BE multidisciplinaire

Vincent Lecrubier et Mathilde Peyrega

11 mai 2009

Table des matières

1 Préambule **4**

2 Problématique du bureau d'étude **5**
2.1 Objectif 5
2.2 Présentation du modèle physique 5
2.2.1 Le modèle des Vortex 5
2.3 Méthodes de calcul analytique pour l'intégration 5
2.3.1 Méthode d'Euler explicite 5
2.3.2 Méthode de Runge-Kutta 2 (aussi dite schéma d'Heun) 6
2.3.3 Méthode de Runge-Kutta 4 (aussi dite schéma de Runge-Kutta classique 6

3 Choix de conception et architecture du programme **7**
3.1 Diagramme de classes 7
3.2 Description des packages 8
3.2.1 package GUI 8
3.2.2 package modèles 9
3.2.3 package initialiseurs 10
3.2.4 package intégrateurs 10
3.2.5 Amélioration possible du code 10

4 Validation du programme **12**
4.1 Cas de deux vortex 12
4.2 Cas des allées tourbillonnaires 12

5 Résultats et comparaison des différentes méthodes de calcul **13**
5.1 Comparaison des différentes méthodes de calcul 13
5.1.1 Cas d'une allée tourbillonaire 13
5.1.2 Cas de deux vortex 14
5.1.3 Cas d'un coeur tourbillonnaire 14

6 Conclusion **16**

1 Préambule

L'ojectif de ce bureau d'étude est de faire la synthèse des compétences acquises en aérodynamique et en programmation orientée objet afin de modéliser un problème aérodynamique et de pouvoir comparer plusieurs méthodes de résolution analytique.

2 Problématique du bureau d'étude

2.1 Objectif

L'objectif de ce bureau d'étude est de concevoir et de développer un logiciel en Java permettant de visualiser le déplacement des tourbillons, afin de modéliser une couche de mélange par des allées tourbillonaires.

2.2 Présentation du modèle physique

2.2.1 Le modèle des Vortex

Nous utilisons ici la méthode des Vortex qui permettent de modéliser des singularités tourbillonnaires ponctuelles caractérisées par leur position, leur vitesse, et leur intensité. Et un ensemble d'allées tourbillonaires permet de représenter une couche de mélange.

Si l'on considère un ensemble de N allées tourbillonnaires distantes d'une longueur supérieure au rayon de coupure $\sigma = \frac{\delta}{2\pi}$ la vitesse en un point du champ est définie par :

$$U(x,y) = -\frac{1}{2a}\sum_{k=1}^{N}\frac{\Gamma_k sinh\frac{2\pi(y-y_k)}{a}}{cosh\frac{2\pi(y-y_k)}{a} - cos\frac{2\pi(x-x_k)}{a}}$$

$$V(x,y) = \frac{1}{2a}\sum_{k=1}^{N}\frac{\Gamma_k sinh\frac{2\pi(x-x_k)}{a}}{cosh\frac{2\pi(y-y_k)}{a} - cos\frac{2\pi(x-x_k)}{a}}$$

2.3 Méthodes de calcul analytique pour l'intégration

Nous devons écrire une application qui calcule la position des singularités vortex à une certaine itération indépendante de l'initialisation des Vortex. Il existe plusieurs méthodes d'intégration de la position. Nous avons choisi d'en étudier 3 : les méthodes d'Euler explicite, de Runge-Kutta d'ordre 2 et de Runge-Kutta d'ordre 4. Nous allons présenter ci-dessus le principe de ces différentes méthodes pour résoudre une équation de la forme

$$\frac{dU}{dt} = F(U,t)$$

.

2.3.1 Méthode d'Euler explicite

On suppose que l'on a :

$$U^{n+1} \approx U^n + \Delta t F(U^n, t_n)$$

Ainsi on peut calculer par itération successive pour tout n les valeurs de U^n. Cependant, cela revient à faire une approximation à l'ordre 1. L'erreur commise à chaque étape est de l'ordre h^2 et l'erreur totale accumulée est de l'ordre de h.

