacteur chimique à échelle réduite.

Conclusion

Nous avons pu découvrir un exemple concret d'une étude de modélisation numérique par la méthode des volumes finis en réalisant un réacteur chimique à échelle réduite. Le projet s'est articulé selon les différents critères du cahier des charges. Le premier a été la création d'un ensemble de sous-routines appelé pré-processeur nous permettant par la suite de réaliser le gros du calcul. Ensuite, nous avons codé le solveur ayant la capacité de calculer tous les flux dans notre maillage selon les conditions aux limites fixées par le cahier des charges. Nous avons ensuite eu la chance de vérifier le bon fonctionnement de notre programme à travers deux applications simplifiant le calcul des flux qui sont l'observation du phénomène d'advection pur puis de la diffusion pure.

Concernant notre point de vue personnel, ce BE nous a permis de nous familiariser avec une nouvelle méthode de discrétisation. Etudier un cas concret comme le transport de la température dans un réacteur chimique a fortifié notre logique de programmation, autrement dit, nous avons pu augmenter nos performances en FORTRAN grâce à une nouvelle organisation des programmes par exemple (sous-routine, module, etc...).