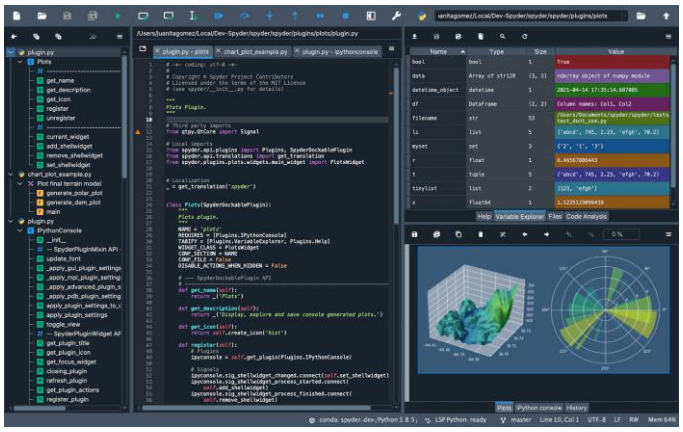
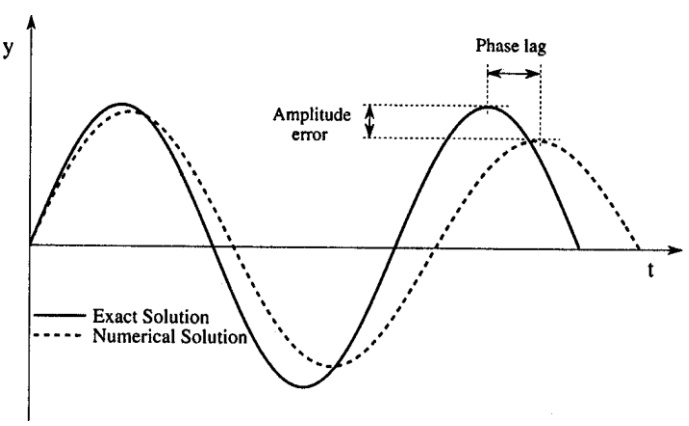
UE 703 Méthodes Numériques pour la Mécanique

POLYCOPE DE TP – M1 ENERGIE





Olivier BOTELLA

LEMTA – Université de Lorraine - CNRS
2 avenue de la Forêt de Haye, TSA 60604
54518 Vandoeuvre (France)

Email : Olivier.Botella@univ-lorraine.fr

Lectures web page : https://arche.univ-lorraine.fr/course/view.php?id=6745

Analyse Numérique - M1 Energie

Enoncés des TP - Méthodes numériques avec PYTHON et FLUENT

https://arche.univ-lorraine.fr/course/view.php?id=6745

olivier.botella@univ-lorraine.fr

1 Présentation des TP sous Python et ANSYS Fluent

Les TP de cette UE se déroulent en 9 séances de 3h :

- 5 séances de programmation Python des méthodes numériques FD et FV du cours et TD.
- 3 séances d'initiation au logiciel de simulation en mécanique des fluides et thermique ANSYS Fluent[1].

La note des TP sera basée sur l'évaluation par l'encadrant de votre travail et assiduité lors des séances, et aussi sur les compte-rendu individuels que vous devez rendre à la fin du semestre.

Vous devez rédiger un compte-rendu pour chaque sujet de TP :

- Les compte-rendu devront répondre aux questions de l'énoncé, et comporter une dizaine de pages au maximum. Les réponses doivent être rédigées de façon précise et succincte (ne recopiez pas l'énoncé !), et seront illustrées par le listing des programmes Python que vous écrirez, et les figures que vous réaliserez.
- Les compte-rendu doivent être rédigés avec le traitement de texte Word ou convertis en fichiers Pdf. N'oubliez pas de mettre votre nom et l'intitulé du TP sur les compte-rendu.

Les compte-rendu de TP doivent être déposés sur la page ARCHE du cours, avant la date limite que vous fixera l'encadrant.

2 Travaux pratiques d'analyse numérique avec Python

Les applications directes du CM et TD s'effectueront dans le langage de programmation Python et sur l'environnement de développement Spyder. Pour réaliser ces TP, vous devez télécharger sur la page ARCHE du cours[2] l'archive `tp_python_analyse_num_m1energie.zip` et l'extraire sur votre PC. Cette archive contient les programmes Python à compléter lors des séances.

2.1 L'environnement de développement intégré (IDE) Spyder

Le logiciel open-source Spyder est une IDE spécialement conçue pour la programmation scientifique en Python, qui inclus les bibliothèques les plus courantes du langage Python, telles que Matplotlib et Numpy. Son interface graphique et ses fonctionnalités sont très similaires à celles du logiciel Matlab, *cf.* figure 1. Le logiciel Spyder est disponible au téléchargement gratuit https://www.spyder-ide.org/. Il est aussi inclus dans la suite de logiciels Anaconda (https://www.anaconda.com/download). Suivant les salles informatiques de la FST, Spyder est directement installé ou accessible à partir d'Anaconda.

[1] Page Web : https://www.ansys.com/products/fluids/ansys-fluent.

[2] https://arche.univ-lorraine.fr/course/view.php?id=6745. Alternativement, vous pouvez chercher les mots clefs *botella analyse* sur la page d'accueil de Arche: https://arche.univ-lorraine.fr/.

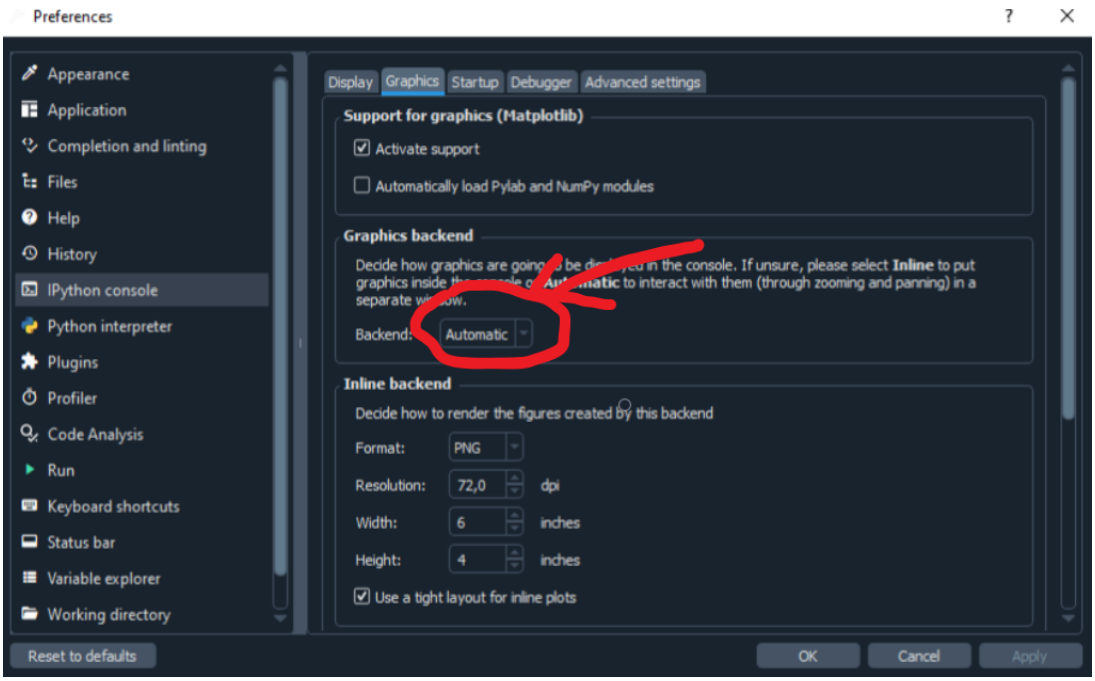


Figure 1: Sélectionner Graphics backend = Automatic dans les préférences de Spyder pour afficher les figures sur des fenêtres graphiques indépendantes. Les préférences de Spyder sont disponibles dans l'onglet Tools (ou Outils).

2.2 Enoncés des TP Python

Vous disposez de 5 séances pour réaliser les 4 travaux pratiques ci-dessous. Le plus long TP, celui sur les schémas différences-finies pour EDP stationnaires 1D, peut nécessiter jusqu'à 2 séances.

Le TP 0 rassemble l'essentiel des fonctionnalités Python qui seront employées pour rélaiser les TP. Ce TP 0 a pour but de servir de "pense-bête" et de test d'utilisation des diverses commandes Python.

Travaux pratiques 0. ***(Les aide-mémoire Python)***

Le répertoire `AIDE_MEMOIRE_PYTHON` contient des scripts qui résument l'essentiel des fonctionnalités Python utilisées pour réaliser les TPs d'analyse numérique, et donnent de nombreux exemples :

- Script `aide_memoire_graphisme.py` : pour réaliser des figures avec la bibliothèque Matplotlib https://matplotlib.org/. Les fonctions tracées sont importées à partir du fichier `fonctions_aide_memoire_graphisme`.
- Script `aide_memoire_python.py` : pour les bases du langage Python (boucles, tableaux Numpy https://numpy.org/, manipulation de fichiers, ...)

Ces scripts incluent des références Web pour approfondir les différentes notions abordées. les références les plus utilisées sont :

- https://courspython.com/
- https://www.w3schools.com/python/default.asp
- https://www.w3schools.com/python/numpy/default.asp

Travaux pratiques 1. ***(Intégration numérique par méthode des trapèzes)***

Le but du TP est de calculer une approximation numérique I_h de l'intégrale (vue en TD du L3 MFE) :

$$I = \int_1^{\pi} \frac{\sin x}{2x^3}\,\mathrm{d}x \simeq 0.198557\ldots, \tag{1}$$

à l'aide de la méthode des trapèzes, et de comparer la valeur numérique I_h avec la valeur exacte en vérifiant que l'erreur numérique :

$$E_h = |I - I_h|, \tag{2}$$

tends vers zéro lorsque $h \to 0$. Le répertoire de travail Spyder est `TP1_L3_QUADRATURE`, et le script principal est `main_TP_quadrature.py`.

1) Ecrivez la fonction Python :

```
def trapeze(a, b, f, n):
# ...
    return Ih
```

qui calcule l'intégrale $I_h = \int_a^b f(x)\mathrm{d}x$ par la formule du trapèze composite avec n sous-intervalles de taille uniforme h[3].

2) Validez votre fonction `trapeze` en vérifiant qu'elle donne l'intégrale exacte d'une fonction linéaire (polynôme de degré 1).

3) Vous allez maintenant utiliser la fonction `trapeze` pour calculer l'intégrale I donnée par l'équation (1).

   a) A l'aide de la bibliothèque de calcul symbolique Sympy[4], calculez une valeur précise (notée `Iex`) en tapant ces lignes dans votre programme Python :

```
import sympy as sp # Calcul symbolique avec sympy
t = sympy.symbols('t')
fex = sp.sin(t)/(2*t**3)
Iex = sp.integrate(fex, (t, 1, sp.pi)).evalf()
```

   b) Vérifiez que $E_h \to 0$, lorsque $h \to 0$, c'est-à-dire lorsque n augmente. Pour des valeurs de h suffisamment petites ($n \geq 20$), calculez le facteur de réduction FdR[5] de la méthode du trapèze. Comparez-le avec sa valeur théorique[6].

   c) Pour $n \geq 50$, représentez en échelle logarithmique l'erreur numérique E_n de la méthode du trapèze (voir les exemples données dans le script `aide_memoire_graphisme.py`) et montrez qu'elle est précise au second-ordre en traçant sur le même graphe la fonction $n \mapsto 1/n^2$ comme sur la figure 2. Pour cela, il est conseillé d'écrire dans un fichier csv plusieurs réalisations de la fonction `trapeze` sous la forme :

[3] On rappelle que, d'après le cours, cette formule s'écrit :

$$I_h = \sum_{k=0}^{n-1} h\left(\tfrac{1}{2}f(x_k) + \tfrac{1}{2}f(x_{k+1})\right),$$

avec $h = (b-a)/n$ et $x_k = a + kh$.

[4] La page web de Sympy est https://www.sympy.org/

[5] La définition du TD est FdR $= E_h/E_{h/2}$ où E_h est donné par (2).

[6] D'après le TD, si h est suffisamment petit alors $FdR \simeq 2^q$ pour une méthode d'ordre de précision q.

