

## Projet 12

---

# Mécanique des fluides : résolution des équations de Navier-Stokes 2D

### Fiche Projet

**Difficulté :** 3

**Notions développées :** Équations de Navier-Stokes, équation de Helmholtz, méthode de projection, factorisation ADI, transformée de Fourier rapide FFT

**Domaines d'application :** Écoulements fluides 2D, instabilité de Kelvin-Helmholtz, dipôle de vorticité

Nous nous intéressons dans ce projet à la résolution des équations qui régissent la dynamique des fluides incompressibles (*i.e.* à masse volumique constante), les équations de Navier-Stokes.

Pour rendre ce problème (généralement assez difficile à résoudre) plus abordable, nous nous plaçons dans le cadre simplifié des équations bidimensionnelles (2D),

avec des conditions aux limites périodiques. Ce modèle nous permettra néanmoins de simuler quelques écoulements intéressants :

- l’instabilité de Kelvin-Helmholtz d’une couche de mélange plane,
- l’évolution d’un structure tourbillonnaire particulière, le dipôle de vorticité.

Sur le plan numérique, plusieurs algorithmes (d’intérêt beaucoup plus général) seront abordés :

- la discrétisation des équations par différences finies (schémas compacts à l’ordre 6) ;
- les schémas d’intégration en temps d’Adams-Bashforth et Crank-Nicolson ;
- la résolution d’une équation de Helmholtz par une méthode de factorisation approximative (ou ADI) ;
- la résolution d’un laplacien en utilisant une transformée de Fourier rapide (FFT) ;
- la résolution d’un système tridiagonal périodique par la méthode de Thomas.

## 12.1 ÉQUATIONS DE NAVIER-STOKES 2D, INCOMPRESSIBLES

L’écoulement 2D (dans le plan  $(x, y)$ ) d’un fluide de masse volumique  $\rho$  constante est complètement décrit par son champ vectoriel de vitesse  $q = (u(x, y), v(x, y)) \in \mathbb{R}^2$  et son champ scalaire de pression  $p(x, y) \in \mathbb{R}$ . Ces grandeurs seront déterminées à partir des lois de conservation suivantes (voir, par exemple, [Hirsh, 1988]) :

- la conservation de la masse :

$$\operatorname{div}(q) = 0, \quad (12.1)$$

ou, en explicitant l’opérateur divergence<sup>1</sup> :

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0; \quad (12.2)$$

- la conservation de la quantité de mouvement, écrite sous forme compacte :

$$\frac{\partial q}{\partial t} + \operatorname{div}(q \otimes q) = -\mathcal{G}p + \frac{1}{Re}\Delta q, \quad (12.3)$$

---

1. Rappelons les définitions sur  $\mathbb{R}^2$  des opérateurs divergence, gradient et laplacien :

$$\text{si } v = (v_x, v_y) : \mathbb{R}^2 \mapsto \mathbb{R}^2 \text{ et } \varphi : \mathbb{R}^2 \mapsto \mathbb{R}$$

$$\operatorname{div}(v) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}, \quad \mathcal{G}\varphi = (\frac{\partial \varphi}{\partial x}, \frac{\partial \varphi}{\partial y}), \quad \Delta\varphi = \operatorname{div}(\mathcal{G}\varphi) = \frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2}, \quad \Delta v = (\Delta v_x, \Delta v_y).$$

ou, explicitée :

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial v}{\partial t} + \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right). \end{cases} \quad (12.4)$$

Dans les équations précédentes, les variables s'entendent sans dimension, c'est-à-dire

$$x = \frac{x^*}{L}, \quad y = \frac{y^*}{L}, \quad u = \frac{u^*}{V_0}, \quad v = \frac{v^*}{V_0}, \quad t = \frac{t^*}{L/V_0}, \quad p = \frac{p^*}{\rho V_0^2}, \quad (12.5)$$

où les variables (\*) sont les grandeurs physiques et  $L, V_0$  sont, respectivement, les échelles de longueur et de vitesse caractéristiques de l'écoulement. Le paramètre de similitude de l'écoulement est le nombre de Reynolds, défini comme le rapport entre les effets d'inertie et de viscosité dans le fluide :

$$Re = \frac{V_0 L}{\nu}, \quad \text{avec } \nu \text{ la viscosité cinématique du fluide.} \quad (12.6)$$

En conclusion, le système d'équations de Navier-Stokes à résoudre est constitué par les équations aux dérivées partielles (EDP) (12.2), (12.4) ; la condition initiale (à  $t = 0$ ) et les conditions aux limites seront précisées plus tard.

## 12.2 MÉTHODE DE RÉSOLUTION

Le système d'équations de Navier-Stokes sera résolu par une *méthode de projection* [Orlandi, 1999] comportant deux pas.

1. *Un pas de prédiction* : on résout les équations de quantité de mouvement (12.4) écrites sous la forme :

$$\frac{\partial q}{\partial t} = -\mathcal{G}p + \mathcal{H} + \frac{1}{Re} \Delta q, \quad \text{pour } q = (u, v) \in \mathbb{R}^2, \quad (12.7)$$

avec le vecteur contenant les termes convectifs

$$-\mathcal{H} = \left( \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y}, \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \right), \quad (12.8)$$

et le vecteur gradient de pression :

$$\mathcal{G}p = \left( \frac{\partial p}{\partial x}, \frac{\partial p}{\partial y} \right). \quad (12.9)$$

Les équations (12.7) seront discrétisées suivant un schéma explicite d'Adams-Bashfort (pour les termes convectifs  $\mathcal{H}$ ), combiné avec un schéma semi-implicite de Crank-Nicolson (pour les termes diffusifs  $\Delta q$ ). L'avancement de la solution du temps  $t_n = n\delta t$  au  $t_{n+1} = (n + 1)\delta t$  est donné par le schéma :

$$\frac{q^* - q^n}{\delta t} = -\mathcal{G}p^n + \underbrace{\frac{3}{2}\mathcal{H}^n - \frac{1}{2}\mathcal{H}^{n-1}}_{\text{Adams-Bashfort}} + \underbrace{\frac{1}{Re}\Delta\left(\frac{q^* + q^n}{2}\right)}_{\text{Crank-Nicolson}}. \quad (12.10)$$

Dans la formulation précédente, la pression apparaît en explicite (évaluée au temps  $t_n$ ), ce qui fait que le champ de vitesse  $q^*$  ainsi calculé ne vérifie pas l'équation de continuité (12.1).

2. *Un pas de correction (ou projection) :* on corrige le champ  $q^*$  (dit *non-solenoïdal*) pour obtenir un champ de vitesse  $q^{n+1}$  (de divergence nulle, ou *solenoïdal*). L'équation de correction est la suivante<sup>2</sup> :

$$q^{n+1} - q^* = -\delta t \mathcal{G}\phi. \quad (12.11)$$

La variable  $\phi$  (reliée à la pression, mais sans signification physique) sera déterminée en prenant la divergence de l'équation (12.11); tenant compte que  $\operatorname{div}(q^{n+1}) = 0$  et que  $\operatorname{div}(\mathcal{G}\phi) = \Delta\phi$ , nous obtenons :

$$\Delta\phi = \frac{1}{\delta t} \operatorname{div}(q^*). \quad (12.12)$$

Pour finir, l'algorithme est bouclé en réactualisant, pour le pas de temps suivant, la pression par<sup>3</sup>

$$p^{n+1} = p^n + \phi - \frac{\delta t}{2Re} \Delta\phi. \quad (12.13)$$

- Pour résumer, l'algorithme de résolution comportera les étapes suivantes (en reprenant les équations sous une forme plus pratique pour la programmation) :

---

2. Cette équation exprime le fait que les champs  $q^*$  et  $q^{n+1}$  ont le même rotationnel (effectivement, on peut éliminer la pression des équations de Navier-Stokes en prenant le rotationnel des équations de quantité de mouvement).

3. Cette équation est obtenue en écrivant l'équation (12.10) avec la pression en implicite

$$\frac{q^{n+1} - q^n}{\delta t} = -\mathcal{G}p^{n+1} + \frac{3}{2}\mathcal{H}^n - \frac{1}{2}\mathcal{H}^{n-1} + \frac{1}{Re}\Delta\left(\frac{q^{n+1} + q^n}{2}\right)$$

et en faisant la différence avec (12.10). Après remplacement de  $q^*$  de l'équation de correction (12.11), nous obtenons (12.13), à une constante additive près. Observons que cette constante n'a aucune influence sur le déroulement de l'algorithme, car la pression intervient seulement par son gradient dans les équations de quantité de mouvement.

**Algorithme 12.1.** Résolution des équations de Navier-Stokes (12.2)-(12.4) : à partir du champ  $(u^n, v^n, p^n)$ , il faut calculer

(A) les termes explicites  $\mathcal{H}^n$  :

$$\mathcal{H}_u^n = - \left( \frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} \right), \quad (12.14)$$

$$\mathcal{H}_v^n = - \left( \frac{\partial uv}{\partial x} + \frac{\partial v^2}{\partial y} \right), \quad (12.15)$$

(B) le champ non-solénoïdal  $q^* = (u^*, v^*)$  par la résolution des équations :

$$\left( I - \frac{\delta t}{2Re} \Delta \right) u^* = u^n + \delta t \left[ - \frac{\partial p^n}{\partial x} + \frac{3}{2} \mathcal{H}_u^n - \frac{1}{2} \mathcal{H}_u^{n-1} + \frac{1}{2Re} \Delta u^n \right], \quad (12.16)$$

$$\left( I - \frac{\delta t}{2Re} \Delta \right) v^* = v^n + \delta t \left[ - \frac{\partial p^n}{\partial y} + \frac{3}{2} \mathcal{H}_v^n - \frac{1}{2} \mathcal{H}_v^{n-1} + \frac{1}{2Re} \Delta v^n \right], \quad (12.17)$$

(C) la variable  $\phi$  par la résolution de l'équation de Poisson :

$$\Delta \phi = \frac{1}{\delta t} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right), \quad (12.18)$$

(D) le champ solénoïdal  $q^{n+1} = (u^{n+1}, v^{n+1})$  :

$$u^{n+1} = u^* - \delta t \frac{\partial \phi}{\partial x}, \quad (12.19)$$

$$v^{n+1} = v^* - \delta t \frac{\partial \phi}{\partial y}, \quad (12.20)$$

(E) le nouveau champ de pression :

$$p^{n+1} = p^n + \phi - \frac{\delta t}{2Re} \Delta \phi. \quad (12.21)$$

Les étapes (A)-(E) seront reprises pour chaque pas de temps.