2.3.2 Méthode de Runge-Kutta 2 (aussi dite schéma d'Heun)

La méthode de Runge-Kutta 2 est donnée par l'équation :

$$y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2)$$

où

$$k_1 = F(t_n, y_n)$$
$$k_2 = F(t_n + h, y_n + hk_1)$$

Runge-Kutta 2 est une méthode d'ordre 2, ce qui signifie que l'erreur commise à chaque étape est de l'ordre h^3 et l'erreur totale accumulée est de l'ordre de h^2.

2.3.3 Méthode de Runge-Kutta 4 (aussi dite schéma de Runge-Kutta classique

La méthode Runge-Kutta 4 est donnée par l'équation :

$$y_{n+1} = y_n + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

où

$$k_1 = F(t_n, y_n)$$
$$k_2 = F(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_1)$$
$$k_3 = F(t_n + \frac{h}{2}, y_n + \frac{h}{2}k_2)$$
$$k_4 = F(t_n + h, y_n + hk_3)$$

L'idée de cette méthode est que la valeur y_{n+1} est approchée par la somme de la valeur y_n et du produit de la taille de l'intervalle h par la pente estimée $\frac{k_1+2k_2+2k_3+k_4}{6}$. Runge-Kutta 4 est une méthode d'ordre 4, ce qui signifie que l'erreur commise à chaque étape est de l'ordre h^5 et l'erreur totale accumulée est de l'ordre de h^4.

3 Choix de conception et architecture du programme

3.1 Diagramme de classes

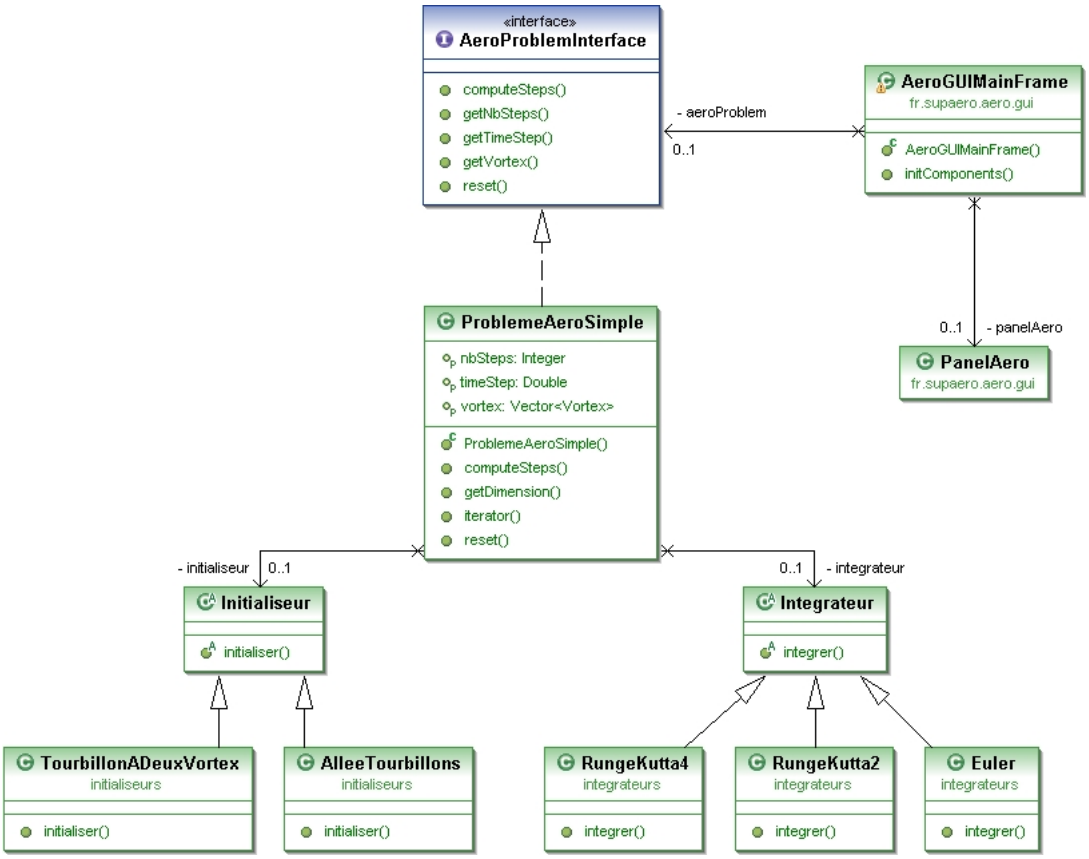


FIGURE 1 – Diagramme de classe du problème aéro simple

Le programme est basé autour de l'interface AeroProblemInterface, qui est implémentée dans plusieurs classes, en particulier ProblemeAeroSimple, qui représente un problème classique, à plusieurs Vortex, et ProblemAeroPeriodique, qui représente un problème constitué de plusieurs allées tourbillonnaires infinies.