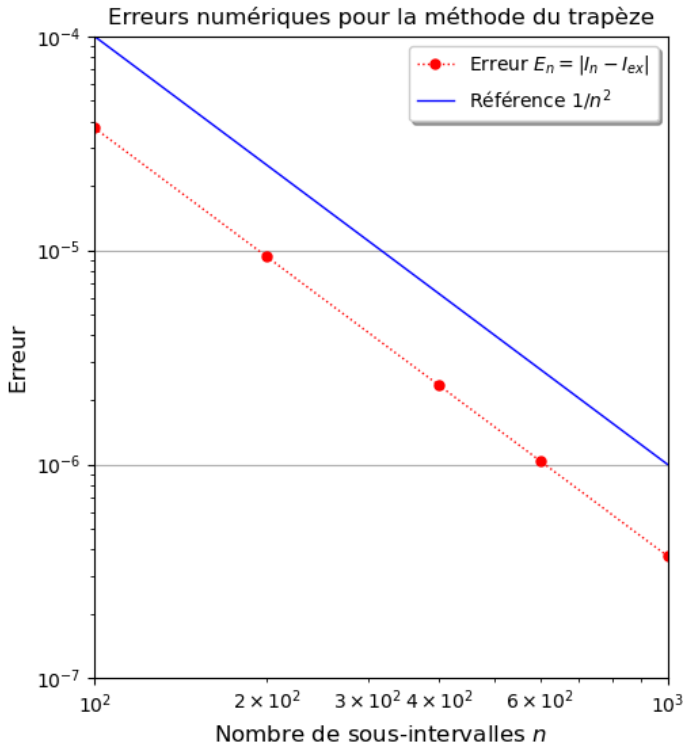


Figure 2: Résultats de la méthode du trapèze pour l'intégrale (1). Pour afficher les figures dans des fenêtres indépendantes de l'IDE, il faut sélectionner Graphics backend = Automatic dans les préférences de Spyder (*cf.* figure 1).

```
n1  E_n1
n2  E_n2
n3  E_n3
etc ...
```

puis lire les données du fichier dans un script Python, et tracer les données lues (un exemple est donné dans le script `aide_memoire_python.py`).

Travaux pratiques 2. ***(Différences-finies pour EDP stationnaires 1D et implémentation de conditions aux limites de neumann)***

Ce TP concerne la discrétisation DF de problèmes stationnaires 1D, de type convection-diffusion (exercice A.8) ou Helmholtz (exercice A.9), de façon à prolonger les notions vues dans ces exercices. Le répertoire de travail Spyder du TP est `TP2_M1_CONV_DIFF_STAT`.

TP2, exercice 1. *(Equation de convection-diffusion stationnaire)*

Cette partie concerne le problème aux limites avec conditions aux limites de Dirichlet :

$$a\frac{\mathrm{d}u}{\mathrm{d}x} - \nu\frac{\mathrm{d}^2u}{\mathrm{d}x^2} = 0, \quad \text{dans } \Omega = [a_x, b_x], \tag{1a}$$

$$u(a_x) = u_\mathrm{g}, \qquad u(b_x) = u_\mathrm{d}. \tag{1b}$$

En utilisant les données de l'exercice A.8, soit $[a_x, b_x] = [0, 1]$, $u_\mathrm{g} = 0$ et $u_\mathrm{d} = 1$, la solution exacte de (1), représentée par la fonction `python uex_convdiff`, exhibe une couche limite près du bord droit $b_x = 1$ lorsque le nombre de Péclet $\mathrm{Pe} = aL/\nu$ est grand.

Le script `main_TP_conv_diff_stat.py` résout cette EDP pour $a = 1$ et $\nu = 0.02$ (variables Python `adv` et `nu` respectivement) par méthode de différences-finies centrées (fonction `schema_centre_convdiff_stat`). Ensuite, il trace sur une même figure le graphe de la solution numérique (vecteur `Unum`) aux points du maillage (`Xmesh`), et celui de la solution exacte `Uex` sur le maillage fin `Xfine`.

Question 1. *Exécutez le script pour $n = 11$ points et calculez le nombre de Péclet de maille Pe_h. Que remarquez-vous ? Calculez le nombre minimal de points pour que les problèmes numériques disparaissent. Exportez les graphes nécessaires pour le compte-rendu final.*

On veut maintenant vérifier numériquement l'ordre de précision de la méthode de différences finies centrées, en calculant pour plusieurs valeurs du pas d'espace Δx la norme l^2 de l'erreur :

$$\|u_{\text{ex}} - u_h\|_{l^2}, \tag{2}$$

où $\|\cdot\|_{l^2}$ représente l'erreur quadratique moyenne aux points intérieurs du maillage, définie par :

$$\|\phi\|_{l^2}^2 = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)^2. \tag{3}$$

Cette norme l^2 se calcule aisément avec la fonction Python `linalg.norm()` qui calcule la norme d'un tableau Numpy.

Question 2. *Pour la méthode FD centrée, calculez l'erreur moyenne (2) de la solution numérique. Vérifiez que l'erreur tends vers zéro lorsque n devient grand et, graphiquement, que la solution numérique tende vers la solution exacte. Calculez le FdR et vérifiez que l'ordre de précision théorique est atteint.*

Question 3. *Programmez la méthode FD upwind (vue en TD) d'une façon similaire à la méthode centrée. Rendez le listing Python dans votre compte-rendu. Montrez que l'erreur moyenne tends vers zéro lorsque n devient grand. Par expérimentation numérique, vérifiez que la solution numérique ne présente jamais d'oscillations numériques, quelle que soit la valeur de Pe_h. Montrez que le FdR correspond à l'ordre de précision théorique vu en TD.*

Question 4. *Comparez sur le même graphe (en coordonnées logarithmiques, cf. TP sur la méthode des trapèzes) l'erreur moyenne des méthodes centrées et upwind. Tracez aussi sur ce graphe leur ordre de précision théorique de type $n \mapsto 1/n^q$.*

TP2, exercice 2. *(EDP de Helmholtz et conditions aux limites de Neumann)*

Cet exercice concerne l'EDP de Helmholtz de l'exercice A.9, qui est définie dans $\Omega =]-1, 1[$ avec les condition aux limites mixtes de type Neumann (au bord gauche) et Dirichlet (au bord droit) :

$$\lambda u - \frac{\mathrm{d}^2 u}{\mathrm{d}x^2} = f(x), \qquad \lambda \geq 0, \tag{1a}$$

$$\frac{\mathrm{d}u}{\mathrm{d}x}(-1) = \sqrt{\lambda}/\tanh(2\sqrt{\lambda}), \qquad u(1) = 1. \tag{1b}$$

Pour $f(x) = \lambda$, la solution exacte de (1) s'écrit :

$$u_\lambda(x) = 1 - \frac{\sinh\big(\sqrt{\lambda}\,(1-x)\big)}{\sinh(2\sqrt{\lambda})}. \tag{2}$$

Lorsque $\lambda \gg 1$, on peut observer que $u_\lambda(x) \sim 1$ dans tout le domaine sauf dans un voisinage du bord $x = -1$, de taille $\mathcal{O}(1/\sqrt{\lambda})$, où la solution exhibe une couche limite.

Le but de l'exercice est de comparer la précision numérique de diverses variantes d'implémentation d'une conditions aux limites de Neumann dans un schéma FD. Les trois variantes qui ont été vues en cours et dans l'exercice A.9 sont :

I) Méthode de l'exemple 7.3 du cours,

II) Méthode de l'exemple 7.4,

III) Méthode du point fantôme vue en *TD*.

Dans cet exercice, vous allez écrire les programmes Python pour chaque variante, en vous inspirant des programmes de l'exercice sur l'équation de convection-diffusion. Vous comparerez la précision des 3 méthodes et ferez le lien avec l'ordre de précision théorique donnée par l'erreur de troncature du TD. Cette étude se fera pour le cas $\lambda = 100$.

Question 1. *Ecrivez une fonction Python pour chaque méthode. Rendez le listing Python dans votre compte-rendu. Vérifiez (graphiquement et par calcul de l'erreur l^2) que la solution numérique converge vers la solution exacte lorsque n augmente. Calculez le FdR de chaque méthode. Comparez l'ordre de précision obtenu avec l'ordre de l'erreur de troncature vu en TD.*

Question 2. *Comparez sur une même figure les estimations d'erreur l^2 de chaque méthode. Sont-elles en accord avec leur erreru de troncature ? Distinguez la meilleure de ces trois méthodes.*

Question 3. *En utilisant les graphes de la question précédente, estimez le nombre de nœuds nécessaires pour que le schéma du premier ordre obtienne une précision (en norme l^2) comparable à celle donnée par la méthode III avec $n = 100$ points.*

Travaux pratiques 3. ***(Schémas Numériques pour les Equations Hyperboliques)***

Ce TP concerne la résolution numérique de l'équation de transport linéaire 1D :

$$\frac{\partial u}{\partial t} + a\,\frac{\partial u}{\partial x} = 0 \tag{3a}$$

dans le domaine $\Omega \times \Omega_t =]a_x, b_x[\times]0, t_\mathrm{f}[$, avec la condition initiale :

$$u(x, t = 0) = u_0(x), \quad \forall\, x \in \Omega. \tag{3b}$$

On considère le cas $a > 0$, et la condition d'*inflow* est :

$$u(a_x, t) = g_-(t), \quad \forall\, t \in \Omega_t. \tag{3c}$$

On rappelle que la solution $u(x, t)$ du problème (3) correspond à une translation uniforme de vitesse a de la condition initiale $u_0(x)$, soit :

$$u(x, t) = u_0(x - at). \tag{4}$$

TP3, exercice 1. *(Condition initiale régulière)*

Le répertoire de travail Python du TP est `TP3_M1_EDP_HYPERBOLIQUE`. Dans ce premier exercice, on considère des solutions de l'EDP (3) qui restent continues en tout instant $t \geq 0$.

On choisit comme domaine spatial $\Omega =]0, 1[$, et le coefficient d'advection de l'EDP est $a = 1$ (variable `adv` dans le code Python). La condition initiale à $t = 0$ qui est définie comme :

$$u_0(x) = \exp\left(\frac{1}{2}\left[\frac{x - x_0}{\sigma}\right]^2\right), \tag{1}$$

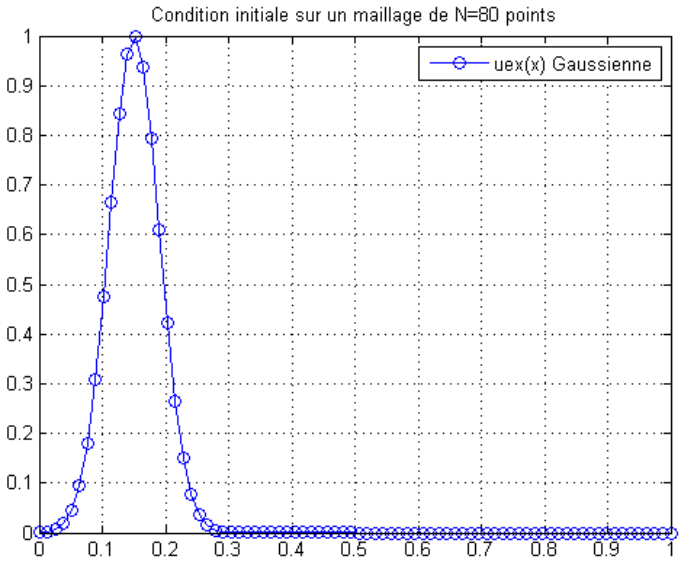


Figure 3: Condition initiale (1) avec $\sigma = 0.04$ et $x_0 = 0.15$.

.

correspond à une gaussienne d'écart type σ, centrée au point x_0 (*cf.* figure 3). Plus la valeur de σ est grande, plus le problème est raide (la solution possède de forts gradients).

Remarque 1. *L'équation de transport (3) étant instationnaire, il faut déterminer la valeur du temps d'intégration finale t_f. Dans le script Python, on définira*

$$t_f \approx \frac{b_x - a_x}{a},$$

de façon que la solution soit sortie du domaine Ω à la fin de l'intégration en temps.

Dans cet exercice, vous allez comparer qualitativement divers schémas pour la résolution de l'équation de transport (3). Les schémas considérés sont les suivants :

- Schéma upwind (*cf.* cours),
- $\theta-$schéma pour $\theta \in [0,1]$ (vu en TD pour l'EDP de la chaleur) :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\left[\theta\, \delta_x^0 u_i^{n+1} + (1-\theta)\, \delta_x^0 u_i^n\right] = 0, \tag{2}$$

- Schéma de Lax-Wendroff :

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + a\, \delta_x^0 u_i^n - a^2 \frac{\Delta t}{2} \delta_{xx}^0 u_i^n = 0. \tag{3}$$

Le schéma upwind est déjà programmé (fonction `schema_upwind(n, c, uold, ug)`). Ces divers schémas seront exprimés en fonction du nombre de courant C uniquement (variable `cfl` dans le script principal). On rappelle la définition de C :

$$C = \frac{a\Delta t}{\Delta x}. \tag{4}$$

Fixer dans le programme la valeur de C plutôt que celle de Δt permet de s'assurer que la condition CFL des schémas numériques est vérifiée.