## 12.3 DOMAINE DE CALCUL, CONDITIONS AUX LIMITES ET MAILLAGE

Le domaine de calcul est rectangulaire, de dimensions  $L_x \times L_y$  (voir fig. 12.1). Quant aux conditions aux limites, les conditions de périodicité simplifient considérablement la résolution des équations de Navier-Stokes. Nous considérons donc que les champs de vitesse  $q(x, y)$  et de pression  $p(x, y)$  sont périodiques en  $x$  et  $y$  :

$$q(0, y) = q(L_x, y), \quad p(0, y) = p(L_x, y), \quad \forall y, \quad (12.22)$$

$$q(x, 0) = q(x, L_y), \quad p(x, 0) = p(x, L_y), \quad \forall x. \quad (12.23)$$

Les équations seront discrétisées sur un maillage de type différences finies en utilisant  $n_x$  points de discrétisation suivant la direction  $x$  et  $n_y$  points suivant la direction  $y$ . Le maillage  *primaire* est défini par les points de coordonnées (voir fig. 12.1)

$$x_c(i) = (i - 1)\delta x, \quad \delta x = \frac{L_x}{n_x - 1}, \quad i = 1, \dots, n_x \quad (12.24)$$

$$y_c(j) = (j - 1)\delta y, \quad \delta y = \frac{L_y}{n_y - 1}, \quad j = 1, \dots, n_y. \quad (12.25)$$

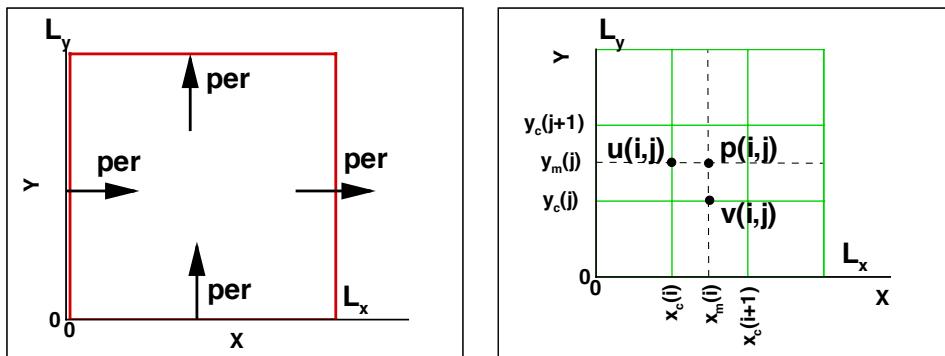


Figure 12.1 Domaine de calcul, conditions aux limites et maillage décalé.

Les centres des mailles forment le maillage *secondaire*, défini par les coordonnées :

$$x_m(i) = (i - 1/2)\delta x, \quad i = 1, \dots, n_{xm} \quad (12.26)$$

$$y_m(j) = (j - 1/2)\delta y, \quad j = 1, \dots, n_{ym}, \quad (12.27)$$

où, pour simplifier la présentation, on a noté :  $n_{xm} = n_x - 1$ ,  $n_{ym} = n_y - 1$ .

Dans la maille  $(i, j)$ , i.e. le rectangle  $[x_c(i), x_c(i+1)] \times [y_c(j), y_c(j+1)]$ , les inconnues  $u$ ,  $v$ ,  $p$  seront évaluées en des points distincts :

- $u(i, j)$  est calculé au point  $(x_c(i), y_m(j))$  (face ouest),
- $v(i, j)$  au point  $(x_m(i), y_c(j))$  (face sud),
- et  $p(i, j)$  au point  $(x_m(i), y_m(j))$  (centre de la maille).

On parle dans ce cas d'une discrétisation sur un maillage *décalé*, cette technique assurant une meilleure stabilité de la méthode numérique décrite précédemment.

## 12.4 DISCRÉTISATION DES ÉQUATIONS

Dans ce paragraphe, l'algorithme 12.1 (page 250) sera écrit sous sa forme discrète, qui sera ensuite programmée. Commençons par observer que les conditions aux limites de périodicité prennent la forme discrète suivante :

$$\begin{aligned} u(1, j) &= u(n_x, j), \quad \forall j = 1, \dots, n_y \\ u(i, 1) &= u(i, n_y), \quad \forall i = 1, \dots, n_x, \end{aligned} \quad (12.28)$$

plus les relations similaires pour  $v$  et  $p$ . Les inconnues du problème seront, par conséquent, les valeurs discrètes :

$$u(i,j), \quad v(i,j), \quad p(i,j), \quad \text{pour } i = 1, \dots, n_{xm}, \quad j = 1, \dots, n_{ym}.$$

Une astuce de programmation pour prendre en compte facilement les conditions aux limites (12.28) dans la discrétisation des opérateurs différentiels est d'introduire les vecteurs suivants :

$$\begin{cases} ip(i) = i + 1, & i = 1, \dots, (n_{xm} - 1) \\ ip(n_{xm}) = 1 \end{cases}, \quad \begin{cases} jp(j) = j + 1, & j = 1, \dots, (n_{ym} - 1) \\ jp(n_{ym}) = 1 \end{cases} \quad (12.29)$$

$$\begin{cases} im(i) = i - 1, & i = 2, \dots, n_{xm} \\ im(1) = n_{xm} \end{cases}, \quad \begin{cases} jm(j) = j - 1, & j = 2, \dots, n_{ym} \\ jm(1) = n_{ym} \end{cases} \quad (12.30)$$

Avec ces notations, on peut écrire les schémas aux différences finies de manière plus compacte et finalement utiliser ces expressions dans la programmation, grâce aux capacités de calcul vectoriel de MATLAB . Par exemple, pour calculer  $\partial\psi/\partial x(i,j)$  pour  $j$  fixé et  $i = 1, \dots, n_{xm}$ , le schéma aux différences finies centrées à l'ordre deux s'écrit

$$\frac{\partial\psi}{\partial x}(i,j) \simeq \frac{\psi(ip(i),j) - \psi(im(i),j)}{2\delta x}, \quad i = 2, \dots, (n_{xm} - 1),$$

avec un traitement particulier pour  $i = 1$  et  $i = n_{xm}$  :

$$\frac{\partial\psi}{\partial x}(1,j) \simeq \frac{\psi(2,j) - \psi(n_{xm},j)}{2\delta x}, \quad \frac{\partial\psi}{\partial x}(n_{xm},j) \simeq \frac{\psi(1,j) - \psi(n_{xm}-1,j)}{2\delta x}.$$

En introduisant les vecteurs  $im$  et  $ip$  (12.29, 12.30), on peut écrire directement :

$$\frac{\partial\psi}{\partial x}(i,j) \simeq \frac{\psi(ip(i),j) - \psi(im(i),j)}{2\delta x}, \quad i = 1, \dots, n_{xm}. \quad (12.31)$$

Comme règle générale, dans les schémas aux différences finies avec des conditions de périodicité, on va remplacer les indices  $(i+1)$  par  $ip(i)$  et  $(i-1)$  par  $im(i)$  (de même pour  $j$ ).

Regardons maintenant la forme discrète de chaque étape de l'algorithme 12.1.

### a) Calcul des termes explicites (A)

Les deux composantes  $\mathcal{H}_u^n$  et  $\mathcal{H}_v^n$  du terme explicite  $\mathcal{H}^n$  (équations 12.14, 12.15) seront calculées aux mêmes points que les composantes de vitesse correspondantes. En utilisant le schéma aux différences finies centrées (12.31), on obtient les discrétisations suivantes (à suivre sur la figure 12.1) :

- pour le calcul de la vitesse  $u$  (position  $(x_c(i), y_m(j))$ ) : pour  $i=1, \dots, n_{xm}, j=1, \dots, n_{ym}$

$$\frac{\partial u^2}{\partial x}(i,j) \simeq \frac{1}{\delta x} \left[ \left( \frac{u(i,j) + u(ip(i),j)}{2} \right)^2 - \left( \frac{u(i,j) + u(im(i),j)}{2} \right)^2 \right] \quad (12.32)$$

$$\begin{aligned} \frac{\partial uv}{\partial y}(i,j) &\simeq \frac{1}{\delta y} \left[ \left( \frac{u(i,j) + u(i,jp(j))}{2} \right) \left( \frac{v(i,jp(j)) + v(im(i),jp(j))}{2} \right) \right. \\ &\quad \left. - \left( \frac{u(i,j) + u(i,jm(j))}{2} \right) \left( \frac{v(i,j) + v(im(i),j)}{2} \right) \right] \end{aligned} \quad (12.33)$$

$$\mathcal{H}_u^n(i,j) = -\frac{\partial u^2}{\partial x}(i,j) - \frac{\partial uv}{\partial y}(i,j), \quad (12.34)$$

- et de manière similaire pour la vitesse  $v$  (position  $(x_m(i), y_c(j))$ ) : pour  $i=1, \dots, n_{xm}, j=1, \dots, n_{ym}$

$$\frac{\partial v^2}{\partial y}(i,j) \simeq \frac{1}{\delta y} \left[ \left( \frac{v(i,j) + v(i,jp(j))}{2} \right)^2 - \left( \frac{v(i,j) + v(i,jm(j))}{2} \right)^2 \right] \quad (12.35)$$

$$\begin{aligned} \frac{\partial uv}{\partial x}(i,j) &\simeq \frac{1}{\delta x} \left[ \left( \frac{u(ip(i),j) + u(ip(i),jm(j))}{2} \right) \left( \frac{v(i,j) + v(ip(i),j)}{2} \right) \right. \\ &\quad \left. - \left( \frac{u(i,j) + u(i,jm(j))}{2} \right) \left( \frac{v(i,j) + v(im(i),j)}{2} \right) \right] \end{aligned} \quad (12.36)$$

$$\mathcal{H}_v^n(i,j) = -\frac{\partial uv}{\partial x}(i,j) - \frac{\partial v^2}{\partial y}(i,j). \quad (12.37)$$

### b) Calcul du champ non-solénoïdal (B)

Observons que les équations (12.16) et (12.17) peuvent s'écrire sous la forme d'une équation de Helmholtz :

$$\underbrace{\left( I - \frac{\delta t}{2Re} \Delta \right)}_{\text{opérateur de Helmholtz}} \delta q^* = \underbrace{\delta t \left[ -\mathcal{G}p^n + \frac{3}{2}\mathcal{H}^n - \frac{1}{2}\mathcal{H}^{n-1} + \frac{1}{Re} \Delta q^n \right]}_{RHS^n}, \quad (12.38)$$

où la notation  $\delta q^* = q^* - q^n$  a été introduite. Pour résoudre cette équation, plusieurs méthodes sont envisageables. Nous allons utiliser la plus simple à mettre en œuvre : la factorisation approximative (ou ADI - *Alternating Direction Implicit* en anglais). L'opérateur de Helmholtz est approché, avec une précision à l'ordre deux en temps, sous la forme :

$$\left( I - \frac{\delta t}{2Re} \Delta \right) \delta q^* \simeq \left( I - \frac{\delta t}{2Re} \frac{\partial^2}{\partial x^2} \right) \left( I - \frac{\delta t}{2Re} \frac{\partial^2}{\partial y^2} \right) \delta q^*, \quad (12.39)$$

où les termes en  $\mathcal{O}(\delta t^2)$  ont été négligés. Cette factorisation est utilisée pour résoudre l'équation (12.38) en deux étapes :

$$\left( I - \frac{\delta t}{2Re} \frac{\partial^2}{\partial x^2} \right) \overline{\delta q^*} = RHS^n \quad (+ \text{ condition de périodicité en } x) \quad (12.40)$$

$$\left( I - \frac{\delta t}{2Re} \frac{\partial^2}{\partial y^2} \right) \delta q^* = \overline{\delta q^*} \quad (+ \text{ condition de périodicité en } y). \quad (12.41)$$

Observons que nous avons associé au champ  $\overline{\delta q^*}$ , qui n'a pas de signification physique, les mêmes conditions aux limites que pour  $\delta q^*$ . Ce choix, qui est naturel pour des conditions de périodicité, nécessite des développements plus poussés pour d'autres types de conditions aux limites (conditions de Dirichlet par exemple).