Les deux actions principales pouvant être appliquée à un problème aero, sont l'initialisation, et l'intégration. L'initialisation consiste à définir les différents éléments du problème à un instant initial. L'integration consiste à calculer l'état du système à la date t+dt, à partir du problème a la date t.

Afin de réaliser ces deux actions, plusieurs solutions sont envisageables, parmi lesquelles :

- La création d'une méthode « initialiser » et « integrer » directement dans la classe ProblemAero, que l'on appellera approche « directe ». L'approche directe lie dans une meme classe la définition du probleme, et sa méthode de résolution, qui se retrouvent liées de manière rigide.
- L'utilisation du design pattern strategie, ou les actions « initialiser » et « integrer » ne seront pas représentées par des méthodes dans la classse ProblemeAero, mais

par des interfaces implémentées par des classes qui agiront sur la classe ProblemeAero. Ces classes seront donc des intervenants exterieurs, qu'il suffira d'appeller afin d'executer telle ou telle action sur le problème.

L'utilisation du design pattern strategie, ou les actions « initialiser » et « integrer » ne seront pas représentées par des méthodes dans la classse ProblemeAero, mais par des interfaces implémentées par des classes qui agiront sur la classe ProblemeAero. Ces classes seront donc des intervenants exterieurs, qu'il suffira d'appeller afin d'executer telle ou telle action sur le problème.

Le design pattern strategie a donc été retenu car il permet une souplesse beaucoup plus forte, et permet de modifier rapidement l'initialisation du problème, ou sa méthode d'integration, qui sont découplés. Afin de gerer l'interface homme machine, le design pattern MVC a été utilisé. Ce design pattern permet de gerer l'affichage des données indépendamment de leur calcul. La classe ProblemAeroSimple joue le rôle de modèle, pendant que la vue et le controleur sont mis en place par les classes fournies.

3.2 Description des packages

3.2.1 package GUI

Le package GUI contient les classes correspondant à l'IHM :

- AeroGUIMainFrame, qui représente la fenetre principale du programme, en SWING. La fenetre principale contient les controles permettant de modifier les paramètres de l'intégration du problème Aero, ainsi qu'une représentation graphique du problème en cours.
- la classe PanelAero, qui permet l'affichage graphique d'un ensemble de vortex correspondant à un problemeAero.

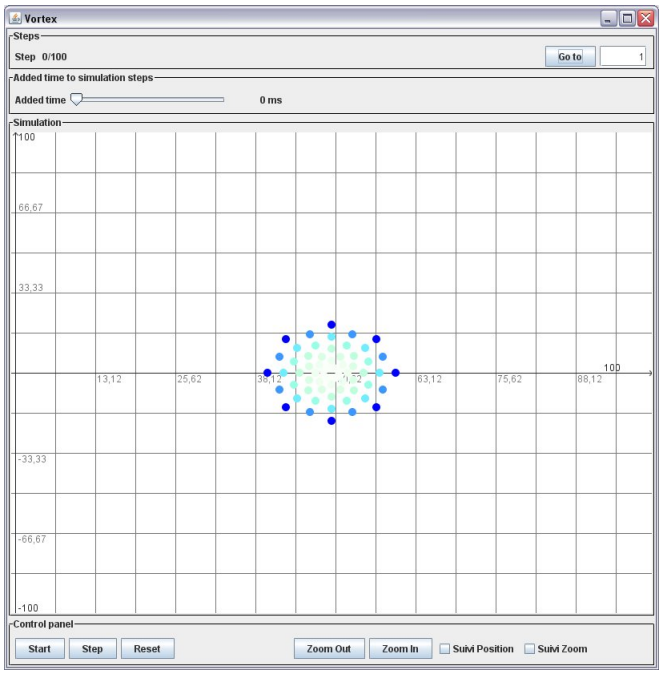


FIGURE 2 –

Nous avons légèrement modifié ces deux classes afin

- De permettre l'utilisation de la souris pour se déplacer dans la vue et effectuer le zoom.
- D'ajouter en fonction de suivi, qui permet à la vue de suivre automatiquement les vortex affichés.
- De colorier les vortex en fonction de leur intensité.