Remarque 2. *L'EDP hyperbolique (3) (qui modélise des écoulements non visqueux) est bien posée (la solution*

existe et elle est unique) avec la seule condition de bord à l'inlet (3c), contrairement aux EDP paraboliques (pour fluides visqueux) qui nécessitent en outre une condition à l'outlet.

En accord avec la théorie des EDP hyperboliques, les schémas décentrés de type upwind ne nécessitent pas de conditions à l'outlet. Par contre les schémas centrés, tels le $\theta-$schéma (2) ou le schéma de Lax-Wendroff (3), nécessitent de définir une condition à l'outflow. Un moyen simple de définir une condition de bord artificielle en $x = b_x$ est la technique du point fantôme discutée dans le CM.

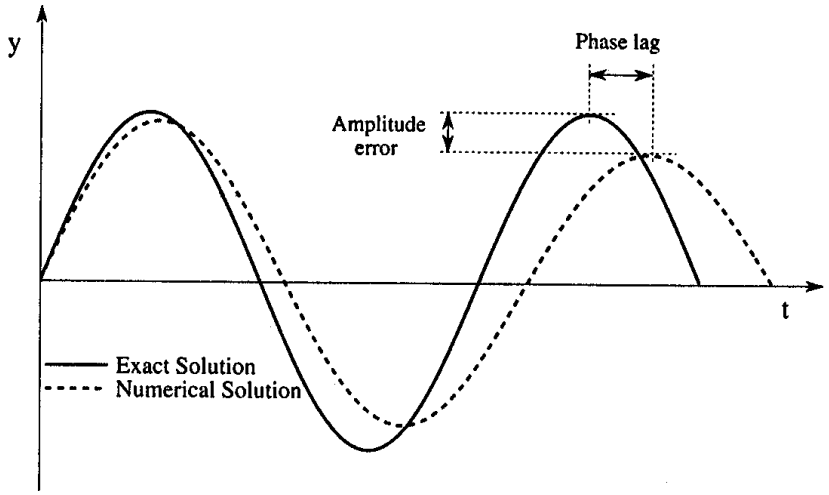


Figure 4: Décomposition de l'erreur numérique en une composante d'erreur d'amplitude (ou dissipation numérique) et une composante d'erreur de phase (ou dispersion numérique).

Le but de votre travail est d'observer par la pratique les notions qui ont été vues en cours et en TD par l'étude de l'erreur de troncature et de l'équation équivalente, *c.à.d* :

- Stabilité,
- Dissipation + dispersion numérique (*cf.* figure 4),
- Oscillations parasites.

Pour chacun des schémas, il s'agira d'évaluer qualitativement ces propriétés et d'en effectuer une synthèse. A la différence du *TP* précédent, vous ne ferons pas de raffinement de maillage : un maillage de $n = 80$ points est suffisant pour représenter spatialement la solution numérique. Il est conseillé de prendre des notes manuscrites lors de l'évaluation de chacun des schémas et de sauvegarder les différentes figures que vous créerez.

Question 1. *Le script principal* `main_TP_edp_hyperbolique.py` *utilise le schéma upwind pour résoudre l'équation de transport pour la condition initiale continue (1) avec $\sigma = 0.04$ et $x_0 = 0.15$. Utilisez ce schéma avec $C = 1.2$. Que se passe-t-il ? Comment remédier au problème ? Observez et évaluez les phénomènes de dissipation et dispersion numérique.*

Pour les questions suivantes, la comparaison des schémas se fera pour une même valeur du nombre de Courant C, choisie légèrement en dessous de la condition CFL.

Question 2. *Utilisez le $\theta-$schéma (fonction* `schema_theta(n, c, uold, ug, theta)`*) avec $\theta = 0$. Donnez le nom de ce schéma. Quels sont les résultats obtenus ? Pouviez-vous le prévoir ?*

Question 3. *$\theta-$schéma avec $\theta = 1/2$. Le reconnaissez-vous ? Quel est son ordre de précision ? Utilisez le avec $C = 0.8, 1.2, 2$, etc ... Quelles sont ses propriétés de stabilité ? Pouviez-vous les prévoir ? Observez et évaluez les phénomènes de dissipation et dispersion numériques.*

Question 4. *$\theta-$schéma avec $\theta = 1$. Même questions que précédemment.*

Question 5. *Programmez le schéma de Lax-Wendroff. Rappelez l'ordre de précision de la méthode. Retrouve-t'on les propriétés de stabilité théoriques ? Comparez ses propriétés principale (précision, dissipation et dispersion numériques) par rapport au schéma upwind.*

Question 6. *Synthèse des résultats : parmi les schémas que vous venez d'utiliser, énoncez selon-vous les trois meilleurs et justifiez brièvement pourquoi vous rejetez les autres. Pour les trois meilleurs schémas, rappelez leurs propriétés et avantages et énoncez un classement subjectif selon ce que vous avez observé.*

TP3, exercice 2. *(Condition initiale discontinue)*

On rencontre souvent des ondes de choc[7] dans les écoulements de fluides compressible, notamment dans le domaine de l'aérodynamique supersonique. Ces écoulements sont modélisés par des EDP hyperboliques qui admettent alors des solutions discontinues. Ces discontinuités sont générées par la non-linéarité de l'EDP. Pour l'équation de transport (3) linéaire, il faut qu'au temps initial la solution soit discontinue, par exemple la solution initiale de Riemann :

$$u_0(x) = \begin{cases} 1 & \text{si } x < x_0, \\ 0 & \text{si } x > x_0. \end{cases} \tag{1}$$

Cette condition initiale se programme facilement avec la fonction `np.heaviside(x, x0)`[8] de la bibliothèque Numpy.

Il est difficile de conserver au travers d'un choc la précision des schémas numériques qui est observée pour des solutions continues. Néanmoins, il est important de développer des méthodes numériques pour la mécanique des fluides telles que :

- une grande précision est obtenue loin du choc,
- il y a peu d'oscillations parasites près des discontinuités,
- La solution numérique conserve un profil de choc nettement défini, et non atténué par une trop forte viscosité artificielle.

Question 1. *Comparez les résultats des meilleurs schémas obtenus pour une solution continue lorsqu'ils sont appliqués à la solution de Riemann (1). Parmi les propriétés enoncées ci-dessus, lesquelles sont observées ? Conclusion + classement.*

Travaux pratiques 4. ***(Modélisation du transport d'un polluant par la méthode des volumes finis)***

Ce TP concerne la modélisation d'un polluant dispersé par un fluide comme représenté sur la figure 5 (gauche). Le transport de ce polluant est modélisé par une équation de convection-diffusion instation-

[7] https://fr.wikipedia.org/wiki/Onde_de_choc
[8] https://numpy.org/doc/stable/reference/generated/numpy.heaviside.html

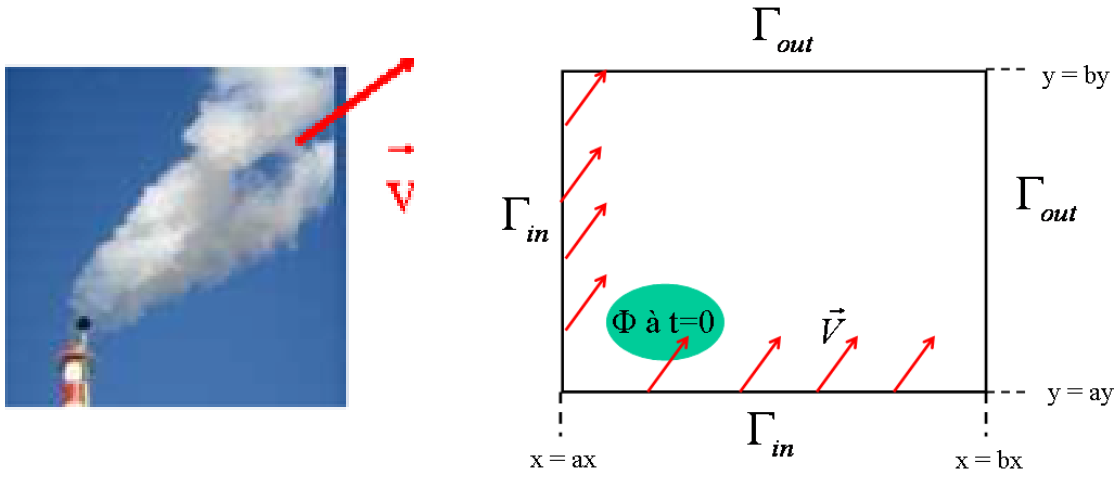


Figure 5: A gauche : dispersion de fumée dans l'atmosphère. A droite : Domaine de calcul Ω.

naire 2D qui sera discrétisée par la méthode des volumes finis. Le répertoire de travail Python du TP est `TP4_M1_FVM_2D`.

L'objectif du TP est de modéliser la dispersion de polluant (de fraction massique $\phi(x, y, t)$) dans le domaine fluide 2D représenté par le domaine de calcul $\Omega =]a_x, b_x[\times]a_y, b_y[$. L'équation d'équilibre pour ϕ traduit que la variation temporelle de la quantité de polluant $\int_\Omega \rho\phi \, \mathrm{d}V$ dans un volume élémentaire Ω (de volume V, de surface Γ, et de normale extérieure $\boldsymbol{n}$) est égale à un bilan de flux $\int_\Gamma \boldsymbol{F} \cdot \boldsymbol{n} \, \mathrm{d}S$ sur les faces Γ du volume élémentaire. Ce flux est modélisé par la somme d'un flux de diffusion :

$$\boldsymbol{F}_\mathrm{d} = \lambda \boldsymbol{\nabla}\phi, \qquad \text{où } \lambda > 0 \text{ est le coefficient de diffusion,} \tag{1}$$

et d'un flux de convection :

$$\boldsymbol{F}_\mathrm{c} = -\rho \boldsymbol{v} \phi, \qquad \text{tel que } \boldsymbol{v} \text{ est la vitesse du fluide.} \tag{2}$$

On suppose que la vitesse $\boldsymbol{v}$ du fluide est telle que représentée sur la figure 5, et on distinguera les frontières d'*inflow* Γ_in (telles que $\boldsymbol{v} \cdot \boldsymbol{n} < 0$) où sont imposées des conditions de Dirichlet homogènes ($\phi_{|\Gamma_\mathrm{in}} = 0$), et les frontières d'*outflow* Γ_out (telles que $\boldsymbol{v} \cdot \boldsymbol{n} > 0$) où le fluide transporte le polluant à l'extérieur du domaine de calcul, et où sont imposées des conditions de Neumann homogènes ($\frac{\partial \phi}{\partial \boldsymbol{n}}_{|\Gamma_\mathrm{out}} = 0$).

A l'instant initial $t = 0$, le polluant est placé à la position (x_0, y_0) du domaine de calcul et il est modélisé par :

$$\phi_0(x, y) = \mathrm{e}^{-(\frac{x-x_0}{\sigma_0})^2} \mathrm{e}^{-(\frac{y-y_0}{\sigma_0})^2}, \tag{3}$$

qui correspond à la gaussienne d'amplitude 1 et de rayon σ_0 représentée sur la figure 6.

Le domaine de calcul $\Omega =]a_x, b_x[\times]a_y, b_y[$ est discrétisé par un maillage uniforme centré en cellules :

$$\Omega_h = \{\Omega_{i,j},\, i = 1, \ldots, N_x,\, j = 1, \ldots, N_y,\}$$

de centre $\boldsymbol{x}_c(i, j) = (x_c(i), y_c(j))$, de volume $V_{i,j} = \Delta x \Delta y$. Pour imposer les conditions aux limites par la méthode du point fantôme, une couche de cellules est créée sur les parois West ($\Omega_{0,j},\, j = 1, \ldots N_y$), East ($\Omega_{N_x+1,j},\, j = 1, \ldots N_y$), South ($\Omega_{i,0},\, i = 1, \ldots N_x$) et North ($\Omega_{i,N_y+1},\, i = 1, \ldots N_x$). Ce maillage peut être visualisé en exécutant le script principal `main_TP_fvm_2D.py`.