Les dérivées secondes intervenant dans les équations (12.40) et (12.41) seront approchées par des différences centrées à l'ordre deux, suivant le schéma général :

$$\begin{aligned} \frac{\partial^2 \psi}{\partial x^2}(i,j) &\simeq \frac{\psi(i+1,j) - 2\psi(i,j) + \psi(i-1,j)}{\delta x^2} \\ \frac{\partial^2 \psi}{\partial y^2}(i,j) &\simeq \frac{\psi(i,j+1) - 2\psi(i,j) + \psi(i,j-1)}{\delta y^2}, \end{aligned} \quad (12.42)$$

ce qui conduit à l'algorithme suivant :

### Algorithme 12.2. Calcul de la vitesse $u^*$ par l'algorithme ADI.

- Premier pas d'ADI. Pour chaque  $j = 1, \dots, n_{ym}$  il faut résoudre le système suivant  $i$  :

$$-\beta_x \overline{(\delta u^*)}(i-1,j) + (1+2\beta_x) \overline{(\delta u^*)}(i,j) - \beta_x \overline{(\delta u^*)}(i+1,j) = RHS_u^n(i,j), \quad (12.43)$$

avec  $i = 1, \dots, n_{xm}$  et  $\beta_x = \frac{\delta t}{2Re} \frac{1}{\delta x^2}$  et la convention (imposée par la condition de périodicité)

$$\overline{(\delta u^*)}(0,j) = \overline{(\delta u^*)}(n_{xm},j), \quad \overline{(\delta u^*)}(n_{xm}+1,j) = \overline{(\delta u^*)}(1,j).$$

Il s'agit en fait de  $n_{ym}$  systèmes linéaires à matrice tridiagonale périodique de taille  $n_{xm} \times n_{xm}$  :

$$M_x = \begin{pmatrix} 1 + 2\beta_x & -\beta_x & 0 & \cdot & \cdot & 0 & 0 & -\beta_x \\ -\beta_x & 1 + 2\beta_x & -\beta_x & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & -\beta_x & 1 + 2\beta_x & -\beta_x \\ -\beta_x & 0 & 0 & \cdot & \cdot & 0 & -\beta_x & 1 + 2\beta_x \end{pmatrix} \quad (12.44)$$

Une méthode rapide qui permet de résoudre simultanément ces systèmes sera développée à partir de l'algorithme de Thomas dans les questions préliminaire de ce projet (page 262).

Nous obtenons à la fin de ce pas  $\overline{(\delta u^*)}(i,j)$ .

- Deuxième pas d'ADI. Pour chaque  $i = 1, \dots, n_{xm}$  il faut résoudre le système suivant  $j$  :

$$-\beta_y(\delta u^*)(i, j-1) + (1+2\beta_y)(\delta u^*)(i, j) - \beta_y(\delta u^*)(i, j+1) = \overline{(\delta u^*)}(i, j), \quad (12.45)$$

avec  $j = 1, \dots, n_{ym}$  et  $\beta_y = \frac{\delta t}{2Re} \frac{1}{\delta y^2}$  et la convention (imposée par la condition de périodicité)

$$(\delta u^*)(i, 0) = -(\delta u^*)(i, n_{xm}), \quad (\delta u^*)(i, n_{ym} + 1) = (\delta u^*)(i, 1).$$

Les  $n_{xm}$  systèmes linéaires ont le même type de matrice que précédemment - elle sera de taille  $n_{ym} \times n_{ym}$  :

$$M_y = \begin{pmatrix} 1 + 2\beta_y & -\beta_y & 0 & \cdot & \cdot & 0 & 0 & -\beta_y \\ -\beta_y & 1 + 2\beta_y & -\beta_y & \cdot & \cdot & 0 & 0 & 0 \\ \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & -\beta_y & 1 + 2\beta_y & -\beta_y \\ -\beta_y & 0 & 0 & \cdot & \cdot & 0 & -\beta_y & 1 + 2\beta_y \end{pmatrix} \quad (12.46)$$

Après la résolution des systèmes, nous obtenons  $(\delta u^*)(i, j)$  et immédiatement

$$u^*(i, j) = u(i, j) + (\delta u^*)(i, j).$$

La procédure de calcul sera tout à fait similaire pour l'autre composante de la vitesse :

### Algorithme 12.3. Calcul de la vitesse $v^*$ par l'algorithme ADI.

- Premier pas d'ADI. Pour chaque  $j = 1, \dots, n_{ym}$  il faut résoudre le système suivant  $i$  :

$$M_x \overline{(\delta v^*)}(i, j) = RHS_v^n(i, j), \quad (12.47)$$

avec  $i = 1, \dots, n_{xm}$  et la matrice  $M_x$  donnée par l'équation (12.44).

Nous obtenons à la fin de ce pas  $\overline{(\delta v^*)}(i, j)$ .

- Deuxième pas d'ADI. Pour chaque  $i = 1, \dots, n_{xm}$ , il faut résoudre le système suivant  $j$  :

$$M_y \overline{(\delta v^*)}(i, j) = \overline{(\delta v^*)}(i, j), \quad (12.48)$$

avec  $j = 1, \dots, n_{ym}$  et la matrice  $M_y$  donnée par l'équation (12.46).

Nous obtenons  $(\delta v^*)(i, j)$  et immédiatement

$$v^*(i, j) = v(i, j) + (\delta v^*)(i, j).$$

c) *Résolution de l'équation de Poisson (C)*

L'équation de Poisson (12.12) s'écrit sous la forme discrète suivante :

$$\Delta\phi(i,j) = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)\phi(i,j) = Q(i,j), \quad i = 1, \dots, n_{xm}, \quad j = 1, \dots, n_{ym}, \quad (12.49)$$

avec

$$Q(i,j) = \frac{1}{\delta t} \operatorname{div}(q^*)(i,j) = \frac{1}{\delta t} \left( \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \right)(i,j).$$

Pour résoudre cette équation, on va utiliser d'abord la périodicité de la variable  $\phi$  suivant la direction  $x$ ; on peut donc utiliser une décomposition en série de Fourier

$$\phi(i,j) = \sum_{l=1}^{n_{xm}} \hat{\phi}_l(j) e^{i \frac{2\pi}{n_{xm}} (i-1)(l-1)}, \quad \forall i = 1 \dots n_{xm}, \quad (12.50)$$

où  $i = \sqrt{-1}$  est l'unité imaginaire. L'utilité de cette décomposition est de diagonaliser l'opérateur laplacien et de ramener le problème initial (12.49) à un problème 1D. Effectivement, en utilisant une approximation de  $\partial^2\phi(i,j)/\partial x^2$  par différences centrées, on obtient :

$$\begin{aligned} \frac{\partial^2}{\partial x^2} \phi(i,j) &\simeq \frac{\phi(i+1,j) - 2\phi(i,j) + \phi(i-1,j)}{\Delta x^2} \\ &= \frac{1}{\Delta x^2} \sum_{l=1}^{n_{xm}} \hat{\phi}_l(j) e^{i \frac{2\pi}{n_{xm}} (i-1)(l-1)} \left( e^{i \frac{2\pi}{n_{xm}} (l-1)} - 2 + e^{-i \frac{2\pi}{n_{xm}} (l-1)} \right) \\ &= \sum_{l=1}^{n_{xm}} \hat{\phi}_l(j) e^{i \frac{2\pi}{n_{xm}} (i-1)(l-1)} \underbrace{\frac{2}{\Delta x^2} \left[ \cos\left(\frac{2\pi}{n_{xm}}(l-1)\right) - 1 \right]}_{k_l} \end{aligned} \quad (12.51)$$

Le second membre  $Q$  étant aussi périodique, il est également décomposé en série de Fourier,

$$Q(i,j) = \sum_{l=1}^{n_{xm}} \hat{Q}_l(j) e^{i \frac{2\pi}{n_{xm}} (i-1)(l-1)},$$

ce qui fait que l'équation initiale (12.49) est équivalente à la résolution, pour chaque nombre d'onde  $l = 1, \dots, n_{xm}$ , d'une équation de la forme :

$$\frac{\partial^2}{\partial y^2} \hat{\phi}_l(j) + k_l \hat{\phi}_l(j) = \hat{Q}_l(j). \quad (12.52)$$

À partir de ce point, nous avons le choix entre deux méthodes : utiliser une décomposition de Fourier suivant  $y$ , ou discréteriser directement la dérivée seconde suivant  $y$  par des différences finies centrées.

Nous choisissons la deuxième méthode, qui a l'avantage de pouvoir être utilisée dans le cas d'une condition à la limite différente suivant  $y$  (condition de paroi par

exemple). L'équation (12.52) devient :

$$\frac{1}{\delta y^2} \hat{\phi}_I(j-1) + \left( -\frac{2}{\delta y^2} + k_I \right) \hat{\phi}_I(j) + \frac{1}{\delta y^2} \hat{\phi}_I(j+1) = \hat{Q}_I(j), \quad j = 1, \dots, n_{ym}. \quad (12.53)$$

Pour cette dernière équation nous avons besoin de conditions aux limites pour  $j=1$  et  $j=n_{ym}$ . Dans notre cas, la condition de périodicité est naturellement imposée.

Un traitement particulier doit être réservé au nombre d'onde  $I=1$  pour lequel  $k_I=0$ . Dans ce cas, il est facile de voir que la matrice du système (12.53) est singulière, ce qui traduit le fait que le problème est mal posé ! Effectivement, la condition de périodicité fait que la solution du laplacien est déterminée à une constante près : cette constante est donnée exactement par la valeur du premier coefficient de la série de Fourier (la valeur moyenne). Nous pouvons donc imposer librement cette constante sans aucune incidence sur les calculs, car, même si la variable  $\phi$  va déterminer le champ de pression à une constante près, dans les équations de Navier-Stokes seulement le gradient de la pression intervient !

Il est donc naturel d'imposer pour  $I=1$ ,  $\hat{\phi}_I(j)=0, j=1, \dots, n_{ym}$  (moyenne nulle).