3.2.2 package modèles

Le package modèle contient les différentes interfaces nécéssaires à la mise en place des choix de conception. Nous retrouvons donc les interfaces « initialiseur » et « integrateur » du design pattern strategie.

Ce package contient aussi les classes permettant de décrire les éléments de base de notre problème : les vortex. Une classe abstraite « Vortex » est réalisée sous deux formes : « Vortex Simple » qui représente un vortex classique, et « Vortex Multiples » qui représente une allée tourbillonnaire infinie.

Enfin, ce package contient les classes permettant de définir les problèmes. La dualité vortex simple / vortex multiples nous impose de définir deux types de prblèmes, les problèmes simples, composé de simples vortex isolés, et les problèmes périodiques, qui contiennent des allées tourbillonnaires.

Remarque : l'architecture du programme permet de définir et integrer des problèmes mixtes, composés de vortex isolés ET d'allées tourbillonnaires. Cependant de tels problèmes ne seront pas résolus correctement, et donneront des résultats fantaisistes.

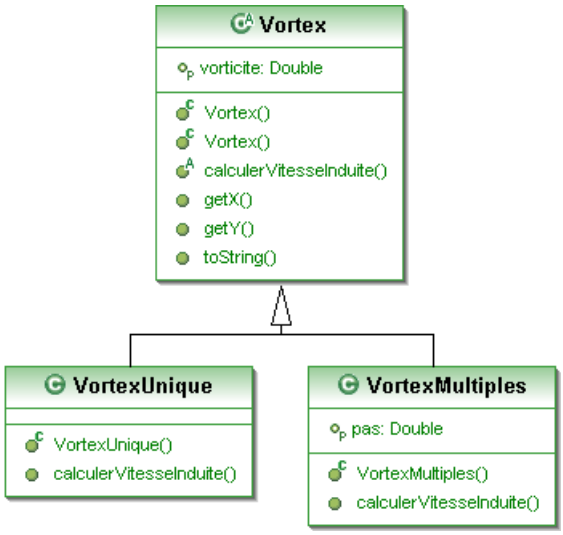


FIGURE 3 –

3.2.3 package initialiseurs

Le package initialiseur contient les différentes classes implémentant l'interface « initialiseur », et permettant donc pour chacune d'initialiser un problème Aero d'une maniere différente.

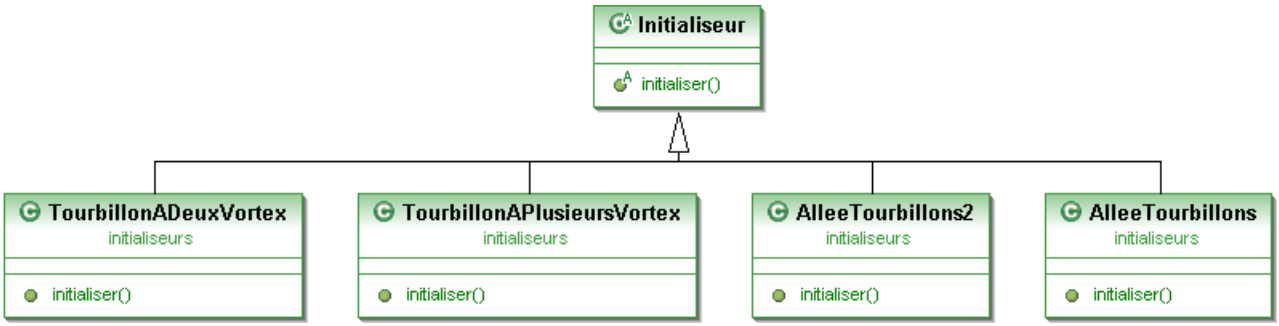


FIGURE 4 –

3.2.4 package intégrateurs

Le package integrateur contient les différentes classes implémentant l'interface « intégrateur », et contient donc les classes correspondant aux différents schémas définis en 2,3.