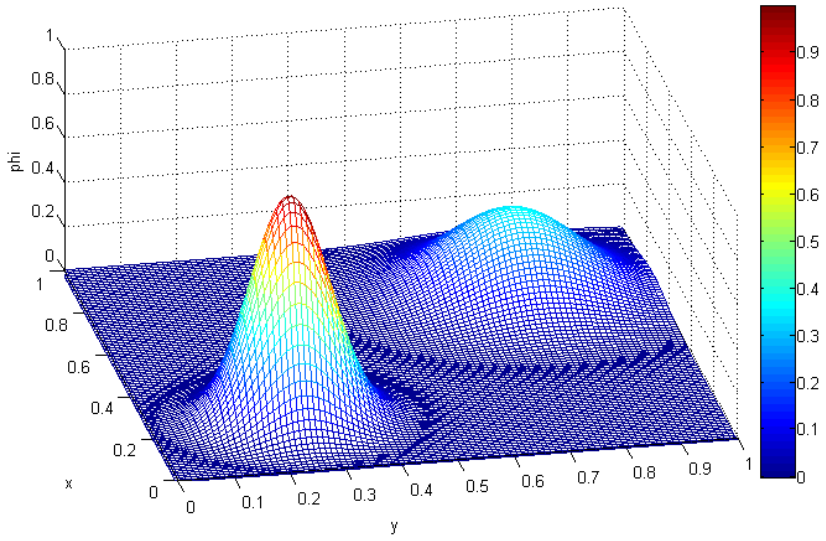


Figure 6: Solution initiale (et finale) du transport d'une gaussienne placée en $(x_0, y_0) = (1/4, 1/4)$).

Question 1. *Ecrivez la forme intégrale de l'équation d'équilibre de ϕ dans une cellule $\Omega_{i,j}$ du maillage, en notant $\eta = \lambda/\rho$ le paramètre de diffusion.*

Question 2. *Faites un schéma du maillage complet en positionnant les variables discrètes $\{\phi_{i,j}\}$. Donnez la valeur que prend $\phi_{i,j}$ dans les cellules fantômes selon le type de conditions aux limites.*

Question 3. *Effectuez une semi-discrétisation de l'équation de convection-diffusion pour ϕ par méthode de volumes-finis centrés en espace. Achevez-la en utilisant le schéma d'Euler progressif (EP1). Ecrivez le schéma complet sous sa forme itérative.*

Le script `main_TP_fvm_2D.py` crée le maillage Ω_h, discrétise la condition initiale (3) (donnée par la fonction `phi_init.py`) aux nœuds $\boldsymbol{x}_c(i,j)$ du maillage, et la représente sur un graphe (fonction `plot_surface`[9]). Vous allez compléter ce script principal en écrivant la boucle en temps qui discrétise l'équation de transport avec le schéma en temps EP1 et une discrétisation spatiale par méthode FVM centrée. Vous réaliserez une simulation complète du transport du polluant, avec une animation de la solution, en reprenant les étapes principales du TP sur l'équation de transport 1D.

Question 4. *Ecrivez la fonction*

```
def schema_EP1_centre(advx, advy, nu, dt, dx, dy, ncvx, ncvy, phiold):
# ...
    return phinew
```

qui correspond à une itération en temps du schéma d'Euler progressif.

Les variables d'entrée et de sortie de cette fonction sont :

- les réels `advx` et `advy` correspondent aux composantes cartésiennes de la vitesse d'advection $\boldsymbol{v}$, supposée constante.
- Le tableau `phiold` de taille $(ncvx+1)\times(ncvy+1)$ doit contenir la solution au temps t_n avec conditions aux limites dans les cellules fantômes.

[9] https://www.geeksforgeeks.org/3d-surface-plotting-in-python-using-matplotlib/

- Le tableau `phinew` de taille $(ncvx+1)\times(ncvy+1)$ (initialisé par `np.zeros( (ncvx + 2, ncvy + 2) )`) donnera la solution au temps $t_{n+1} = (n+1)\Delta t$.

La simulation que vous effectuerez prendra les paramètres suivants pour le domaine spatial :

$$a_x = 0, \qquad b_x = 1, \qquad a_y = 0, \qquad b_y = 1.$$

La condition initiale prend les paramètres suivants :

$$\sigma = 0.1, \qquad (x_0, y_0) = (0.25, 0.25),$$

et les paramètres de l'équation de convection diffusion sont :

$$\eta = 0.01, \qquad \boldsymbol{v} = (1, 1).$$

Vous effectuerez vos simulations avec un maillage d'une centaine de cellules dans chaque direction.

Question 5. *La question suivante vous permettra de fixer les paramètres de l'intégration en temps.*

- *Choisissez un intervalle de temps $[0, t_f]$ telle que le polluant soit sorti du domaine au temps final $t = t_f$.*
- *Etablissez les conditions de stabilité du schéma EP1 centré (cf. poly, proposition 15.7) pour les paramètres de la simulation.*

Question 6. *Programmez dans le script principal la boucle d'intégration en temps du schéma EP1 centré, et faite une animation de la solution numérique. Inspirez-vous au maximum de ce qui a été fait dans le TP sur l'équation de transport 1D. Notez-bien que dans la fonction* `update_plot_movie`*, le tracé de la solution est similaire à celui de la solution initiale.*

Question 7. *Effectuez des simulations pour une valeur de Δt supérieure et inférieure à la condition de stabilité. Qu'observez-vous ? Dans le compte-rendu, montrez le graphe de la solution à des temps bien choisis, et rendez le listing des programmes* Python *que vous avez écrit.*

Lab Exercice on Computational Fluid Dynamics

First steps with ANSYS FLUENT: The axysimmetric expansion flow

Instructor: Olivier Botella `olivier.botella@univ-lorraine.fr`

1 Problem statement

The purpose of this lab exercise[1] is to learn the basics of CFD analysis with the commercial software ANSYS FLUENT[2] by computing the axisymmetric expansion flow presented in the lectures (Fig. 1).

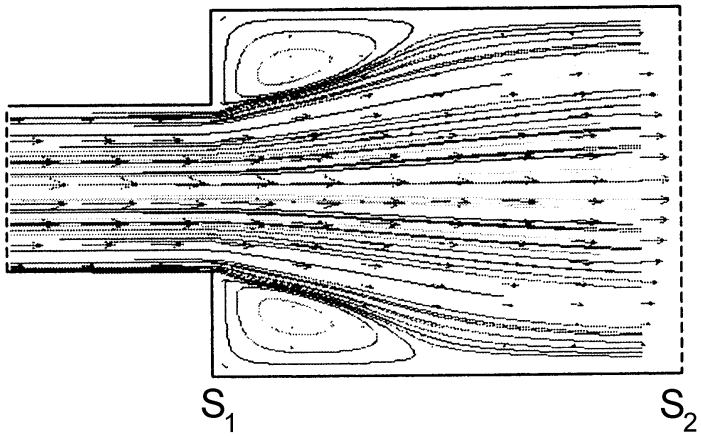


Figure 1: Velocity vectors at the vicinity of the sudden expansion (section S_1 at $x_1 = 1.0\,m$). In the upstream pipe the flow is fully developped, then the singularity causes abrupt flow separation, which causes recirculation zones. In the downstream pipe, the flow reattaches at Section S_2 and reaches fully developed regime.

This analysis will be performed with the ANSYS WorkBench platform, which integrates the following softwares:

- Geometry creation (DesignModeler)
- Mesh generation (Ansys Meshing)
- Flow setup and numerical solution (Fluent)

[1] The pdf version of this document and all necessary documents for completing the exercise are available online at the course webpage: *UE 203 - Computational Fluid Dynamics M1 MEPP* at the ARCHE website (`https://arche.univ-lorraine.fr/course/view.php?id=9396`).

[2] This lab exercise has been performed with Version 15.0 of ANSYS WorkBench , and revised with V2025 R1.

Work to be performed

This document will introduce you to the basic functionality of the various WorkBench softwares and will guide you through the CFD analysis of the expansion flow.

Read carefully the document and execute with the WorkBench the tasks presented in the document. You should answer to the various questions of the document in a Word report. You will send me your report at the end of the CFD lectures as part of your evaluation.

2 The WorkBench platform

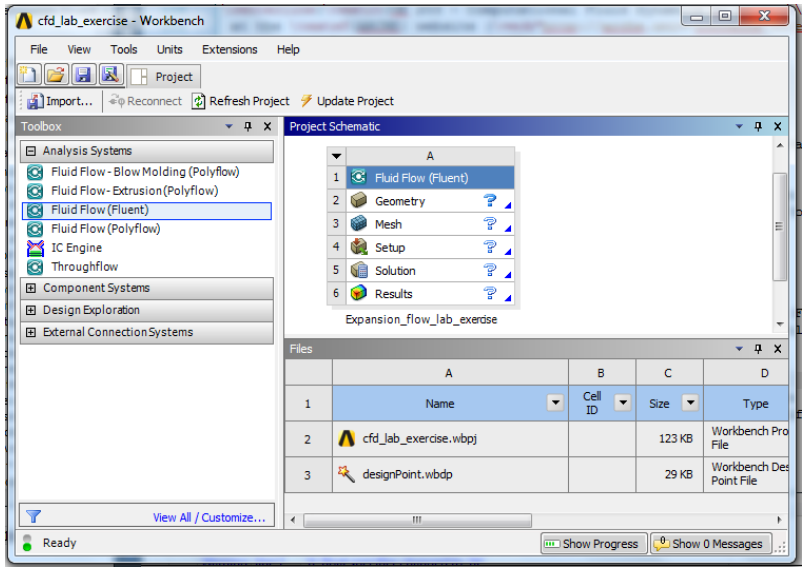


Figure 2: Graphical user interface (GUI) of ANSYS WorkBench .

Launch the ANSYS WorkBench software from your PC and create a Fluid Flow (Fluent) project. The Fluent workflow shown in Fig. 2 shows the consecutive steps of the CFD analysis. WorkBench will automatically manage the various data files generated by your Fluent project.

Save now the WorkBench project (**.wbpj* file) on your disk with a meaningful and correct names for your file and working directory (no special characters, foreign accents, avoid spacings), or the software may malfunction and your data be lost!

The language preferences of the various WorkBench softwares can be modified with the option Regional and Language Options from the menu bar: Tools → Options... You need to restart WorkBench to make the change effective.

3 Computational domain creation with DesignModeler

According to the symmetries of the flow domain of Fig. 1, we assume that the flow is asymmetric so that it is only required to create the computational domain shown in Fig. 3. These assumptions allow us to do 2D computations instead of 3D, that will save computational time and memory space.

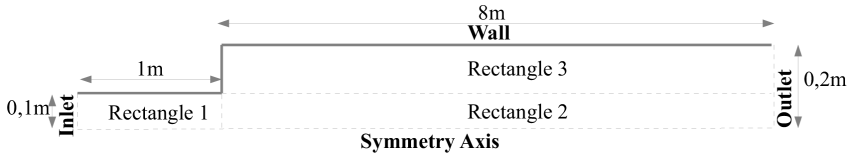


Figure 3: Computational domain and boundary conditions for the cylindrical expansion flow. The 2D computational domain is composed of 3 rectangular surfaces.

For axisymmetric flow computations with ANSYS Fluent , it is important to:

- Create the computational domain in the upper half XY plane (such as $y > 0$) of DesignModeler .
- Ensure that the symmetry axis of the domain is the horizontal axis ($y = 0$).

On WorkBench version 16.0 and above, the default geometry software may be SpaceClaim. To change the default software to DesignModeler , go to the WorkBench menu bar: Tools → Options... Geometry Import.