**Algorithme 12.4.** En pratique, la résolution de l'équation de Poisson comportera les étapes suivantes (points de coordonnées  $(x_m(i), y_m(j))$ ) :

- on calcule le tableau  $Q(i,j), i=1, \dots, n_{xm}, j=1, \dots, n_{ym}$  :

$$Q(i,j) = \frac{1}{\delta t} \left( \frac{u^*(ip(i),j) - u^*(i,j)}{\delta x} + \frac{v^*(i,jp(i)) - v^*(i,j)}{\delta y} \right); \quad (12.54)$$

- on effectue une transformée de Fourier rapide (FFT) de chaque colonne du tableau  $Q$

$$\hat{Q}(I,j) = FFT(Q(i,j)), \quad I = 1, \dots, n_{xm}, \quad j = 1, \dots, n_{ym} \quad (12.55)$$

(Attention, les valeurs  $\hat{Q}$  sont complexes !) ;

- pour  $I=1$  on impose  $\hat{\phi}_1(j)=0, j=1, \dots, n_{ym}$  ;
- pour chaque  $I=2, \dots, n_{xm}$  on résout le système  $M_I \hat{\phi}_I = \hat{Q}(I,j)^T$  à matrice tridiagonale ( $n_{ym} \times n_{ym}$ ) :

$$M_I = \frac{1}{\delta y^2} \begin{bmatrix} -2 + \delta y^2 k_I & 1 & 0 & \cdot & \cdot & 0 & 0 & 1 \\ 1 & -2 + \delta y^2 k_I & 1 & \cdot & \cdot & 0 & 0 & \cdot \\ \cdot & \cdot \\ 0 & 0 & 0 & \cdot & \cdot & 1 & -2 + \delta y^2 k_I & 1 \\ 1 & 0 & 0 & \cdot & \cdot & 0 & 1 & -2 + \delta y^2 k_I \end{bmatrix}, \quad (12.56)$$

avec

$$k_I = \frac{2}{\Delta x^2} \left[ \cos \left( \frac{2\pi}{n_{xm}} (I-1) \right) - 1 \right];$$

- on construit le tableau  $\hat{\Phi}(I,j)$ , dont les lignes sont les vecteurs  $\hat{\phi}_I$  ;

- enfin, par une transformée de Fourier inverse (IFFT), on obtient la solution

$$\phi(i,j) = \text{IFFT}(\hat{\Phi}(l,j)), \quad i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}. \quad (12.57)$$

#### d) Calcul du champ solénoïdal ( $D$ )

Une fois le tableau  $\phi(i,j)$  calculé, il est très facile de corriger le champ de vitesse :

- pour  $i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}$

$$u^{n+1}(i,j) = u^*(i,j) - \delta t \frac{\phi(i,j) - \phi(im(i),j)}{\Delta x}, \quad (12.58)$$

- pour  $i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}$

$$v^{n+1}(i,j) = v^*(i,j) - \delta t \frac{\phi(i,j) - \phi(i,jm(j))}{\Delta y}.. \quad (12.59)$$

#### e) Calcul du champ de pression ( $E$ )

Le calcul du nouveau champ de pression est immédiat suivant l'équation (12.21) :

- pour  $i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}$

$$p^{n+1}(i,j) = p^n(i,j) + \phi(i,j) - \frac{\delta t}{2Re} \left[ \frac{\phi(ip(i),j) - 2\phi(i,j) + \phi(im(i),j)}{\delta x^2} + \frac{\phi(i,jp(j)) - 2\phi(i,j) + \phi(i,jm(j))}{\delta y^2} \right] \quad (12.60)$$

Le gradient de pression doit être également réactualisé :

- pour  $i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}$

$$\frac{\partial p^{n+1}}{\partial x}(i,j) = \frac{p^{n+1}(i,j) - p^{n+1}(im(i),j)}{\delta x}, \quad (12.61)$$

- pour  $i = 1, \dots n_{xm}, \quad j = 1, \dots n_{ym}$

$$\frac{\partial p^{n+1}}{\partial y}(i,j) = \frac{p^{n+1}(i,j) - p^{n+1}(i,jm(j))}{\delta y}. \quad (12.62)$$

#### f) Calcul du pas de temps

Pour que l'algorithme de résolution soit complet, il faut indiquer comment calculer le pas de temps  $\delta t$ . Comme nous utilisons un schéma semi-implicite, la valeur du pas de temps sera limitée par une condition de type CFL (condition de stabilité du schéma, voir le Projet 1, page 1). La dérivation de cette condition par une analyse de stabilité étant assez complexe, la valeur du pas de temps sera calculée à partir de la

relation générale :

$$dt = \frac{cfl}{\max\left(\left|\frac{u}{\delta x}\right| + \left|\frac{v}{\delta y}\right|\right)}, \quad (12.63)$$

avec la constante  $cfl < 1$ . Afin d'optimiser les différentes procédures de calcul, nous allons utiliser un pas de temps constant, évalué à partir de la condition initiale.

## 12.5 VISUALISATION DE L'ÉCOULEMENT

Nous disposons de deux moyens rapides de visualisation de l'évolution de l'écoulement. Le premier consiste à calculer le champ de vorticité, suivant la définition suivante :

$$\omega = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \quad (12.64)$$

avec les valeurs discrètes calculées aux points  $(x_c(i), y_c(j))$  par :

$$\omega(i, j) = \frac{v(i, j) - v(im(i), j)}{\delta x} - \frac{u(i, j) - u(i, jm(j))}{\delta y}. \quad (12.65)$$

Les iso-valeurs (ou lignes de niveau) de la vorticité nous permettront d'identifier les tourbillons dans le champ de l'écoulement (les tourbillons étant définis comme concentrations de vorticité).<sup>4</sup>

Un autre moyen de visualisation est de suivre l'évolution d'un scalaire (traceur) passif dans l'écoulement. Le scalaire passif, comme son nom l'indique, est simplement transporté par le champ de vitesse sans influencer l'évolution de l'écoulement. Il est l'équivalent numérique du colorant utilisé dans les visualisations expérimentales.

L'évolution du scalaire passif  $\chi$  est décrite par une équation de convection-diffusion :

$$\frac{\partial \chi}{\partial t} + \frac{\partial \chi u}{\partial x} + \frac{\partial \chi v}{\partial y} = \frac{1}{Pe} \Delta \chi, \quad (12.66)$$

où le paramètre de similitude  $Pe$  (le nombre de Peclet) caractérise les propriétés de diffusion du traceur passif. L'équation (12.66) sera résolue suivant exactement le même schéma (Adams-Bashfort + Crank-Nicolson) utilisé pour les équations de quantité de mouvement. Cette résolution interviendra après le calcul du champ solenoïdal.

Pour voir des images illustrant les structures tourbillonnaires présentes dans différents écoulements fluides réels, le lecteur peut consulter [Lesieur, 1994].

---

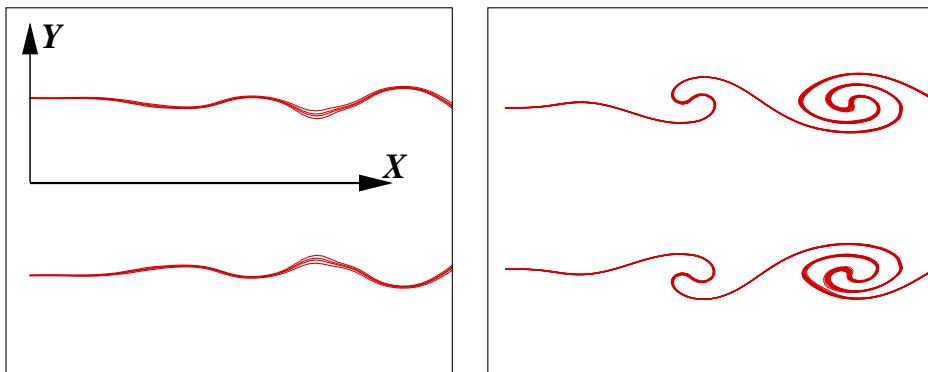
4. MATLAB fournit de fonctions qui tracent les iso-valeurs pour un champ 2D (fonctions `contour` ou `pcolor`).

## 12.6 CONDITION INITIALE

Nous avons vu comment avancer la solution en temps. Il nous reste à spécifier la condition initiale qui va préciser le type d'écoulement étudié. Le champ initial doit être compatible avec les équations de Navier-Stokes ; en pratique, il suffit de prescrire le champ dynamique (vitesses  $u$  et  $v$ ) et garder une pression nulle partout. Le champ de pression correct s'établira rapidement au cours du calcul. Nous allons simuler deux types d'écoulements.

### 12.6.1 Dynamique d'un jet plan. Instabilité de Kelvin-Helmholtz

L'instabilité de Kelvin-Helmholtz se produit lorsqu'il existe dans le champ de l'écoulement un cisaillement entre les couches fluides. Une expérience très simple pour mettre en évidence cette instabilité consiste à disposer dans un long parallélépipède un mélange de deux liquides dont l'un est plus lourd que l'autre. Une simple inclinaison du parallélépipède provoque l'écoulement de la partie plus lourde vers le bas en repoussant la partie légère vers le haut. On observe une ondulation de la surface de séparation qui s'amplifie pour donner naissance à de beaux tourbillons, semblables à des vagues (voir figure 12.2).



**Figure 12.2** Représentation schématique du développement de l'instabilité de Kelvin-Helmholtz : perturbation de la couche de cisaillement (à gauche) et enroulement des tourbillons (à droite).

Pour observer numériquement ce phénomène, nous allons imposer un cisaillement dans le champ fluide en introduisant un profil de vitesse avec un point d'inflexion.

La condition initiale correspond à un écoulement de jet plan :

$$v(x, y) = 0, \quad u(x, y) = u_1(y)(1 + u_2(x))$$

avec le profil de base ( $P_j$  est le paramètre du jet et  $R_j$  le rayon du jet) :

$$u_1(y)/U_0 = \frac{1}{2} \left( 1 + \tanh \left( \frac{1}{2} P_j \left( 1 - \frac{|L_y/2 - y|}{R_j} \right) \right) \right); \quad (12.67)$$

et la perturbation (nécessaire pour engendrer l'instabilité de Kelvin-Helmholtz) :

$$u_2(x) = A_x \sin\left(2\pi \frac{x}{\lambda_x}\right). \quad (12.68)$$

### 12.6.2 Mouvement d'un dipôle de vorticité

Un dipôle de vorticité est une paire de deux tourbillons de vorticités égales et de signes opposés (sens de rotation opposés). Comme chaque tourbillon induit dans le champ de l'écoulement une vitesse orientée suivant son sens de rotation et proportionnelle à son intensité, le dipôle sera une structure stable qui se déplace suivant son axe de symétrie avec une vitesse constante.

Nous allons construire un dipôle de vorticité en juxtaposant deux tourbillons individuels suivant un modèle très simple. Un tourbillon centré en  $(x_v, y_v)$ , de taille  $I_v$  et intensité  $\psi_0$ , est décrit analytiquement par la fonction de courant :

$$\psi(x, y) = \psi_0 \exp\left(-\frac{(x - x_v)^2 + (y - y_v)^2}{I_v^2}\right), \quad (12.69)$$

ce qui nous permet de calculer les deux composantes du vecteur vitesse par :

$$\begin{cases} u = \frac{\partial \psi}{\partial y} = -2 \frac{(y - y_v)}{I_v^2} \psi(x, y) \\ v = -\frac{\partial \psi}{\partial x} = 2 \frac{(x - x_v)}{I_v^2} \psi(x, y). \end{cases} \quad (12.70)$$

Un dipôle sera maintenant construit par la superposition des champs de vitesse de deux tourbillons de même taille et intensité, distribués symétriquement par rapport à une direction choisie. Par exemple, un dipôle se propageant suivant l'axe des  $x$ , peut être obtenu en superposant (voir fig. 12.3) deux tourbillons définis par :

tourbillon 1 :  $+\psi_0, I_v, x_v, y_v = L_y + a$

tourbillon 2 :  $-\psi_0, I_v, x_v, y_v = L_y - a$ ,

avec  $a$  une distance définissant la distance entre les tourbillons.