3.2.5 Amélioration possible du code

De nombreuses pistes sont exploitables pour accelerer l'execution du calcul, car notre conception ne s'est pas attardée sur l'aspect performances du calcul.

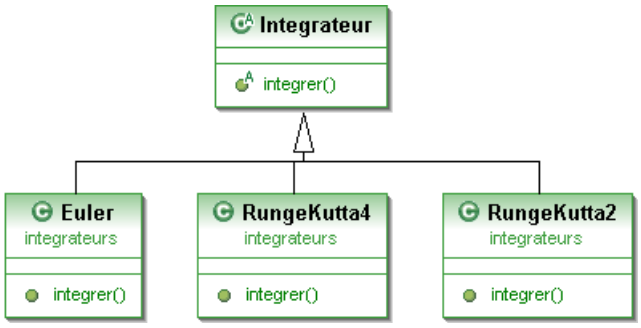


FIGURE 5 –

- La premiere piste est l'optimisation du code afin d'éviter au maximum les appels de méthodes trop longues, en les codant directement là ou elles sont utiles.
- Une technique intéressante est l'organisation des vortex non plus sous forme d'une liste, mais sous forme d'un arbre, ou chaque feuille est un vortex du problème initial, et ou l'on regroupe les vortex par proximité géographique, en créant des vortex « virtuels » équivalent à un groupe de vortex. Ainsi, pour calculer la vitesse induite par ce groupe de vortex, il serait possible, à une distance suffisante, d'utiliser le vortex équivalent, ce qui permettrai d'économiser du temps de calcul, d'autant plus que le nombre de vortex du problème est élevé, tout en perdant un petit peu de précision.

4 Validation du programme

4.1 Cas de deux vortex

Afin de valider les schémas d'intégration, nous avons dans un premier temps créé un problème dont la solution analytique est connue. Ce problème consiste en deux vortex d'intensité égale, séparés d'une distance d=2*r La solution analytique nous dit que ces deux vortex tourneront sur un cercle de rayon r, centré sur le barycentre du système, et auront une vitesse égale à w/(2*pi*d), et donc une vitesse angulaire égale à $frac\omega 2 * \pi * d)^2$.

La simulation de ce système permet de valider les codes de calcul. Le résultat trouvé est cohérent pour les trois méthodes que nous avons mises en place, mais les aspects numériques créent une légère instabilité du système. Ainsi, la distance entre les deux tourbillons, qui devrait rester constante égale à d, augmente légèrement au fil du temps, donnant des courbes semblables à l'illustration 1. La période de rotation du système est elle aussi affectée, dans la même mesure.

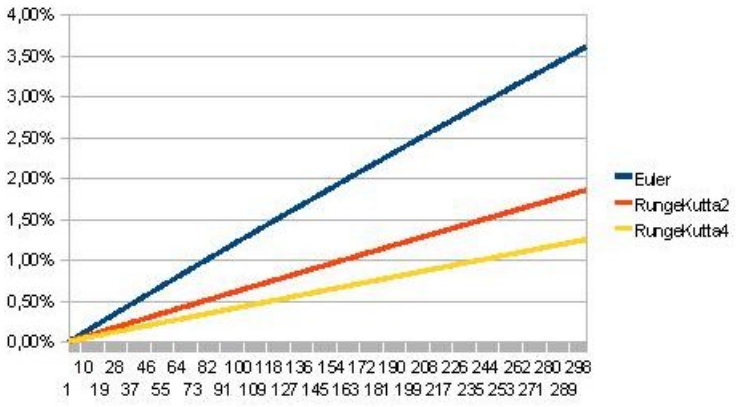


FIGURE 6 –

4.2 Cas des allées tourbillonnaires

Les schémas d'intégration étant validés grâce au cas à deux tourbillons, la validation dans le cas d'allées tourbillonnaires s'est résumé à la vérification du calcul correct des vitesses induites par une allée infinie de tourbillons, seule différence entre les deux cas.