3.1 Overview of DesignModeler

The graphics interface (GUI) of DesignModeler is shown in Fig. 4. The most relevant components are:

- Graphics Window: sketching area where the various geometrical components of the computational domain are viewed and manipulated with the mouse.
- Menu Bar: where you can read/write the geometry files, create new sketches, *etc.*...
- Toolbars: The most important tools are:
  - View toolbars : icons for moving and zooming the geometry with the mouse in the graphics window.
  - Selection toolbars : Icons for mouse selection of the various elements of the geometry (vertex, edge, face and volume) in the graphics window.
  - New sketch : A "sketch" is a 2D drawing of the contours of the computational domain.
- Mode Tabs: there are 2 different modes:

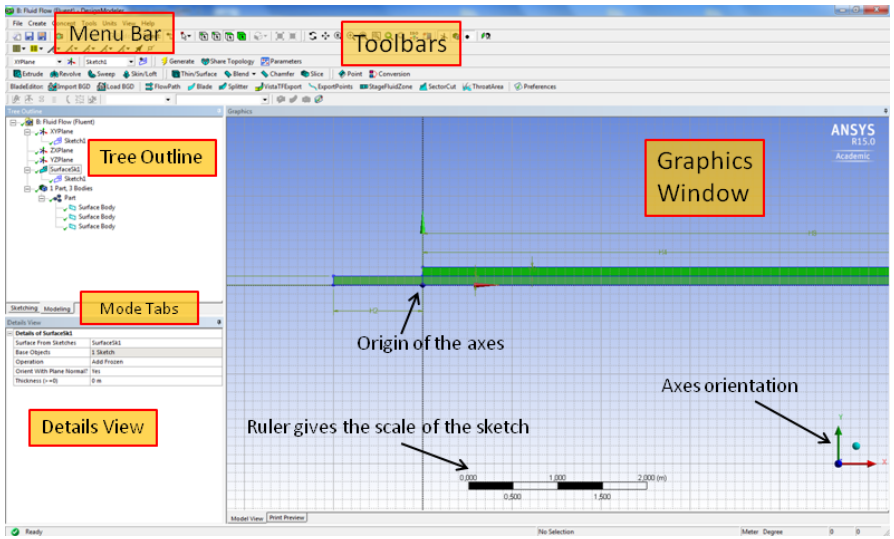


Figure 4: GUI of DesignModeler . Note that 2D geometries should be created in the XY plane. Click on the axes to change orientation of the graphics window.

  - Sketching mode: where you draw the contours of the geometry with the help of the various sketching tools shown in Fig. 5.
  - Modeling mode: that shows the Tree outline, where the basic operations for creating the geometry are shown sequentially: the default planes (where the sketches are stored), the geometry operations on the sketches, the list of created parts.
- Details View: where details of the selected geometry (dimensions, constraints, options, *etc.*...) are displayed and can be modified. Note that grey colored details cannot be modified.

It is important that you memorize the structure of this GUI, since you will use it extensively to do the lab exercises. Note that the GUI of Ansys Meshing is structured similarly.

3.2 Generation of the computational domain

DesignModeler creates the computational domain with the following consecutive steps:

Step 1) Sketching mode: draw the contours of the domain in one (or several) 2D sketches. It is easier to split the computational domain in several elementary entities (like the 3 rectangles of Fig. 3) and draw them separately.

Step 2) Modeling mode: create "solid" bodies (in 2D, it is a surface body) from these contours.

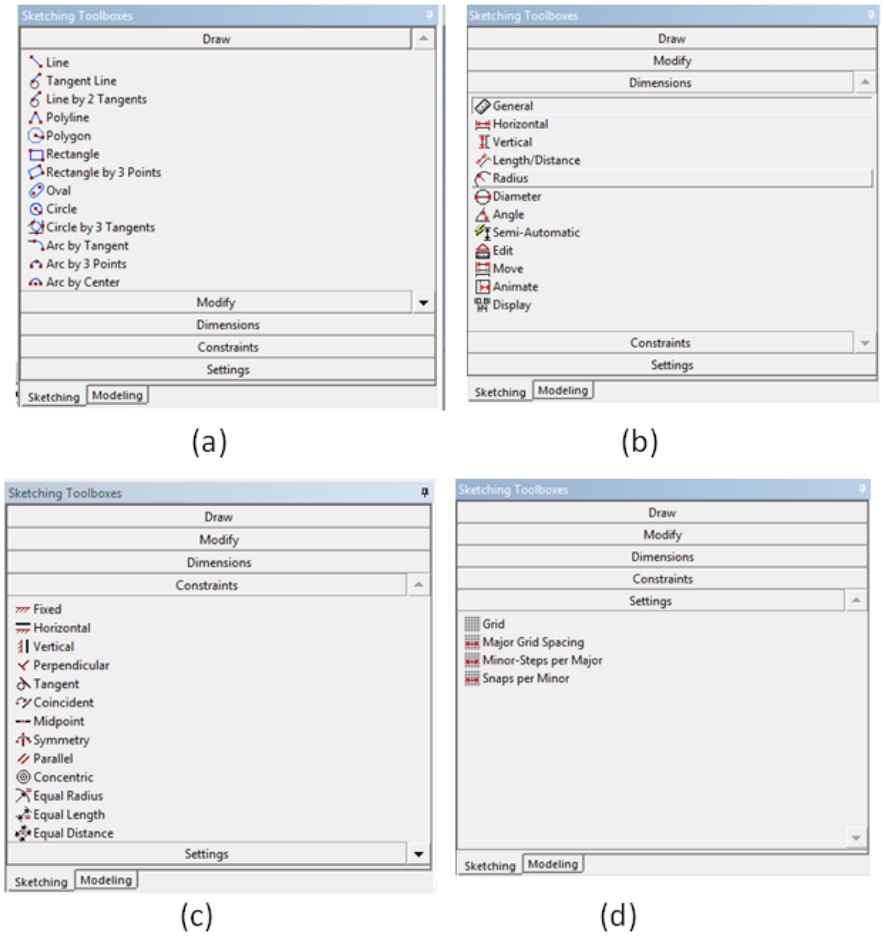


Figure 5: Toolbox available for sketching. (a) Draw: tools for drawing elementary 2D entities (lines, rectangles, circles, ...), (b) Dimensions: tools for setting the dimensions of the entities (by mouse selection in the graphic windows), (c) Constraints: tools for imposing constraints on each entities to assemble the final geometry (alignment with the axes, *etc.*...), (d) Settings: tools for displaying grid lines in the graphic window to make sketching easier.

Step 3) Modeling mode: combine these bodies into a unique computational domain (which is called a part in DesignModeler).

Apply now this strategy to create the domain of Fig. 3. In the same sketch sketch, you will draw each of the 3 rectangles shown in this figure, then assemble them as a unique part with 3 distinct surface bodies.

Note that the Undo/Redo tools are only available in the sketching mode. In the Modeling Mode, you can either Suppress a body[3] or Delete it permanently from the tree outline.

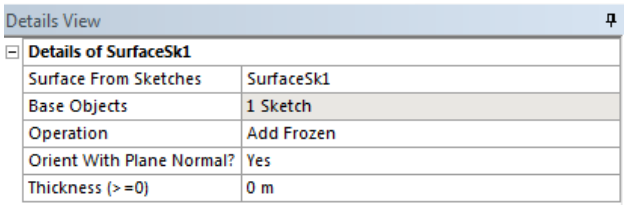


Figure 6: Base Objects: the 3 selected rectangles. Add Frozen (in French: Ajouter Corps Bloqué): it means that the three surfaces remain independent in the computational domain. You will be able to mesh each surface independantly in Ansys Meshing . Note that grey colored details cannot be modified.

0) In the Tree Outline of the modeling mode, click on XYPlane and create a new sketch . Go to the Settings toolbox (Fig. 5 (d)) to show the grid and apply adequate parameters for viewing the geometry: Major Grid spacing: 1 m, Minor-steps per Major: 10.

1) In the sketching mode, create Rectangle 1 with the Draw toolbox of Fig. 3 and set its dimensions[4]. Repeat for the other 2 rectangles. Align them as shown in Fig. 4 by using the Constraints toolbox[5].

2) In the Tree Outline of the modeling mode, select the sketch and click on Concept → Surfaces From Sketches from the menu bar. A surface sketch SurfaceSk1 is created in the tree outline, whose details are shown on Fig. 6. To finalize the operation click on Generate .

3) You should have created 3 parts made from the 3 surface bodies you have drawn (look in the Tree Outline). Now, to assemble the 3 bodies in the same part, select them all and right-click on Form New Part. As a result, you have created the computational domain as 1 part made of 3 surfaces bodies, as in the tree outline of Fig. 4.

4) The last step is to define the computational domain as "fluid". To do this, set Fluid/solid to be Fluid in the Details View of the part.

[3] It will be ignored when the geometry is generated.
[4] Define them with Dimensions toolbox and set their values in the Detail View.
[5] Use the Coincident constraint to stitch the edges of each rectangle to the X or Y axes.

If an element is highlighted with a yellow bolt in WorkBench, it means that the software waits for the user to update/generate this element.

Question 1. *Once the computational domain has been created, display it in your* Word *report and exit* DesignModeler.

When you exit the software, the computational domain is automatically saved on disk in the DesignModeler database format (**.agdb*), and ready to be used by the next software in the WorkBench workflow. You can view all files generated during the lab exercise (and their location on your hard-disk) by toggling View → Files in the WorkBench menu bar.

4 Mesh generation with Ansys Meshing

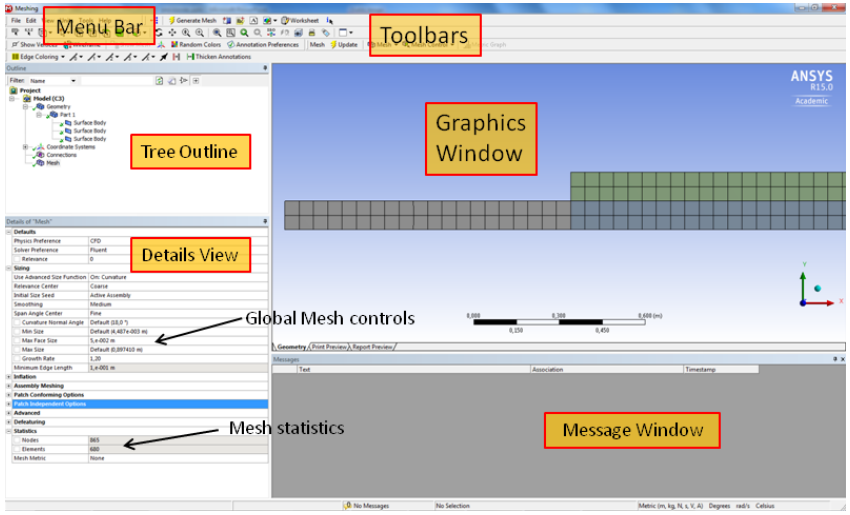


Figure 7: GUI of Ansys Meshing. The quad mesh shown in the graphics window has been generated with Element Size=0.05 m and all other parameters set to default values.

Open now Ansys Meshing from the WorkBench. The GUI, shown in Fig. 7, is similar to DesignModeler with the same mouse functionality. In this first lab exercise, we will only consider the generation of the most simple case of quadrilateral mesh, *i.e.* the Cartesian mesh (see the CFD lectures). Ansys Meshing is able to generate a mesh automatically, with limited user's input. For example, after clicking on the Generate Mesh tool, you should have to create the uniform mesh shown in Fig. 7. The only user input is the (Element Size in the Details View of the mesh).

Question 2. *Try generate meshes with various cell sizes. Display the resulting meshes in your* Word *report with the corresponding number of cells (*Elements*) and vertices (*Nodes*) given in the* Statistics *detail view of the mesh.*

4.1 Generation of a graded Cartesian mesh

The flaws of the "automatic" mesh of Fig. 7 is that the user does not control the mesh resolution in the vicinity of strong flow gradients, which are located in the boundary layers along solid walls and in the recirculation zone (see Fig. 1). Ansys Meshing allows various strategies for refining locally the mesh in selected areas of the computational domain. In this lab exercise, we will use the "bottom up" strategy to create a better quality mesh: first introduce a 1D meshing of the edges of the domain, then use a quad method to mesh the computational domain (which is called a Face, since it is 2D). The resulting mesh, shown in Fig. 8, is generated by the following steps:

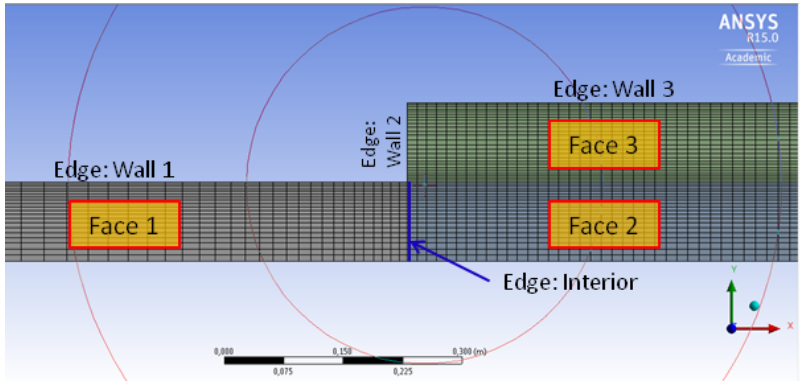


Figure 8: Zoom of the mesh near the singular expansion. Note that the mesh is refined near the walls and the salient corner. This mesh has 14300 cells.