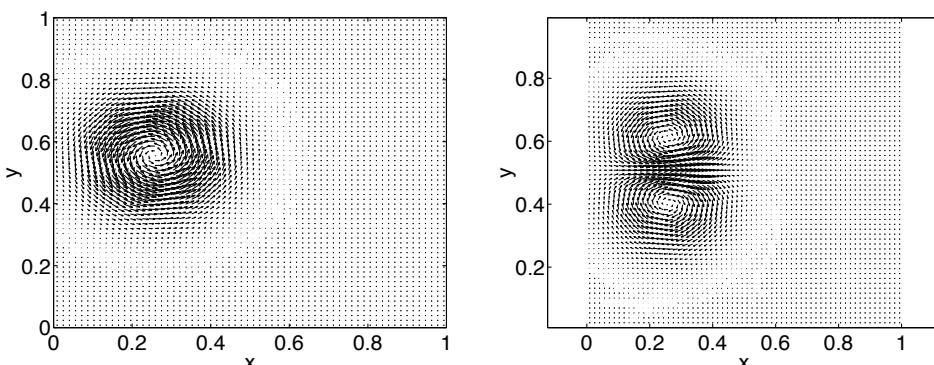


Figure 12.3 Champ de vitesse induit par un seul tourbillon (à gauche) et un dipôle (à droite).

Il faut préciser que le champ de vitesse ainsi construit ne vérifie pas exactement les équations de Navier-Stokes. Il sera corrigé automatiquement par le schéma numérique après le premier pas de temps.

## 12.7 MISE EN ŒUVRE

La mise en œuvre de ce projet étant assez laborieuse, nous allons construire progressivement les différents modules qui vont constituer le programme final. Chaque module sera validé sur des problèmes tests plus simples.

### 12.7.1 Résolution d'un système linéaire à matrice tridiagonale et périodique

La forme particulière (*tridiagonale, périodique*) des matrices (12.44, 12.46, 12.56) sera exploitée pour programmer des fonctions efficaces de résolution des systèmes linéaires correspondants. Les programmes seront basés sur l'algorithme de Thomas, présenté d'abord pour les matrices tridiagonales et ensuite pour des matrices tridiagonales et périodiques.

**Algorithme 12.5.** *Algorithme de Thomas pour la résolution d'un système linéaire à matrice tridiagonale<sup>5</sup>.*

*Le système :*

$$\begin{pmatrix} b_1 & c_1 & 0 & . & . & 0 & 0 \\ a_2 & b_2 & c_2 & 0 & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & a_{n-1} & b_{n-1} & c_{n-1} \\ 0 & 0 & 0 & 0 & . & a_n & b_n \end{pmatrix} \begin{pmatrix} X_1 \\ X_2 \\ . \\ . \\ X_{n-1} \\ X_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ . \\ . \\ f_{n-1} \\ f_n \end{pmatrix}$$

*est résolu en introduisant la récurrence :*

$$\begin{cases} X_k = \gamma_k - \frac{c_k}{\beta_k} X_{k+1}, & k = 1 \dots (n-1) \\ X_n = \gamma_n \end{cases} \quad (12.71)$$

*En utilisant ces relations dans le système initial, on peut déterminer les coefficients  $\gamma_k$  et  $\beta_k$  :*

$$\begin{cases} \beta_1 = b_1 \\ \beta_k = b_k - \frac{c_{k-1}}{\beta_{k-1}} a_k, & k = 2 \dots n \\ \gamma_1 = \frac{f_1}{\beta_1} = \frac{f_1}{b_1} \\ \gamma_k = \frac{f_k - a_k \gamma_{k-1}}{\beta_k}, & k = 2 \dots n \end{cases}$$

5. On peut démontrer qu'il s'agit d'une forme particulière de l'algorithme de Gauss.

Une fois les coefficients  $\gamma_k$  et  $\beta_k$  calculés, les inconnues  $X_k$  seront calculées par substitution inverse en (12.71), à partir de la valeur de  $X_n$ .

**Algorithme 12.6.** Algorithme de Thomas pour la résolution d'un système linéaire à matrice tridiagonale et périodique.

Le système

$$\left( \begin{array}{ccccccc|c} b_1 & c_1 & 0 & \cdot & 0 & 0 & |a_1 \\ a_2 & b_2 & c_2 & \cdot & 0 & 0 & |0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & |0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & |0 \\ 0 & 0 & 0 & \cdot & a_{n-1} & b_{n-1} & |c_{n-1} \\ \hline c_n & 0 & 0 & \cdot & 0 & a_n & |b_n \end{array} \right) \left( \begin{array}{c} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{n-1} \\ \hline X_n \end{array} \right) = \left( \begin{array}{c} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_{n-1} \\ \hline f_n \end{array} \right)$$

est écrit sous la forme<sup>6</sup> :

$$\left( \begin{array}{ccccccc|c} b_1^* & c_1 & 0 & \cdot & 0 & 0 & 0 & |v_1 \\ a_2 & b_2 & c_2 & \cdot & 0 & 0 & 0 & |0 \\ \cdot & |0 \\ \cdot & |0 \\ 0 & 0 & 0 & \cdot & a_{n-1} & b_{n-1} & c_{n-1} & |0 \\ 0 & 0 & 0 & \cdot & 0 & a_n & b_n^* & |v_n \\ \hline -1 & 0 & 0 & \cdot & 0 & 0 & -1 & |1 \end{array} \right) \left( \begin{array}{c} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_{n-1} \\ \hline X_n \\ X^* \end{array} \right) = \left( \begin{array}{c} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_{n-1} \\ f_n \\ \hline 0 \end{array} \right) \quad (12.72)$$

avec :

$$\left\{ \begin{array}{lcl} v_1 & = & a_1 \\ v_n & = & c_n \\ b_1^* & = & b_1 - a_1 \\ b_n^* & = & b_n - c_n \end{array} \right. \quad \text{et} \quad X^* = X_1 + X_n$$

Une forme équivalente de (12.72) est :

$$\underbrace{\left( \begin{array}{ccccccc|c} b_1^* & c_1 & 0 & \cdot & 0 & 0 & 0 & | \\ a_2 & b_2 & c_2 & 0 & \cdot & 0 & 0 & | \\ \cdot & | \\ 0 & 0 & 0 & 0 & \cdot & a_n & b_n^* & | \end{array} \right)}_{M^*} \left( \begin{array}{c} X_1 \\ X_2 \\ \cdot \\ \cdot \\ X_n \end{array} \right) + \left( \begin{array}{c} v_1 \\ 0 \\ \cdot \\ \cdot \\ v_n \end{array} \right) X^* = \left( \begin{array}{c} f_1 \\ f_2 \\ \cdot \\ \cdot \\ f_n \end{array} \right)$$

et

$$X^* = X_1 + X_n.$$

La solution est cherchée sous la forme :

$$X_k = X_k^{(1)} - X_k^{(2)} \cdot X^*, \quad k = 1 \dots n \quad (12.73)$$

6. Cette méthode est basée sur une formule similaire à celle de Sherman-Morrison, voir [Joly, 2004].

ce qui nous amène à résoudre deux systèmes à matrice tridiagonale de dimension  $n$  :

$$\begin{cases} M^* \cdot X^{(1)} = (f_1 \ f_2 \ \dots \ f_{n-1} \ f_n)^T \\ M^* \cdot X^{(2)} = (v_1 \ 0 \ \dots \ 0 \ v_n)^T \end{cases} \quad (12.74)$$

L'inconnue supplémentaire introduite est calculée par :

$$X^* = \frac{X_1^{(1)} + X_n^{(1)}}{1 + X_1^{(2)} + X_n^{(2)}} \quad (12.75)$$

Pour résumer, les étapes de l'algorithme seront :

- résoudre les deux systèmes (12.74) par l'algorithme de Thomas (12.5) (cette étape peut être optimisée car les deux systèmes ont la même matrice  $M^*$ ) ;
- calculer  $X^*$  de (12.75) ;
- trouver la solution par (12.73).

## Exercice 12.1

Écrire une fonction

```
function fi=trid_per_c2D(aa,ab,ac,fi)
```

qui résout simultanément  $m$  systèmes à matrice tridiagonale et périodique. L'algorithme 12.6 sera utilisé pour chaque système  $j$  (avec  $1 \leq j \leq m$ ) décrit par les équations :

$$\text{aa}(j,i)*X(j,i-1)+\text{ab}(j,i)*X(j,i)+\text{ac}(j,i)*X(j,i+1)=\text{fi}(j,i), \\ i=1,\dots,n$$

avec la condition de périodicité  $X(j, 1) = X(j, n)$ .

Indications :

- on peut utiliser le calcul vectoriel sous MATLAB pour appliquer simultanément les formules de l'algorithme à tous les  $m$  systèmes ; par exemple, le calcul des coefficients  $b_1^*, b_n^*$  de la matrice  $M^*$  (12.72) s'écrit simplement

$$\text{ab}(:,1)=\text{ab}(:,1)-\text{aa}(:,1); \\ \text{ab}(:,n)=\text{ab}(:,n)-\text{ac}(:,n);$$

et les calculs seront effectués pour tous les  $j$  tels que  $1 \leq j \leq m$ .

- les coefficients  $\beta_k$  et  $\gamma_k$  correspondant à la résolution des systèmes (12.74) par l'algorithme de Thomas seront calculés une seule fois, car la matrice est la même pour les deux systèmes.

Tester la fonction en utilisant comme modèle le programme MATLAB *test\_trid.m*.<sup>7</sup>

La solution de cet exercice se trouve à la page 271.

### 12.7.2 Résolution de l'équation instationnaire de la chaleur

Les fonctions nécessaires à l'intégration des équations de Navier-Stokes seront développées dans le cas plus simple de l'équation instationnaire de la chaleur (voir aussi le Projet 1 pour la forme 1D) :

$$\frac{\partial u}{\partial t} - \Delta u(t, x, y) = f(x, y), \quad \text{sur } \Omega = [0, L_x] \times [0, L_y], \quad (12.76)$$

avec des conditions de périodicité dans les deux directions de l'espace et une condition initiale  $u(0, x, y) = u^0(x, y)$  donnée. L'équation de la chaleur sera intégrée en temps jusqu'à l'obtention d'un état stationnaire (ou permanent), solution de l'équation :

$$-\Delta u_s(x, y) = f(x, y), \quad \text{sur } [0, L_x] \times [0, L_y], \quad (12.77)$$

avec les mêmes conditions de périodicité. La solution stationnaire  $u_s(x, y)$  de l'équation (12.77) est interprétée comme la limite pour  $t \rightarrow \infty$  de la solution  $u(t, x, y)$  de (12.76).