Pour cela nous avons créé différents problèmes dont les solutions analytiques sont stationnaires, et vérifié que la solution numérique ne divergeait pas trop rapidement.

5 Résultats et comparaison des différentes méthodes de calcul

5.1 Comparaison des différentes méthodes de calcul

5.1.1 Cas d'une allée tourbillonaire

Le cas proposé dans l'énoncé du BE est un problème représentant une allée tourbillonnaire, au niveau de la surface de glissement horizontale entre deux domaines fluides présentant une différence de vitesse. Ce type de situation engendre une instabilité de Kelvin-Helmholtz, consistant en une suite périodique de tourbillons se formant à l'interface entre les deux domaines. Afin de simuler ce problème, nous avons donc utilisé la classe ProblemeAeroPeriodique, initialisée avec plusieurs séries infinies de vortex, qui représentent à l'instant initial la zone ou le gradient de vitesse est important, à l'interface entre les domaines. Afin d'accélérer la formation de l'instabilité de Kelvin Helmholtz, nous initialisons cette couche sous une forme déjà légèrement perturbée, comprenant des oscillations de période donnée, qui donneront naissance aux tourbillons. Physiquement, les observations montrent que des tourbillons se forment très rapidement, et l'épaisseur de la couche de mélange augmente progressivement. Afin de mesurer la précision de nos modèles, et pour pouvoir les comparer, nous avons introduit un scalaire significatif de l'épaisseur de la couche de mélange. Ce scalaire n'est autre que l'écart type de des positions verticales des différents vortex élémentaires du problème. Initialement alignés horizontalement ou presque, ces vortex se mettent a tourner en groupe, et forment rapidement des tourbillons, et le scalaire choisi nous renseigne sur le diamètre de ces tourbillons. Les différentes méthodes d'intégration donnent des résultats très similaires sur le début du phénomène, à fortiori si le pas de temps est faible. Cependant, les méthodes les moins précises divergent assez rapidement, et on assiste à des artefacts, comme des discontinuités dans l'épaisseur de la couche de mélange. Selon le pas de temps choisi, ces problèmes arrivent plus ou moins rapidement.