Edge	Interior	Wall 1	Wall 2	Wall 3
# of divisions	20	40	25	300
Bias type				
Bias factor	3	5	2,5	4

Table 1: Sizing details for the edges shown in Fig. 8.

1) Edge meshing (Tool for edge selection): In the Graphics Window, select one of the edges of Face 1 and insert an edge sizing method[6]. The Edge Sizing method appears in the Tree Outline of the mesh. Choose Type: Number of Divisions and enter the details given in Table 1 (more informations are given at the end of the section). Repeat for the 3 other edges of Face 1.

2) Face meshing (Tool for face selection:) Complete the meshing of Face 1 by inserting a Mapped Face Meshing with Method: Quadrilaterals. Generate the mesh of Face 1 by right-clicking on Generate Mesh on Selected Bodies.

[6]Right-click on the moose: Insert → Sizing.

3) Preview the mesh by clicking on Mesh in the Tree Outline. If the mesh density is not similar to Fig. 8 or the grid lines are not parallel to the axes, you certainly made a mistake in some edge sizing details. In this case, delete the mesh[7], review your edge sizings and generate again the mesh of Face 1.

4) Repeat these steps for Face 2 and 3.

Figure 9: Exemples of graded mesh with geometric refinement. At left: the mesh is refined on the west boundary only. At right: refinement at both west and east boundaries.

Here are the main options of the Edge Sizing details panel.

- Number of Divisions corresponds to the number N of vertices of the edge.
- Choose Behaviour: Hard to disable global sizing options.
- Bias Type enables to create a graded mesh with geometric refinement of the form $\Delta x_{i+1} = R\Delta x_i$, where the Bias Growth Rate R is the ratio between successive cells. Examples of graded meshes are given in Fig. 9.
- Bias Option: the option Smooth Transition needs the value of R, otherwise Bias Factor corresponds to the ratio between the maximal and minimal cells of the edge, ie: $\Delta x_{\max}/\Delta x_{\min} = R^{1/(N-1)}$.

4.2 Definition of the boundary conditions

Now that the meshing of the computational domain is completed, the last step before starting the CFD analysis with the Fluent solver is to label the boundary conditions as in Fig. 3. To perform this, click on Model in the Tree Outline, then select a boundary edge in the Graphics Window and insert a Named Selection[8]. In the Tree Outline, rename it as shown in Fig. 3. The label of the boundary edges will be recognized by Fluent and the corresponding boundary condition will be applied.

For an easy reviewing of the boundary conditions in Fluent by the user (think that an actual 3D mesh may have hundreds of boundary edges), it is worthwhile to group all edges with the same boundary conditions (*e.g.* the 3 edge walls) in a single named selection.

Question 3. *Once the mesh is created and the boundary conditions labeled, display the mesh in your* Word *report, give the mesh statistics and exit* Ansys Meshing .

[7] Right-click on Mesh in the Tree Outline and select Clear Generated Data. This operation only deletes the mesh, not the meshing methods you have defined.

[8] Right-click on the moose: Insert → Named Selection.

5 CFD analysis with Fluent

Open Fluent from WorkBench . The Fluent Launcher panel will open. Select

Dimension : 2D, Solver Options : Double Precision

and leave the default settings for the other options. Note that:

- Solver Options : Choosing Double Precision allows floating point operations with a 16-digit precision. It is not recommended to use Single Precision because less accurate computations (8-digit accuracy) may impact the convergence of the iterative solution, even if it reduces the memory usage and yield faster computations.
- Parallel (Local Machine) : Depending on the number of processors (cores) available on your PC, you can select the number of processes used by Fluent to perform the computations in parallel (see Fig. 10). This exercise can be performed in serial (Solver Processes=1). In an ideal case, a parallel computation (Solver Processes ≥ 2) allows to divide the time of a serial computation by the number of processors.

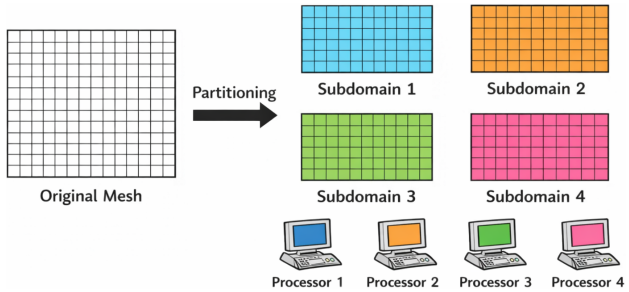


Figure 10: Example of mesh partitioning for parallel computation (MPI protocol). The computation in each of the 4 subdomains is performed in a separate processor.

5.1 Overview of Fluent

The graphics interface (GUI) of Fluent is shown in Fig. 11. The most relevant components are:

- Toolbars: As in DesignModeler and Ansys Meshing .
- Graphics Window: Graphical results are displayed here. The default mouse button functionality is[9]:
  - Left button translates.
  - Middle button zooms.

[9] Note that they are different from the other softwares in WorkBench . They can be changed from Display → Mouse buttons... in the menu bar.

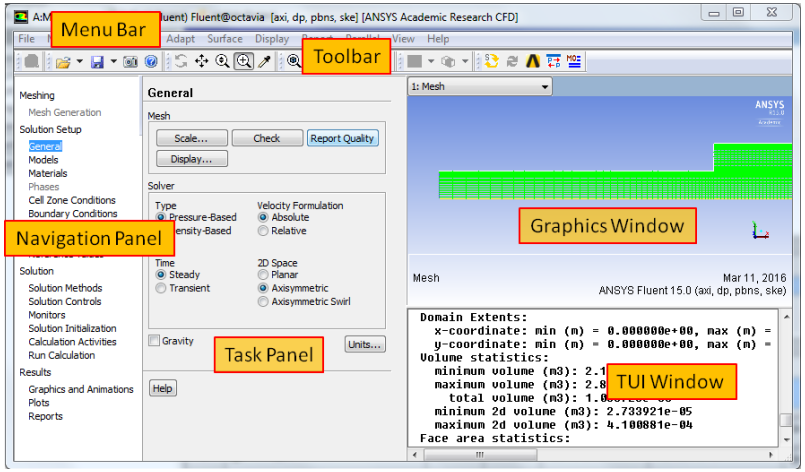


Figure 11: GUI of Fluent .

  - Right button selects/probes.

- Navigation Panel: The main stages of the CFD analysis are displayed on the left of the GUI from top to bottom:
  - Solution Setup: define the CFD problem, such as the fluid models (laminar, turbulent, heat transfer, multiphase, ...), type of materials (water, gas, solid, ...), boundary conditions.
  - Solution: define the numerical inputs (such as discretization schemes, solution algorithms, initial condition, ...) and solve the problem.
  - Results: once the solution is calculated, analyze (postprocess) the results.
- Task Panel: Selecting an item in the Navigation Panel opens the corresponding dialog box in this panel.
- Menu Bar: Additional items can be accessed from the menu bar.
- TUI[10] Window: display text results of the computation.[11]

5.2 Input/output files

There are 2 types of data files generated by Fluent .

- Case File *(*.cas)* : stores the mesh, the setup and the solution parameters of the simulation.

[10] TUI: Text User Interface.

[11] For advanced user: all the tasks defined from graphic panels have a corresponding TUI command, in order to be able to run Fluent in batch mode for automating the solver set up/solution/postprocess stages with scripts and journal files.

- Data File *(*.dat)* : stores the last computed solution.

You can manage these files from the menu bar File → Import or Export. You can also import a mesh generated from a third-party meshing software. Since you are using WorkBench , the mesh generated previously with Ansys Meshing is imported at the start of Fluent , while Case and Data files are automatically managed when you start or exit Fluent .

5.3 Solution setup

All the Fluent tasks for set-up of the CFD problem are done in the Solution Setup panel. Fluent has predefined default settings for the various stages of the set-up, but it is important to set them correctly for the problem under consideration. Thus, it is important to have knowledge of the physics involved in your CFD problem for selecting the relevant flow simplifications, models and equations to be solved.

Question 4. *We consider the flow of water ($\rho = 998.2\,kg/m^3$, $\mu = 0.001003\,kg/ms$) with inlet velocity $u_{inlet} = 1.5 \times 10^{-3}\,m/s$. Compute the Reynolds number in the upstream pipe (based on its diameter). Infer the flow regime and the relevant flow simplifications.*

Check the mesh and select the CFD solver: General task page

It is important to check the suitability of the imported mesh, as previous WorkBench operations may have created small errors that render the mesh unusable for Fluent computations.

Question 5. *Check the mesh by clicking on* Check *on the* Mesh *subpanel. Verify that the domain dimensions are the same as in Fig. 3 and report the mesh characteristics on your* Word *report.*

If the mesh check has failed, it may be caused by the fact that part of the domain falls below the upper half plane, see Sect. 3. A simple way to fix this problem is to translate the mesh[12] such that the symmetry axis is set at $y = 0$ (or slightly above).

As mentioned in the CFD lectures, there is no universal numerical method that works for all flow regimes. Choose the Solver Type according to:

- Incompressible flows are computed with the Pressure-Based Solver,
- Compressible flows at high Mach numbers are computed with the Density-Based Solver.

The other options are related to the problem simplification mentioned in Section 1: Time: Steady, 2D Space: Axisymmetric.

For large simulations, the computation time and the memory usage can be reduce by reordering the cell numbering of the mesh constructed by Ansys Meshing , in order to improve the solution of algebraic systems. To perform the domain reordering in Fluent , type the following command in the console window :

```
/mesh/reorder/band-width/reorder-domain
```

It will allow faster and more efficient computations, but will not affect the final results. Note that this reordering is done by default in parallel computations.

[12] Do it with Mesh → Translate... in the menu bar.

Set up the flow physics: Models task page

The Pressure-based solver is now configured with the relevant flow physics in the Models task page: turbulence models, flow with heat transfer (Energy, Radiation), combustion (Species), multiple phases (Multiphase), ...

It is of the upmost importance to select the model of the viscous stresses of the Navier-Stokes equations (Viscous item). Click on the Edit... button to review the various Viscous options:

- Inviscid : viscosity is assumed to be zero, which corresponds to the Euler equations, mainly used for external compressible flows.
- Laminar : corresponds to the classical Navier-Stokes equations with the Cauchy stress tensor. The viscosity model (constant, non-Newtonian or temperature dependent) is defined later in the Materials task page.
- Other options are used for turbulent computations, and correspond to choosing the modified viscous tensor resulting from RANS[13] modeling of the Navier-Stokes equations. We can mention the well-know k-epsilon model which is widely used in CFD: using this model will add 2 transport equations (for the turbulent kinetic energy k and dissipation ϵ) to the NS equations. The parameters for these transport equations can be set from these this panel.

The last versions of Fluent uses by default a RANS model with all other options (Multiphase, Energy, *etc.*...) disabled. Check your models carefully according to the flow physics.

Set up the material properties: Materials task page

Materials refer to the various fluids or solids involved in your CFD problem. The materials available for your simulation are listed on the Materials task page. By default, Fluent selects the Fluid material as air with constant properties (density and viscosity). Click on Create/Edit... to add water-liquid to the list of available materials (use the Fluent Database, where a wide range of materials are predefined). You can edit the material properties if needed, for example if you wish to use a non-Newtonian fluid with shear-thinning viscosity.

Set up the cell zone conditions

As stated in the lectures, a zone corresponds to a portion of the mesh defined as fluid, solid or porous, where distinct materials can be defined[14]. These zones have been defined previously in DesignModeler : in this first lab exercise, there is only one fluid zone filled with water. Click on Edit... in the Cell Zone Conditions task page to set this zone as water-liquid (by default Fluent fills all fluid zone with air).

[13] *RANS : Reynolds Averaged Navier-Stokes Equations*, obtained through statistical modeling of turbulent flows.

[14] As an example, the computational domain for a conjugate heat transfer problem would be divided into fluid and solid zones.

Set up the boundary conditions (BCs)

This task page defines the BCs settings at the boundary edges (in 2D) of the computational domain. The Zone panel displays the name of the boundary edges as you previously labelled them in Ansys Meshing [15]. You can visualize theses edges by clicking on Display Mesh... . Fluent usually recognizes these labels and assigns them with a corresponding BC Type. Make sure the the inlet boundary is defined as velocity-inlet, the symmetry axis as axis, and the outlet as pressure-outlet (see Fig 1). Some BCs needs additional settings which are defined by clicking on Edit... . For this lab exercise, you only have to set the inlet velocity given by Question 1.