Afin de tester les programmes développés dans ce paragraphe, nous allons considérer

$$f(x, y) = (a^2 + b^2) \sin(ax) \cos(by), \quad \text{avec } a = \frac{2\pi}{L_x}, \quad b = \frac{2\pi}{L_y}, \quad (12.78)$$

qui vérifie bien les conditions de périodicité. Il est évident que cette forme du second membre a été choisie pour que la solution exacte de l'équation (12.77) soit :

$$u_{ex}(x, y) = \sin(ax) \cos(by). \quad (12.79)$$

La solution numérique obtenue sera systématiquement comparée à cette solution exacte (analytique).

#### a) Méthode explicite

Pour intégrer en temps l'équation (12.76), le schéma le plus simple est le schéma d'Euler explicite (voir le Projet 1) :

$$u^{n+1} = u^n + \delta t(f + \Delta u^n). \quad (12.80)$$

---

7. Quelques commentaires sur ce programme : les éléments des tableaux `aa`, `ab`, `ac` sont définis de manière aléatoire (fonction `rand`) ; pour chaque  $j$ , la matrice  $A$  du système est reconstituée ; la matrice  $A$  est rendue à diagonale dominante pour s'assurer que le système soit inversible ; le second membre du système est calculé par  $f = A * \tilde{X}$ , avec  $\tilde{X}$  imposé ; chaque système est résolu en utilisant les fonctions MATLAB `\`, pour la syntaxe  $X = A \backslash f$  ; la fonction `trid_per_c2D` est validée si la solution renournée est exactement  $\tilde{X}$ .

En supposant que la solution est calculée aux points de coordonnées  $(x_c, y_m)$  (figure 12.1), la discrétisation spatiale sera ( $i = 1, \dots, n_{xm}, j = 1, \dots, n_{ym}$ ) :

$$\begin{aligned} u^{n+1}(i, j) = u^n(i, j) + \delta t & \left[ f(i, j) + \frac{u(ip(i), j) - 2u(i, j) + u(im(i), j)}{\delta x^2} \right. \\ & \left. + \frac{u(i, jp(j)) - 2u(i, j) + u(i, jm(j))}{\delta y^2} \right]. \end{aligned} \quad (12.81)$$

La solution sera avancée en temps à partir d'une condition initiale  $u^0 = 0$ , jusqu'à la *convergence* (obtention de la solution stationnaire) exprimée par le critère numérique :

$$\varepsilon = \|u^{n+1} - u^n\|_2 < 10^{-6}, \quad (12.82)$$

avec la norme définie par  $\|\varphi\|_2 = (\int_{\Omega} \varphi^2 dx dy)^{1/2}$ .

L'inconvénient de ce schéma est la limitation sur le pas de temps imposée par la condition de stabilité (voir la bibliographie complémentaire) :

$$\delta t \left( \frac{1}{\delta x^2} + \frac{1}{\delta y^2} \right) = \frac{cfl}{2}, \quad cfl \leq 1. \quad (12.83)$$

## Exercice 12.2

Résoudre numériquement l'équation (12.76) en utilisant le schéma explicite (12.81). Comparer la solution numérique obtenue avec la solution exacte (12.79) en traçant sur une même figure les iso-contours (sous MATLAB, utiliser la fonction **contour**) des deux solutions. Données numériques :  $L_x = 1, L_y = 2, n_x = 21, n_y = 51, cfl = 1$ . Indications :

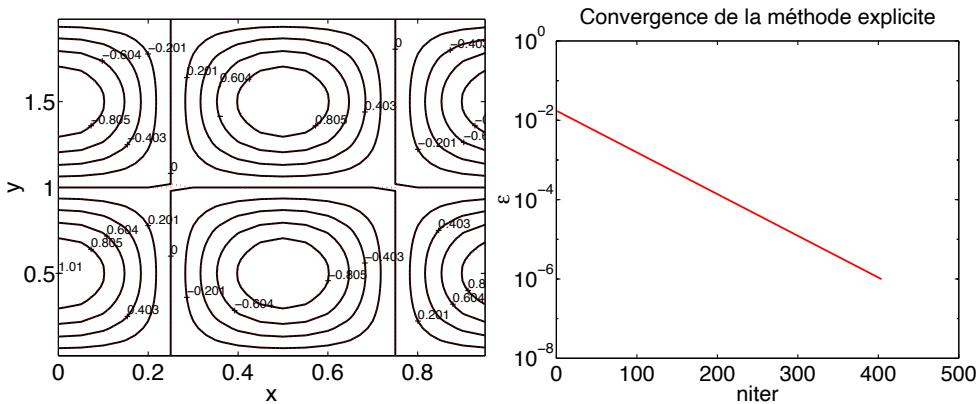
- les données concernant le maillage peuvent être définies comme variables globales ;
- la programmation doit être modulaire, pour pouvoir utiliser les fonctions par la suite (écrire, par exemple, des fonctions pour calculer  $f, \Delta u^n$ , pour la visualisation, etc.) ;
- utiliser une boucle **while** pour l'avancement en temps ;
- éviter les boucles suivant les indices  $i$  et  $j$  en utilisant les opérations vectorielles - par exemple la fonction qui calcule  $\Delta u^n$  peut s'écrire ( $u$  étant un tableau de dimensions  $n_{xm} \times n_{ym}$  et  $im, jm, ip, jp$ , les tableaux des indices donnés par 12.30 et 12.29) :

```
function hc=calc_lap(u)
global dx dy
global im ip jp jm ic jc

hc = (u(ip,jc)-2*u+u(im,jc))/(dx*dx)
    +(u(ic,jp)-2*u+u(ic,jm))/(dy*dy);
```

La solution de cet exercice se trouve à la page 271.

Un exemple de résultat est représenté sur la figure 12.4. La méthode explicite est facile à programmer, mais la convergence est lente à cause de la limitation sur le pas de temps et, par conséquent, le temps de calcul est important. Pour remédier à cet inconvénient, nous allons utiliser une méthode implicite pour résoudre le même problème.



**Figure 12.4** Test de la résolution de l'équation stationnaire de la chaleur. Superposition des iso-valeurs de la solution numérique et analytique (à gauche) et convergence de la méthode explicite (à droite).

### b) Méthode implicite

Les schémas d'Adams-Bashfort et Crank-Nicolson appliqués à l'équation (12.76) conduisent à :

$$\frac{u^{n+1} - u^n}{\delta t} = \underbrace{\frac{3}{2}\mathcal{H}^n - \frac{1}{2}\mathcal{H}^{n-1}}_{\text{Adams-Bashfort}} + \underbrace{\frac{1}{2}\Delta(u^{n+1} + u^n)}_{\text{Crank-Nicolson}} \quad (12.84)$$

avec, dans notre cas  $\mathcal{H}^n = \mathcal{H}^{n-1} = f(x, y)$ , qui ne dépend pas du temps. L'équation à résoudre devient :

$$\left( I - \frac{\delta t}{2} \Delta \right) \delta u = \delta t(f + \Delta u^n), \quad \text{avec} \quad \delta u = u^{n+1} - u^n. \quad (12.85)$$

La factorisation ADI pour cette équation s'écrit sous la forme :

$$\begin{cases} \left( I - \frac{\delta t}{2} \frac{\partial^2}{\partial x^2} \right) \overline{\delta u} = \delta t(f + \Delta u^n) & +\text{périodicité en } x \\ \left( I - \frac{\delta t}{2} \frac{\partial^2}{\partial y^2} \right) \delta u = \overline{\delta u} & +\text{périodicité en } y. \end{cases} \quad (12.86)$$

La discréétisation de ces équations conduit à des systèmes linéaires à matrice tridiagonale et périodique (voir l'algorithme 12.2) qui seront résolus en utilisant la fonction `trid_per_c2D`, écrite précédemment.

La méthode implicite nécessite plus de calculs pour inverser les systèmes linéaires, mais son coût global est inférieur à celui de la méthode explicite car il n'y a plus de restriction sur le pas de temps  $\delta t$  (la méthode implicite est inconditionnellement stable).

### Exercice 12.3

Reprendre l'exercice 12.2 en utilisant la méthode implicite. Le pas de temps sera calculé par la formule (12.83), avec  $cfl = 100$  ! Comparer le temps d'exécution avec celui de la méthode explicite.

En observant que, pour un pas de temps constant, les coefficients des matrices intervenant dans la méthode ADI ne changent pas, optimiser la fonction `trid_per_c2D`

- en réduisant la taille des tableaux qui stockent les coefficients des matrices,
- en effectuant les opérations qui ne dépendent pas du second membre du système une seule fois, en dehors de la boucle (`while`) en temps.

La solution de cet exercice se trouve à la page 271.

### Exercice 12.4

Considérons maintenant l'équation de convection-diffusion non-linéaire :

$$\frac{\partial u}{\partial t} + \frac{\partial u^2}{\partial x} - \Delta u = f(x, y), \quad \text{sur } [0, L_x] \times [0, L_y], \quad (12.87)$$

avec des conditions aux limites de périodicité et une condition initiale  $u^0(x, y) = 0$ .

1. Déterminer l'expression analytique du second membre  $f(x, y)$  pour que la solution stationnaire de (12.87) soit (12.79).
2. Retrouver numériquement cette solution en utilisant la méthode implicite (12.84). La seule différence par rapport au programme précédent est le calcul du terme  $\mathcal{H}$  (attention,  $\mathcal{H}^n \neq \mathcal{H}^{n-1}$ ).
3. Le pas de temps étant considéré constant et donné par (12.83), trouver la limite de stabilité ( $cfl_{max}$ ) pour la résolution spatiale donnée.

La solution de cet exercice se trouve à la page 272.

### 12.7.3 Résolution de l'équation stationnaire de la chaleur en utilisant les FFT

Les modules nécessaires à la résolution de l'équation de Poisson (algorithme 12.4) seront testés dans le cas plus simple de l'équation de la chaleur stationnaire (12.77).

#### Exercice 12.5

1. Résoudre l'équation (12.77) avec le second membre (12.78) en utilisant la méthode décrite pour la résolution de l'équation de Poisson (page 256) (une transformée de Fourier rapide + la résolution d'un système tridiagonal). Comparer avec la solution exacte.<sup>8</sup>  
Données numériques :  $L_x = 1$ ,  $L_y = 2$ ,  $n_x = 65$ ,  $n_y = 129$ .
2. Optimiser la partie résolution du système tridiagonal (cf. exercice 12.3).
3. (optionnel) Résoudre la même équation en utilisant deux FFT.

La solution de cet exercice se trouve à la page 272.

### 12.7.4 Résolution des équations de Navier-Stokes

Les modules les plus importants pour la résolution des équations de Navier-Stokes étant développés, il reste à les assembler pour simuler les écoulements décrits dans le paragraphe 12.6.

