

But : Il faut *in fine* obtenir un code de calcul 2D en cartésien et maillage à pas constant. Ci-après est proposée une démarche permettant d'aboutir pas à pas.

A la fin de chaque étape créer un dossier *etapex* qui contient tous les fichiers utiles pour la compilation et l'exécution du code, à l'image du dossier *etape0*.

EN 1D :

Etape 1 : Prise en main du code 1D

Compiler, lancer le code sans modification. Tracer des solutions.

Modifier le fichier d'entrée (DATAINIT.DAT) et se rendre des comptes de effets.

Etape 2 : Modification basique.

Changer la condition initiale :

- a) pour permettre la résolution d'un écoulement uniforme au repos
- b) pour permettre la résolution d'un tube à choc.

Hints :

*On écrira les données à transmettre dans le fichier INITGAZ.DAT

*On conservera un fichier solution du Tube à Choc dans un fichier « Solref », qui servira de fichier de référence quand on comparera les solutions des codes modifiés.

Vers le 2D :

Etape 3 : Ajouter la variable v (vitesse transverse) dans le code : Il faudra écrire l'équation relative à cette variable au préalable.

Refaire les calculs avec une vitesse $v=0$ puis vitesse $v \neq 0$.

Hint :

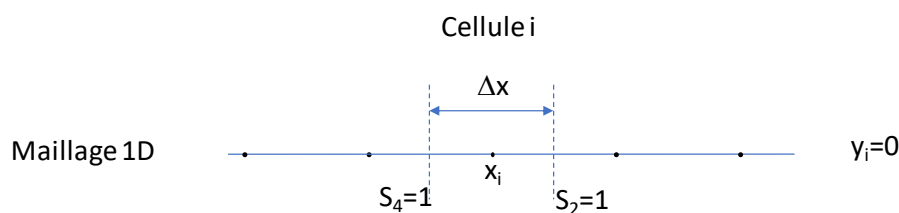
*La solution en tube à choc **1D** est inchangée si l'écoulement à une vitesse transverse (v) non nulle....

* A ce niveau les seules sous-routines à changer sont : `initgaz`, `prim`, `ecritplot`, `green_flux`.

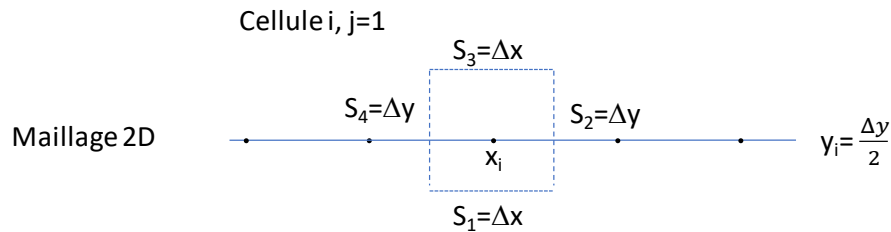
Etape 4 : Modifier compléter les éléments géométriques du maillage afin de décrire un domaine bidimensionnel simple (rectangle).

Hint : L'essentiel de cette étape se réalise dans la fonction qui crée le maillage. On fera apparaître arbitrairement un indice j , qui vaut 1.

Les principales caractéristiques d'une cellule de calcul 1D sont :



En 2D on obtient un maillage équivalent à ce maillage 1D :



Ne pas oublier qu'en 2D, le volume de la cellule est une surface, définie par son vecteur normal et son aire ($\Delta x \Delta y$), qu'en 1D ce volume est un segment de droite (Δx).

Ne pas oublier qu'en 2D, un vecteur a 2 composantes (p.e. : les normales aux faces).

Etape 5 : Projection des vecteurs, des flux dans les bons repères

On dispose maintenant d'un repère fixe (\vec{i}, \vec{j}) et des repères liés aux faces ($\vec{n}_k, \vec{\tau}_k$)

Il faut aussi regarder les projections des vecteurs sur les repères liés aux faces.... et le passage inverse.

Etape 6 en 3 sous-étapes : Ajouter les calculs nécessaires dans la direction transverse.

Etape 6A : Repérer les boucles portant sur i et ajouter si **nécessaire** des boucles sur j en fixant $n_y=1$. On notera que l'ajout d'une boucle de ' $j=1,1$ ' ne change rien... (Attention au passage de NY dans la liste d'arguments des sous-routines)

Etape 6B : Augmenter le rang des tableaux quand nécessaire : faire un tableau après l'autre sous peine de perdre tout le travail. Par exemple, le tableau $VAR(: , :)$ devient $VAR(: , : , :)$ et ses dimensions $VAR(0 : NX+1 , NEQ)$ deviennent $VAR(0 : NX+1 , 0 : NY+1 , NEQ)$.

Vérifier à la fin de l'étape que les solutions du tube à choc 1D sont toujours bien retrouvées.

Etape 7 : Bien comprendre quels sont les flux calculés dans la sous-routine intercell en 1D. Calcule-t-on tous les flux ? Jusqu'à présent, sur quelles faces travaille-t-on (boucle sur les faces) ?

Réfléchir aux flux calculés en 2D, et seulement **ensuite** compléter la sous-routine intercell.

Vérifier à la fin de l'étape que les solutions du tube à choc 1D sont toujours bien retrouvées.

Refaire les cas tests de l'étape 2 en prenant un seul élément dans la direction transverse ($NY=1$)

Il faudra aussi modifier la sous-routine LIMGAZ, et faire attention au calcul du pas de temps.

Surtout bien vérifier que les boucles sur les faces sont maintenant bien réalisées.

Etape 8 : Faire les cas test du tube à choc vertical (haute pression en haut, basse pression en bas) ($NX=1$).

A la fin de cette étape, le code fonctionne en 2D.

Le 2D :

Etape 9 : Faire un calcul de tube à choc 2D et visualiser les résultats sur paraview.