5.4 Set up the numerical parameters of the computation

Now that the physical model, solver and boundary conditions have been set, the next step is to define the numerical parameters (algorithm for pressure-velocity coupling, spatial discretization, convergence criteria) of the pressure-based solver: this is performed with the Solution panel.

By default, Fluent chooses "robusts" numerical methods and parameters, which yield converged solutions for a wide range of flows. Henceforth, the default discretization methods are usually low-order and dissipative (for stabilizing the solution) and should be modified to get accurate results. The following sections gives guidelines for choosing these numerical methods. Since these methods are improved and their range of applicability expanded at each yearly releases of Fluent , the recommendations given at the time of writing (updated winter 2016) may be obsolete in the following years.

Set up the discretization methods: Solution methods task page

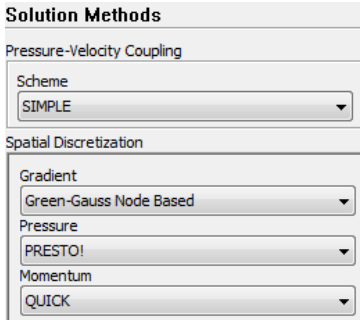


Figure 12: Recommended discretization methods for the expansion flow.

In this task page (shown in Fig 12) you first set up the iterative algorithm for the pressure-velocity coupling :

- SIMPLE[16] is the original algorithm for incompressible flow computations, still nowadays still in use for steady flows (single or multiphase).

[15] If some of the boundary edges are missing, it may be caused by a bug in the WorkBench . Ask advice to your instructor.

[16] Acronym for *Semi-Implicit Method for Pressure-Linked Equations*.

- Prefer PISO for time-dependent computations.
- Starting from ANSYS V19, the Coupled algorithm is proposed as default. This algorithm may reduce the computational time of SIMPLE or PISO, but has not been fully tested for all the physical models of Fluent .

Note that the use of a particular pressure-velocity algorithm does not change the final solution if convergence is reached. However, the convergence properties of the iterative solution (numerical instabilities, convergence rate and simulation time) are strongly depend to the chosen algorithm.

Next, define the various finite-volume (FV) discretization schemes for the NS equations[17] (Gradient: cell-face interpolation of velocity gradient used in the discretization of the viscous term and high-order discretization of the convection term, Momentum: discretization of convection term, Pressure: interpolation of cell-face pressure). The various pros and cons of each scheme has been discussed in the lecture notes. As a reminder:

- Avoid Green-Gauss Cell-Based interpolation which is the least accurate.
- Use PRESTO! for pressure interpolation in incompressible flows, or the solution may have convergence problems.
- For convection, use First Order Upwind solely at the initial stages of the computation or if the solution exhibits convergence problems. Otherwise, second-order schemes such as QUICK (more accurate and less diffusive) are preferred.

Set up the parameters of the iterative algorithm: Solution Controls task page

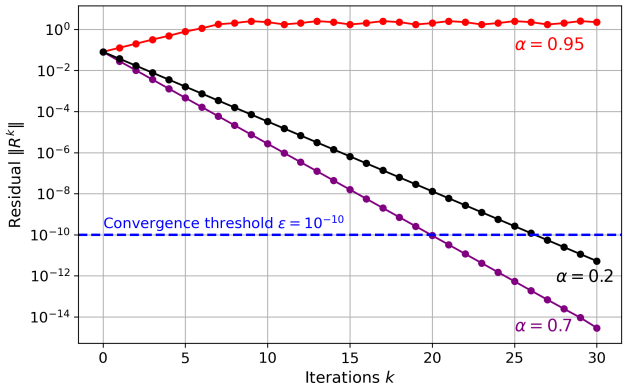


Figure 13: Effects of α on the convergence of an iterative algorithm for solving (1).

The basics of the iterative solution of agebraic equations in Fluent is explained for the linear system :

$$\mathcal{A}\Phi = F, \tag{1}$$

[17] When applicable, discretization schemes of the additional transport equations need to be defined too. Examples will be given for the turbulent computation of Section 6.

where Φ is the vector containing velocity or pressure mesh values, the matrix $\mathcal{A}$ and vector F represent the FV discretization of flow equations.

Starting from the Initial Solution Φ^0, an iterative algorithm seeks the solution to (1) as the series $\{\Phi^k,\, k = 1, \ldots, k_{\max}\}$ defined as:

$$\Phi^k = \Phi^{k-1} + \alpha\, \delta\Phi, \tag{2}$$

where k is the iteration number, $k_{\max}$ in the maximal number of iterations, and $\alpha \in [0, 1]$ is the Under-Relaxation Factor. Fluent considers that convergence is reached when the residual R^k of Eq. (1), defined as $R^k = F - \mathcal{A}\Phi^k$, verifies the Criterion:

$$\|R^k\| < \epsilon. \tag{3}$$

For the iterative solution of each flow equations (Momentum, Pressure, ...), you have to input the value of following parameters in Fluent :

- The initial condition[18] Φ^0,
- The convergence threshold ϵ,
- The under-relaxation factor $\alpha \in [0, 1]$,
- The maximal number of iterations $k_{\max}$.

The value of the under-relaxation factor α has impact on the rate of convergence of the algorithm, as illustrated in Fig. 13 : a large value of α gives faster convergence (compare $\alpha = 0.7$ and $\alpha = 0.2$), but a value too close to 1 yields non-convergence ($\alpha = 0.95$).

The Under-Relaxation Factors are set in the Solution Controls task page for each flow equation. The default values given by Fluent are *robusts*, *i.e.* they give a solution for a wide range of applications. There is no need to modify them unless your computation undergoes convergence problems : in this case, lower the under-relaxation factor of the equation that is not converging.

The set-up of Φ^0, ϵ and $k_{\max}$ is described next.

Monitoring the convergence of the iterative algorithm: Solution controls task page

During the simulation, Fluent allows you to monitor the evolution of the residual R^k in a graphic window as in Fig. 14. The monitoring parameters of each flow equations are set in the Monitors task page. By default, Fluent defines the Absolute Criteria of (3) as $\epsilon = 10^{-3}$ for all equations[19]. Often in practical applications, this condition is not sufficiently restrictive to give an accurate solution. To change this criterion, click on Edit... in the Residuals panel (see Fig. 14).

5.5 Perform the flow computation

The computations are performed in the Solution panel. You need first to define the initial condition Φ^0 of the iterative algorithm. This is performed in the Solution Initialization task page. Fluent proposes several Initialization Methods for defining suitable initial conditions, notably from the values defined at the boundary conditions. For this lab exercise, you will start the algorithm from rest ($\boldsymbol{v} = 0$, $p = 0$). To do this, choose Standard Initialization in the task page,

[18] For unsteady problems, the initial condition corresponds to the solution at initial time $t = 0$.

[19] Except the Energy equation for heat transfer applications where it is set as $\epsilon = 10^{-6}$.

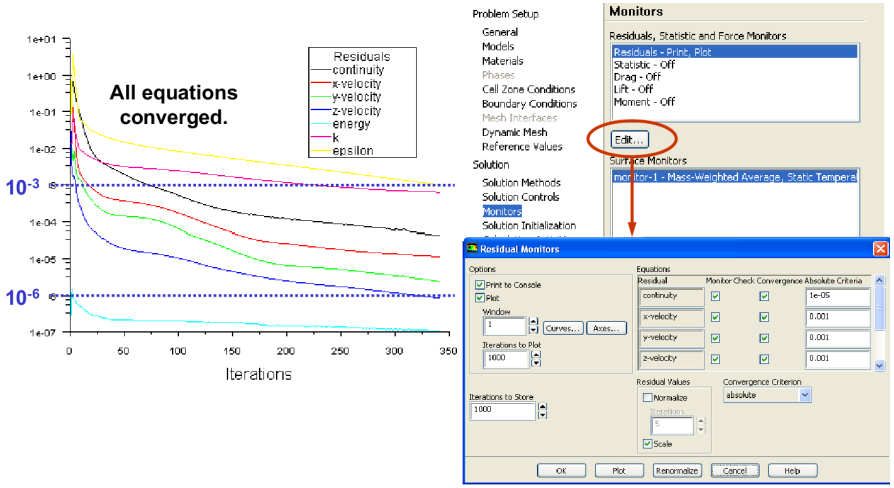


Figure 14: Example of residual monitoring plot for a non-isothermal turbulent flow computation.

Compute From: all-zones, and press Initialize. Optional : the initial solution is ready to be visualized by the postporcessing tools of Section 5.6.

To start the iterative algorithm, go to the Run Calculation task panel, enter a non-zero value for the (maximal) Number of Iterations $k_{\max}$ and press Calculate. As the computation is performed, Fluent plots in a graphic window the residuals at each iteration[20]. The computation will stop when the convergence criteria (3) is reached or $k > k_{\max}$, and the text output in the TUI will be something like :

```
 iter  continuity  x-velocity  y-velocity     time/iter
   470  8.2896e-06  2.4845e-06  9.6828e-08  0:00:07  530
!  470 solution is converged
```

If the last line "`solution is converged`" was missing, then convergence is not reached because the value of $k_{\max}$ is too low. In that case, input a larger value for $k_{\max}$ and press Calculate to resume the computation. If you want to restart the algorithm from the start ($k = 0$), redo the Solution Initialization before clicking on Calculate.

Question 6. *Perform a computation of the flow with the convergence threshold $\epsilon = 10^{-5}$ for all flow equations. Verify that the solution is converged. Display the residual plot and the text output in your* Word *report*[a].

[a]For exporting a Fluent plot, right-click at the top of the graphics window and select Copy to Clipboard.

For advanced users: to perform periodically a back-up copy of the solution during the iterative algorithm, go to the Calculation Activities panel and enter the frequency back-up in the Autosave Every (Iterations) group box.

[20] The frequency of the writing and plotting can be changed with the Reporting Interval option.

5.6 Postprocessing: Results panel

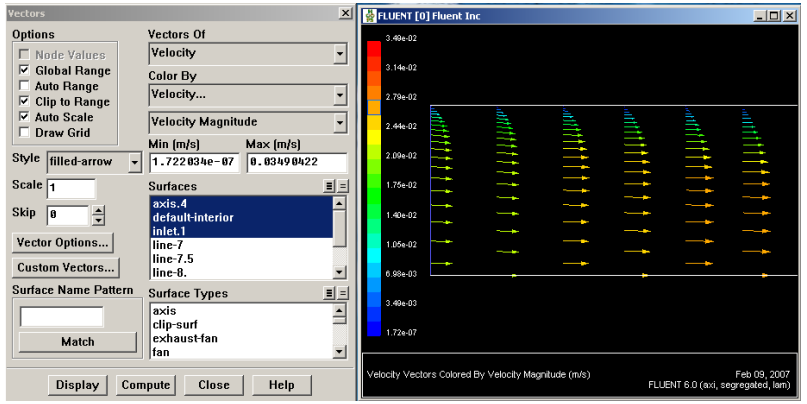


Figure 15: At left: Vectors dialog box. At right: 2D vizualisation of the velocity vectors upstream of the contraction.

Once the converged solution is obtained, Fluent proposes a variety of postprocessing tools to analyze the solution in order to extract relevant flow information. This is performed under the Results panel. There are 3 main types of analysis:

- 2D (or 3D) solution visualization (Graphics and Animation task panel): for plotting the velocity vectors, contour levels of a flow variable (velocity, pressure) inside the computational domain.
- 1D visualization (Plots task panel): for profile plotting of the solution on selected sections (or lines) of the computional domain.
- Numerical reporting of mean flow quantities (Reports task panel): such as mass flow through selected sections, forces at boundaries, ...

Velocity vector visualization: Graphics and Animation task panel

Choose Vectors in the Graphics group box and click on Set Up... . The Vectors dialog box of Fig. 15 (left) opens. From the Surfaces drop-down list choose interior to display the velocity vectors in the whole computational domain[21], and press Display. You can use the mouse to zoom in/out in the graphics window.

Question 7. *Zoom in the computational domain to observe the recirculation zone of the flow (see Fig. 1). Use the* Scale *options of the* Vectors *dialog box if the vectors are too small. Display this plot in your* Word *report.*

[21] Alternatively, you can visualize the flow on selected surfaces and boundaries from the Surfaces drop-down list.

Isocontour visualization: Graphics and Animation task panel

Choose Contours in the Graphics group box and click on Set Up... to open the Contours dialog box. Select a flow variable in the Contours of drop-down list and press Display.