#### Exercice 12.6

Écrire le programme pour la résolution des équations de Navier-Stokes 2D, avec des conditions aux limites de périodicité. Le programme sera modulaire et sera organisé comme suit :

- définition des variables globales ;
- données du programme et définition de la condition initiale ;
- construction du maillage ;
- définition des tableaux pour les variables principales et initialisation à zéro ;
- construction du champ initial en fonction du cas de calcul (voir plus bas) et visualisation ;
- calcul du pas de temps ;
- calcul des variables nécessaires à l'optimisation de la méthode ADI et à l'optimisation de la résolution de l'équation de Poisson ;

8. La solution numérique étant calculée à une constante près, pour comparer avec la solution exacte il faut calculer cette constante en imposant que les deux solutions soient les mêmes en un point choisi ( $i=j=1$ , par exemple). Il faut donc comparer  $u_{ex}$  et  $u_{num} + (u_{ex}(1, 1) - u_{num}(1, 1))$ .

- début boucle en temps :
  - résolution de l'équation de quantité de mouvement pour  $u$ ;
  - résolution de l'équation de quantité de mouvement pour  $v$ ;
  - calcul de la divergence du champ non-solénoïdal ;
  - résolution de l'équation de Poisson ;
  - correction du champ de vitesse ;
  - calcul de la pression ;
  - calcul du nouveau gradient de pression ;
  - résolution de l'équation pour le scalaire passif ;
  - vérification de la divergence du champ de vitesse ;
  - visualisation de la vorticité et scalaire passif ;
- fin boucle en temps.

La solution de cet exercice se trouve à la page 272.

### Cas de calcul<sup>9</sup>

1. Jet plan - instabilité de Kelvin-Helmholtz : paramètres

$$L_z = 2, L_y = 1, n_x = 65, n_y = 65, cfl = 0, 2$$

$$Re = 1000, Pe = 1000, U_0 = 1, P_j = 20, R_j = L_y/4, A_x = 0, 5, \lambda_x = 0, 5L_x.$$

La condition initiale pour le scalaire passif est identique à celle de la vitesse  $u$ .

2. Même configuration que précédemment, sauf  $cfl = 0, 1, \lambda_x = 0, 25L_x$ .

3. Dipôle de vorticité :

$$L_z = 1, L_y = 1, n_x = 65, n_y = 65, cfl = 0, 4, Re = 1000, Pe = 1000$$

tourbillon 1 :

$$\psi_0 = +0, 01, x_v = L_x/4, y_v = L_y/2 + 0, 05, I_v = 0, 4\sqrt{2} \min \{x_v, y_v, L_x - x_v, L_y - y_v\}$$

tourbillon 2 :

$$\psi_0 = -0, 01, x_v = L_x/4, y_v = L_y/2 - 0, 05, I_v = 0, 4\sqrt{2} \min \{x_v, y_v, L_x - x_v, L_y - y_v\}$$

Le scalaire passif sera initialisé sous la forme d'une bande constante, placée au milieu du domaine de calcul

$$\begin{cases} \chi(i, j) = 1, & \text{si } n_{xm}/2 - 10 \leq i \leq n_{xm}/2 + 10, \\ \chi(i, j) = 0, & \text{autrement.} \end{cases}$$

4. Même configuration que précédemment + un dipôle similaire qui se propage en sens inverse.

5. Imaginer une autre configuration avec plusieurs dipôles de vorticité.

---

9. Les résultats qui doivent être obtenus seront illustrés à la fin du projet.

## 12.8 SOLUTIONS ET PROGRAMMES

Les programmes MATLAB sont organisés en deux répertoires :

- *QP* pour les questions préliminaires (exercices 12.1 à 12.5),
- *QNS* pour la résolution des équations de Navier-Stokes (exercice 12.6).

Un troisième répertoire, *INTERFACE*, contient les programmes réunis, avec une interface graphique qui permet de naviguer facilement entre les différentes questions (un cas de calcul Navier-Stokes supplémentaire est traité également). Pour lancer l’interface graphique, il faut simplement aller dans le sous-répertoire *Tutorial*, lancer le programme *Main* et se laisser guider.

### Solution de l’exercice 12.1 : résolution d’un système linéaire à matrice tridiagonale et périodique

La fonction MATLAB *trid\_per\_c2D.m* résout simultanément  $m$  systèmes linéaires à matrice tridiagonale, périodique, de dimension  $n$ . Les commentaires permettent de suivre facilement les étapes de l’algorithme 12.6. L’utilisation de la mémoire a été optimisée en utilisant un nombre minimum de tableaux pour les coefficients intermédiaires. Remarquer la programmation vectorielle de l’algorithme. Rappelons que cette fonction est appelée et testée dans le programme *test\_trid.m*.

### Solution de l’exercice 12.2 : Méthode explicite pour l’équation instationnaire de la chaleur

Le programme *Qexp\_lap.m* utilisant la méthode explicite est tout à fait élémentaire et se passe de tout commentaire. Les fonctions utilisées par ce programme sont les suivantes :

- *calc\_lap.m* : calcul du laplacien  $\Delta u$ ;
- *fsource.m* : calcul du terme source  $f$  ;
- *fexact.m* : calcul de la solution exacte ;
- *norme\_L2.m* : calcul de la norme  $\|u\|_2$  ;
- *visu\_isos.m* : visualisation des iso-contours de la solution numérique et de la solution exacte.

### Solution de l’exercice 12.3 : Méthode implicite pour l’équation instationnaire de la chaleur

Le programme *Qimp\_lap.m* implémente la méthode implicite avec optimisation de la résolution des systèmes à matrice tridiagonale et périodique. Pour cette dernière opération, il suffisait d’observer dans la fonction *trid\_per\_c2D* (fichier *trid\_per\_c2D.m*) que tous les calculs qui ne font pas intervenir le second membre *f\_i* peuvent être faits une seule fois, en dehors de la boucle en temps. C’est le rôle de la fonction *ADI\_init* qui retourne les vecteurs *am\_i*, *api*, *a1ph*, *xS2* qui

seront ensuite utilisés dans la fonction **ADI\_step** pour trouver la solution finale des systèmes linéaires pour chaque second membre **fi** donné. Techniquelement, cette optimisation revient à séparer en deux parties la fonction originale **trid\_per\_c2D**.

Le programme *Qimp\_lap.m* garde la même structure que celui écrit pour la méthode explicite, à l'exception de la partie de calcul effectif de la solution.

### **Solution de l'exercice 12.4 : Méthode implicite pour l'équation de convection-diffusion non-linéaire**

Le programme *Qimp\_lap\_nonl.m* est similaire au programme précédent (*Qimp\_lap.m*). La seule différence est le calcul du terme  $\mathcal{H}$  à chaque pas de temps. Il s'effectue à l'intérieur de la boucle en temps par l'appel

```
hc =calc_hc(Lx,Ly,xx,yy,u);
```

où la fonction correspondante est écrite dans le fichier *calc\_hc.m*. Bien entendu, le terme source  $f$ , calculé dans *fsource\_nonl.m* a été modifié pour correspondre au problème non-linéaire.

### **Solution de l'exercice 12.5 : Résolution de l'équation stationnaire de la chaleur en utilisant les FFT**

L'algorithme 12.4 est implémenté dans le programme *Qfft\_lap.m*. L'étape de résolution du système tridiagonal est optimisée par séparation en deux fonctions **Phi\_init** et **Phi\_step**; la différence par rapport à l'optimisation de la méthode ADI est que les coefficients des matrices changent d'un système à l'autre (à cause du nombre d'onde), ce qui nécessite leur stockage dans des matrices.

### **Solution de l'exercice 12.6 : Résolution des équations de Navier-Stokes**

Le programme principal *QNS.m* permet de choisir parmi les cas de calcul suggérés. Les commentaires permettent d'identifier les différentes étapes de l'algorithme de résolution. Les principales fonctions utilisées par ce programme sont celles développées dans les questions préliminaires (**ADI\_init**, **ADI\_step**, **Phi\_init**, **Phi\_step**). Les fonctions spécifiques au calcul Navier-Stokes sont les suivantes :

- *init\_KH.m* construit la condition initiale pour l'instabilité de Kelvin-Helmholtz (jet plan) ;
- *init\_vortex.m* construit la condition initiale pour un tourbillon ; les dipôles de vorticité seront obtenus par la superposition de plusieurs tourbillons ;
- *visu\_vort.m* permet de visualiser le champ de vorticité (iso-contours) ;
- *visu\_sca.m* visualise les contours du scalaire (traceur) passif ;
- *affiche\_div.m* affiche la divergence du champ de vitesse pour vérifier le bon déroulement du calcul (elle doit rester proche de zéro-machine, c'est-à-dire  $10^{-15}$ ).

Les figures suivantes illustrent l'évolution des écoulements correspondant aux quatre cas simulés numériquement. Pour les deux premiers cas (figures 12.5 et 12.6), les tourbillons de Kelvin-Helmholtz se forment progressivement, avec une distribution spatiale dictée par la condition initiale (valeur de  $\lambda_x/L_x$ ). Si on veut faire un parallèle avec l'écoulement réel, cette simulation numérique (avec des conditions de périodicité) reproduit les phénomènes observables dans une fenêtre de visualisation fixe (la boîte de calcul) qui se déplace en aval avec la vitesse caractéristique de l'écoulement.

Le troisième cas de calcul (figure 12.7) montre l'évolution d'un dipôle de vorticité qui se propage par lui-même suivant l'axe horizontal. Il est intéressant d'observer l'empreinte laissé par le dipôle (grâce à la vitesse induite par la vorticité qu'il contient) dans le scalaire passif qui était initialement au repos. Ce type de structure se rencontre dans plusieurs types d'écoulements fluides, étudiés en océanographie, météorologie ou combustion.

Le dernier cas de calcul (figure 12.8) montre l'interaction frontale entre deux dipôles de vorticité de même intensité. Le résultat est le création de deux nouveaux dipôles (c'est un phénomène de *changement de partenaires*) qui se propagent perpendiculairement à la direction initiale de déplacement. Remarquons que, pour ces deux derniers calculs, les dipôles ne quittent jamais la boîte de calcul, car les conditions de périodicité imposées les font revenir par la frontière opposée (on peut d'ailleurs continuer la simulation pour s'en convaincre).

D'autres cas de calcul peuvent être imaginés suivant le même modèle (prendre, par exemple, une condition initiale avec quatre dipôles de vorticité).

## BIBLIOGRAPHIE

- [Hirsh, 1988] C. HIRSCH : *Numerical computation of internal and external flows*, John Wiley & Sons, 1988.
- [Joly, 2004] P. JOLY : *Analyse numérique matricielle*, Cassini, Paris, 2004.
- [Lesieur, 1994] M. LESIEUR : *La Turbulence*, Presses Universitaires de Grenoble, 1994.
- [Orlandi, 1999] P. ORLANDI : *Fluid Flow Phenomena*, Kluwer Academic Publishers, 1999.