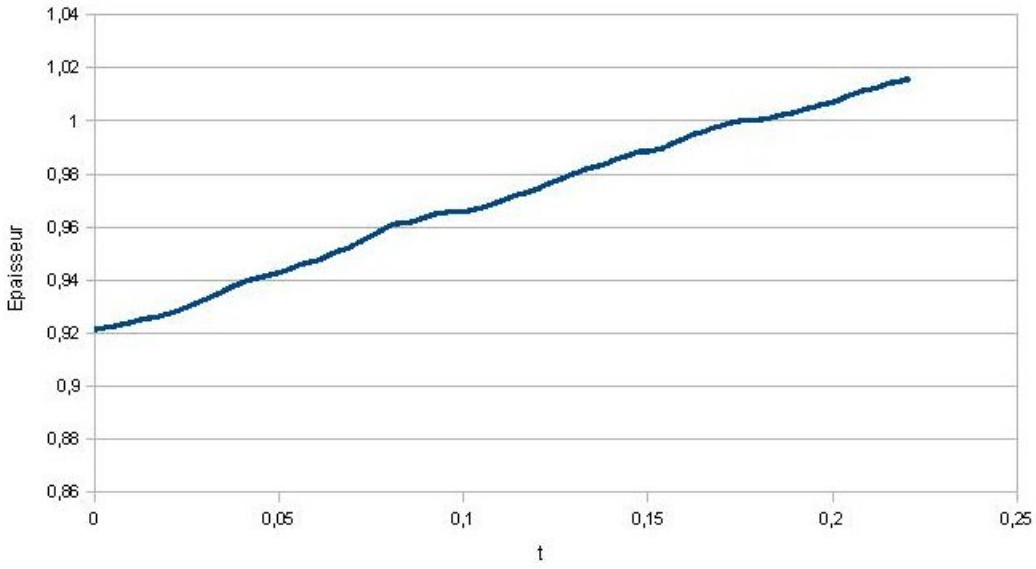


FIGURE 7 – Illustration 2 : Epaisseur en RK4, dt = 5e-5

5.1.2 Cas de deux vortex

Ce cas a servi pour valider les schémas d'intégration, le scalaire significatif choisi est la distance entre les deux tourbillons, qui doit rester constante, si le code de calcul est parfait. Nous voyons ici que pour des pas de temps raisonnablement faible, l'erreur relative générée par la simulation au bout de 1000 itérations est directement reliée au pas de temps, et que les méthodes d'ordre plus élevé sont naturellement les plus précises. Ce raisonnement a ses limites et ne s'applique plus si les pas de temps sont beaucoup trop grands, l'erreur relative est alors supérieure à 1, le modèle n'est donc plus fiable du tout, et les méthodes réagissent toutes différemment.

5.1.3 Cas d'un coeur tourbillonnaire

Nous avons ici simulé un tourbillon formé de plusieurs vortex répartis dans un disque. Les intensités des tourbillons sont choisies afin que la circulation de l'écoulement augmente linéairement lorsqu'on s'écarte du centre du tourbillon, le rotationnel est donc constant dans le coeur du tourbillon. Les scalaires choisis ici sont l'écart type des position horizontales et verticales des vortex. La solution analytique d'un tel problème nous montre que ces scalaires devraient rester constant dans le temps, le tourbillon tournant sur lui même sans grossir. Les simulations donnent un résultat intéressant : au départ, le tourbillon reste assez stable, tout en grossisant légèrement de manière isotrope. Au bout d'un certain temps dépendant du pas de temps, de la méthode et des conditions initiales, la so-