Most flow variables are defined in Fluent with intuitive names. For example, for asymmetric flow:

u = Axial Velocity, v = Radial Velocity, $||\boldsymbol{v}||$ = Velocity Magnitude.

The physical pressure (used in thermodynamics) is the Absolute Pressure $p_{\text{abso}}(\boldsymbol{x}, t)$, which is

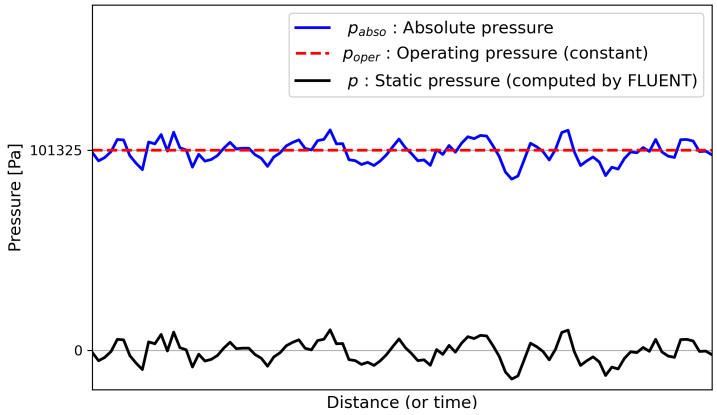


Figure 16: Illustration of the pressure decomposition (4) used in Fluent .

decomposed in Fluent as :

$$p_{\text{abso}}(\boldsymbol{x}, t) = p(\boldsymbol{x}, t) + p_{\text{oper}}, \tag{4}$$

where :

- p_{oper} : the Operating Pressure is defined as a constant in the Operating Conditions... panel, and is set by default as the atmospheric pressure $p_{\text{abso}} = 101325\,\text{Pa}$. Modifying p_{oper} is only significant for other regimes than incompressible flows.
- $p(\boldsymbol{x}, t)$: the Static Pressure (or gauge pressure) is the field actually computed by Fluent .

This decomposition makes sense in an incompressible flow where the static pressure shows small deviations from the atmospheric pressure (see Fig. 16). Other useful quantities related to pressure are :

- $p_{\text{dyn}} = \frac{1}{2}\rho\,||\boldsymbol{v}||^2$: the Dynamic Pressure,
- $p_{\text{total}} = p + \frac{1}{2}\rho\,||\boldsymbol{v}||^2$: the Total Pressure, useful for calculating head losses.

Note that in incompressible flows :

- Density ρ is constant so the pressure is not used for its calculation[22],

[22]In compressible flows : $\rho = p_{\text{abso}}/rT$; in incompressible ideal gas flow : $\rho = p_{\text{oper}}/rT$.

- Only pressure differences are relevant for calculating head losses and hydrodynamic forces, which can be performed in Fluent with the Static Pressure $p(\boldsymbol{x}, t)$.

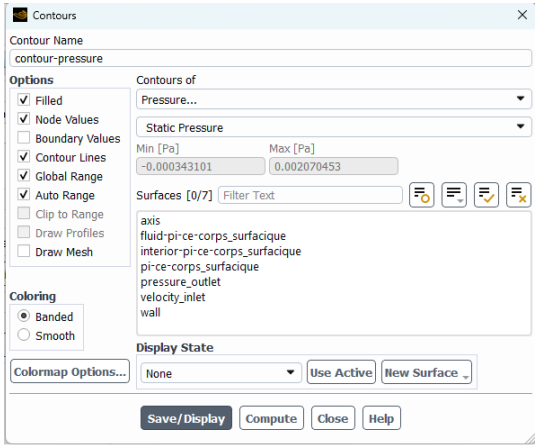


Figure 17: Recommended settings for pressure contour visualisation (choose less than 50 contour levels). It is possible to modify the level range by deselecting the Auto Range option and choosing manually the Min and Max values.

Question 8. *Display the isobaric (pressure) lines of the flow near the contraction. The recommended settings for pressure visualisation are shown in Fig 17 (You can modfiy the number of contours in the Colormap Options ...). Export the figure to your* Word *report. Compare the isobaric contours downstream of the singularity with the theoretical pressure of fully developped Hagen-Poiseuille flow.*

Velocity profiles on sections of the computational domain

This part concerns the plotting of velocity vectors on selected surfaces (lines in 2D) of the computational domain, for example on the $x = 7\,\text{m}$ section[23] near the outlet boundary, in order to verify that the flow is full developed there. To do this, select Surface → Line/Rake... from the menu bar to create the line $x_0 = 7.0$, $y_0 = 0$, $x_1 = 7.0$, $y_1 = 0.2$. Give the line a meaningful label, such as line-7.0 in New Surface Name. The surface line-7.0 will now be available in the Surfaces drop-down list of any Graphics and Vizualization dialog box.

Question 9. *Plot the velocity profiles on Sections* $x = 6.0, 6.5, 7., 7.5$ *and* pressure_outlet.* *(Give them a meaningful name!). Check that the Poiseuille velocity profile is obtained on these sections. Export the graphics to your* Word *report.*

[23]Remember that the origin $x = 0$ correspond to the section of the abrupt expansion.

Plot 1D profiles

The plotting of 1D profiles of predefined flow variables (axial or radial velocity, static or total pressure, ...) is performed on the Plots task panel. Choose XY Plot in the Plots group box and click on Set Up... to open the Solution XY Plot dialog box. For plotting along the radial direction y, choose the direction vector X = 0, Y = 1, Z = 0 in the Plot Direction group box.

Question 10. *Plot the velocity and (static) pressure along the sections of Question 9. Export the graphics to your* Word *report. Do these profiles correspond to a fully developed Poiseuille flow ?*

Define custom field functions

It is possible to define new flow variables (named "custom field functions") from Fluent predefined variables, by using the "pocket calculator" provided by Define → Custom Field Functions... from the menu bar. This new variable can be visualized as previously.

Question 11. *Define the new function* tau_xy *that calculates the shear stress in cylindrical coordinates :*

$$\tau_{xy} = \mu\Big(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\Big).$$

Plot the shear stress profile on the vertical sections of Question 9 and along the upper wall. Export the graphics to your Word *report.*

Calculate mean values

For example, you can calculate the mean flow velocity in Section $x = 8\,\mathrm{m}$ of the pipe. In the Reports task panel, open the Surface Integrals dialog box and choose Area Weighted Average in the Report Type drop-down list.

Question 12. *Verify that the mean flow velocity $\overline{V}$ is constant on the sections of Question 9. Report these values in your* Word *report.*

The Flux Reports dialog box allow you to calculate fluxes at selected surfaces of the domain.

Question 13. *Use the Mass Flow Rate report to verify the mass flow conservation between the inlet and outlet boundaries.*

5.7 Application : calculation of standard pressure drop

In a duct of diameter D, the pressure (or head) loss due to viscous friction between sections x_1 and x_2 reads :

$$H_1 - H_2 = \lambda \frac{\overline{V}^2}{2g} \frac{x_2 - x_1}{D}, \tag{5}$$

where λ is the (dimensionless) friction factor, $\overline{V}$ is the mean flow velocity, g is the gravity, and $H_i = \overline{p}_{\text{total}}(x_i)/\rho g$ is the average head at Section x_i. This formula is valid when the streamlines

are parallel to each others, *i.e.* when the flow is fully developed in the pipe.

Question 14. *According to the previous questions, the flow is fully developed between the zones $6.5 \lesssim x \leq 8$ of the pipe. Use* Fluent *to report the total pressure $\overline{p}_{tot}$ in various sections of the fully developed region, then use (5) to calculate the corresponding value of λ. Compare your results with the empirical relations obtained in a straight pipe:*

- *Poiseuille formula for laminar flows, $Re_D \lesssim 2000$:*

$$\lambda = \frac{64}{Re_D},$$

- *Blasius formula for turbulent flows, $4 \times 10^3 \lesssim Re_D \lesssim 10^5$:*

$$\lambda = 0.3164 Re_D^{-1/4}.$$

6 Numerical computations in the turbulent regime

Before starting the turbulent computations, it is recommended to close FLUENT and make a copy of your laminar flow project in WorkBench in case of later use.

6.1 Performing the computation

We now consider the flow of water with inlet velocity $u_{\text{inlet}} = 1.5 \times 10^{-1}\,\mathrm{m/s}$.

Question 15. *Calculate the Reynolds number. Deduce the flow regime and the flow equations to solve.*

If a turbulence model is needed, use k-epsilon Standard. In addition to the averaged momentum and continuity equations, transport equations for k and ϵ are now solved. You need to define the settings for these additional equations, in particular:

- For inlet BCs, choose Intensity and Hydraulic Diameter in the drop-down list Turbulence Specification Method, and set a turbulent intensity of 5% and an hydraulic diameter of 0.2 m.
- Use 2nd-order methods for the convective terms of these equations.
- Define the tolerance monitoring parameters for these equations.

Question 16. *Starting from rest, compute the solution with residual threshold $\epsilon = 10^{-6}$ for all equations. Export the residual history to your* Word *report.*

When performing a high-Reynolds number with impulsive start from the rest, the iterative solution may undergo convergence problems: the residuals do not decrease monotonously, stall or increase until solution blow-up. If this happen, you need to modify the default solver settings for obtaining convergence. The "optimal" parameters for convergence are usually problems dependent, and fine-tuning of these parameters requires expertise and trial-and-errors. General guidelines for convergence troubleshooting are given in the lecture notes. Here is a quick summary:

– Initialize the solution with a previously calculated solution at lower Reynolds number, then gradually increases the inlet velocity.
– Use first-order methods in the initial stages of the iterative algorithm, then use more accurate methods.
– Decreases the values of the under-relaxation factors of the equation(s) that cause(s) difficulties (usually, it is the continuity equation).
– Improve the mesh quality and use finer cells in zones of high flow gradients.

However, for this lab exercise you shouldn't experience convergence problems: After an initial stage where the residuals seem to be stalled, the solution should converge after a few thousand iterations. Note however that this behaviour may depends on the discretization parameters you chose or the version of Fluent you are using.

6.2 Analyze the results

General flow features

Question 17. *Verify that the flow is fully developed at the outlet. Display in the 2D domain the velocity profiles near inlet (several sections in the upstream pipe) and outlet (sections of Question 9). Compare with the laminar velocity profiles obtained previously. Report these results in your* Word *report.*

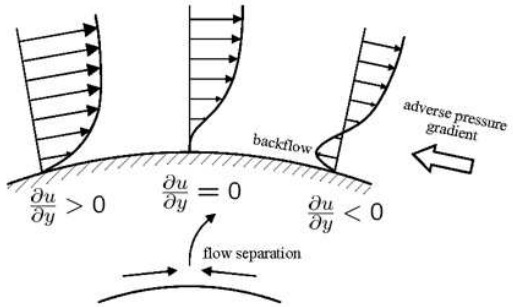


Figure 18: Velocity profiles at a wall boundary layer near a separation point.

Calculation of singular head losses

At the vicinity of a geometrical singularity such as the sudden expansion at $x_1 = 0\,m$ (*cf.* Fig. 1), the flow is no more unidirectional and the head loss equation (5) is not valid anymore. Experiments show that singular head losses are proportional to the mean kinetic energy:

$$\Delta H = \frac{\xi}{2g}\,\overline{V}^2,$$

where ξ is a dimensionless coefficient that depends on the type of singular geometry. For singular expansion flows, the value of the head-loss coefficient ξ can be estimated analytically at high flow rates with the following assumptions:

- Losses due to wall friction are negligible,
- The streamlines are parallel through Section S_1 ($x_1 = 1.0\,m$) and Section S_2 (*cf.* Fig. 1), where the flow returns to fully developed regime. This implies that pressure distribution is uniform at these sections.

With these hypotheses, conservation of mass and momentum between S_1 and S_2 yields:

$$H_1 - H_2 = \frac{\xi}{2g}\,\overline{V}^2(x_1), \qquad \text{with} \quad \xi = \left(1 - \frac{S_1}{S_2}\right)^2. \tag{6}$$

This formula can be found in any fluid dynamics textbook.

Question 18. *From the head losses given by your numerical solution, calculate the head-loss coefficient ξ and compare with the analytical value. The difficulty is to estimate the abscissa x_2 where the streamlines do reattach. To estimate x_2, you may use the property that the shear stress is zero at a separation point (where flow separates or reattaches, see Fig. 18): the value of x_2 is then obtain by plotting the computed shear stress along the upper wall (cf. Question 11).*