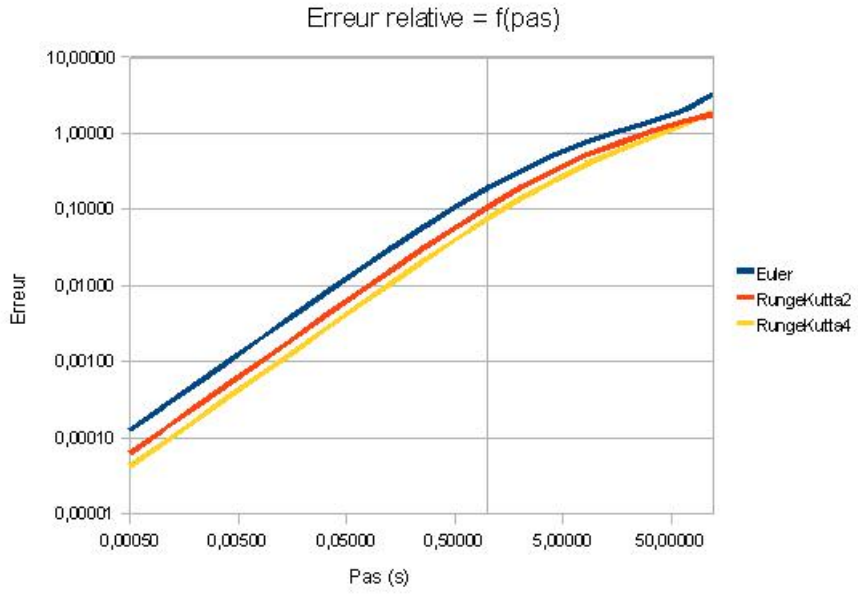
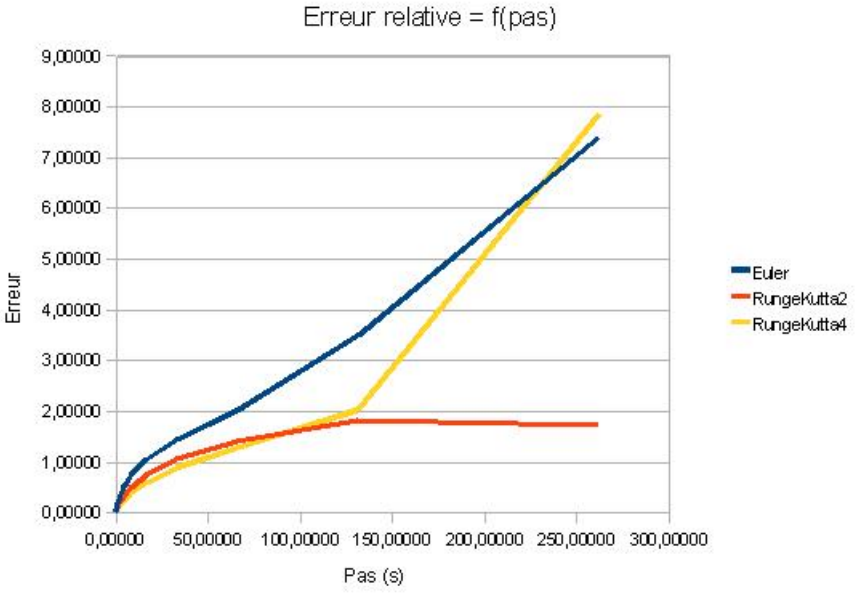
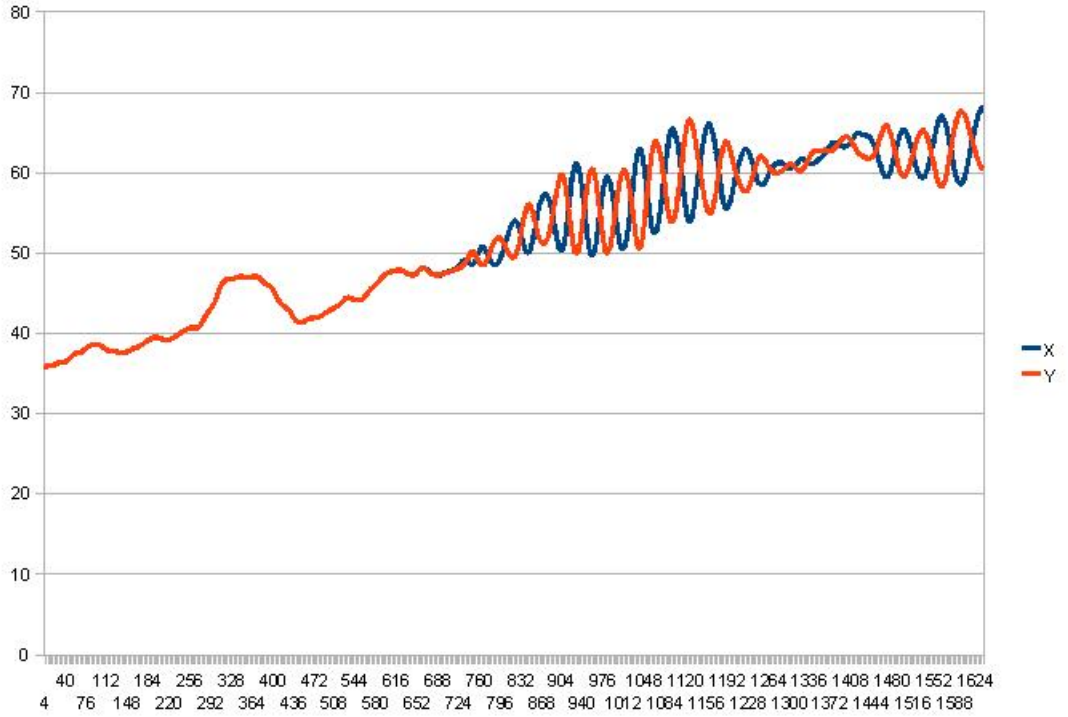


FIGURE 8 – Illustration 2 : Epaisseur en RK4, dt = 5e-5



lution numérique s'écarte vraiment de la solution analytique, car le tourbillon commence à prendre une forme elliptique, tout en continuant sa rotation, ce qui donne les oscillations déphasées sur la courbe suivante.



6 Conclusion

Lors de ce bureau d'étude nous avons réussi à visualiser une modélisation par vortex de couche aérodynamique. La conjugaison des différentes compétences en aérodynamique et en informatique nous a permis de nous intéresser à différente problématique comme la complexité informatique et l'intérêt d'une conception logitiel claire qui permet à l'aérodynamicien une visualisation rapide mais aussi plusieurs initialisations des simulations. D'ailleurs, l'amélioration de l'interface est une des améliorations possibles du logitiel.