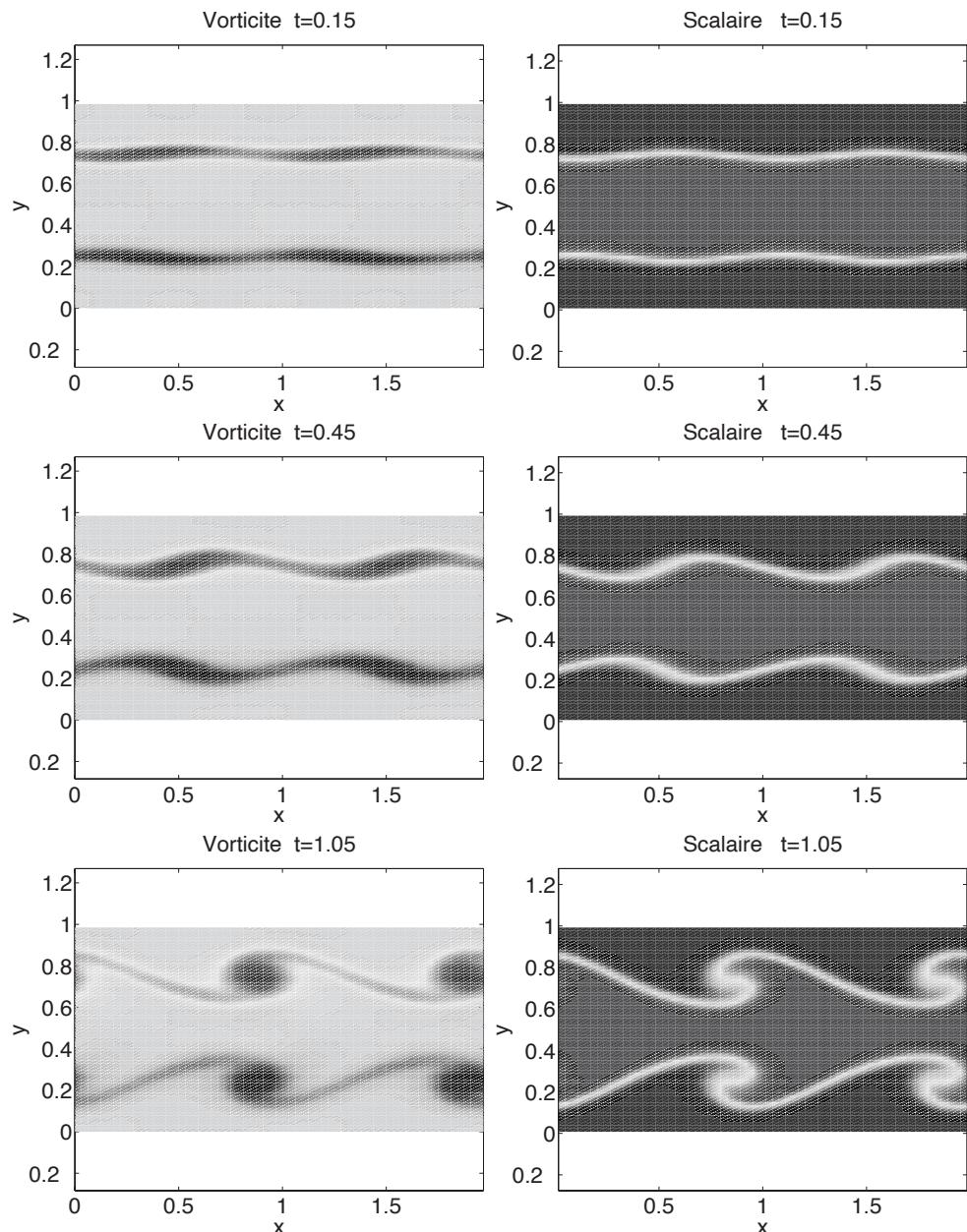


Figure 12.5 Cas de calcul 1. Développement de l'instabilité de Kelvin-Helmholtz pour  $\lambda_x/L_x = 0.5$ .

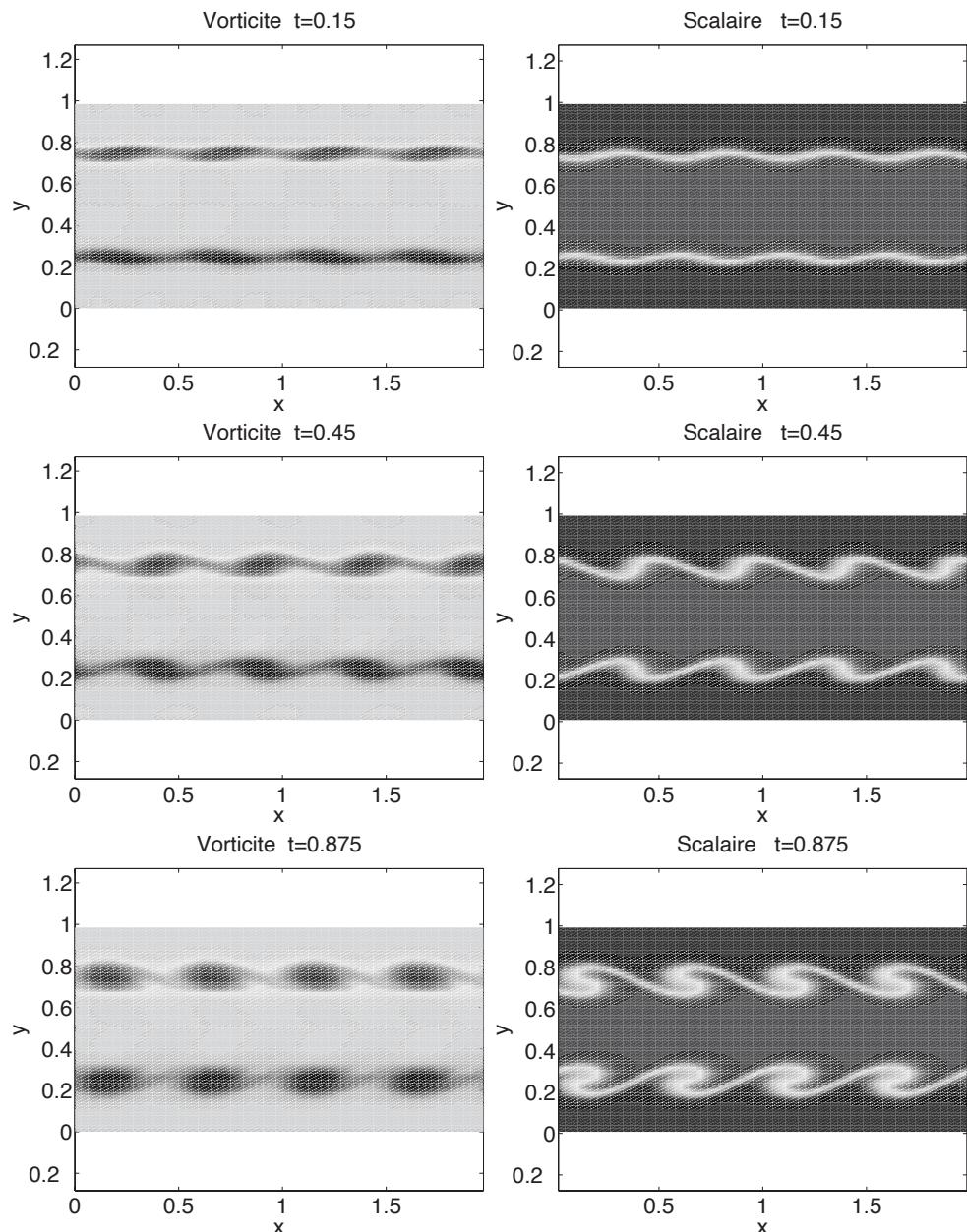
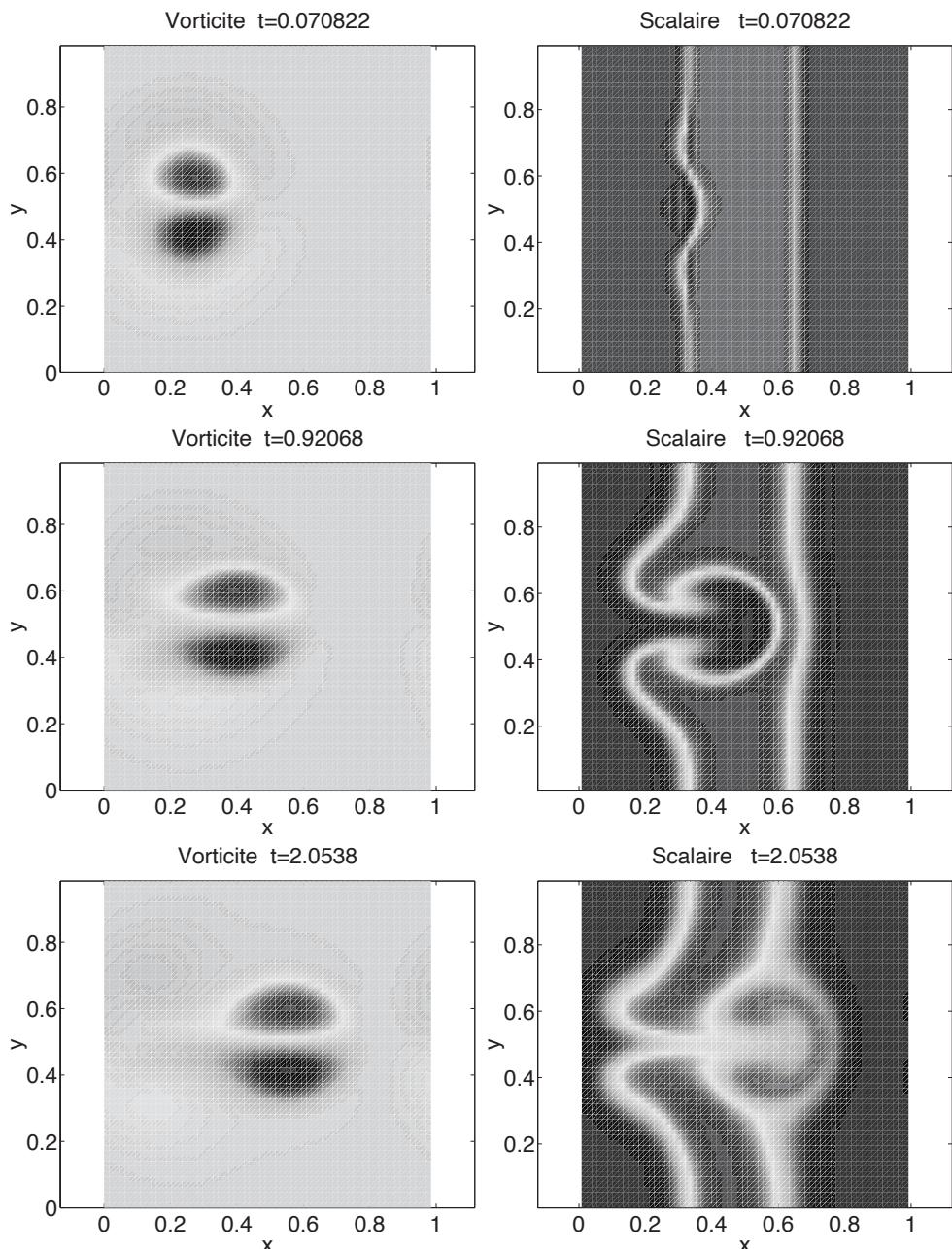
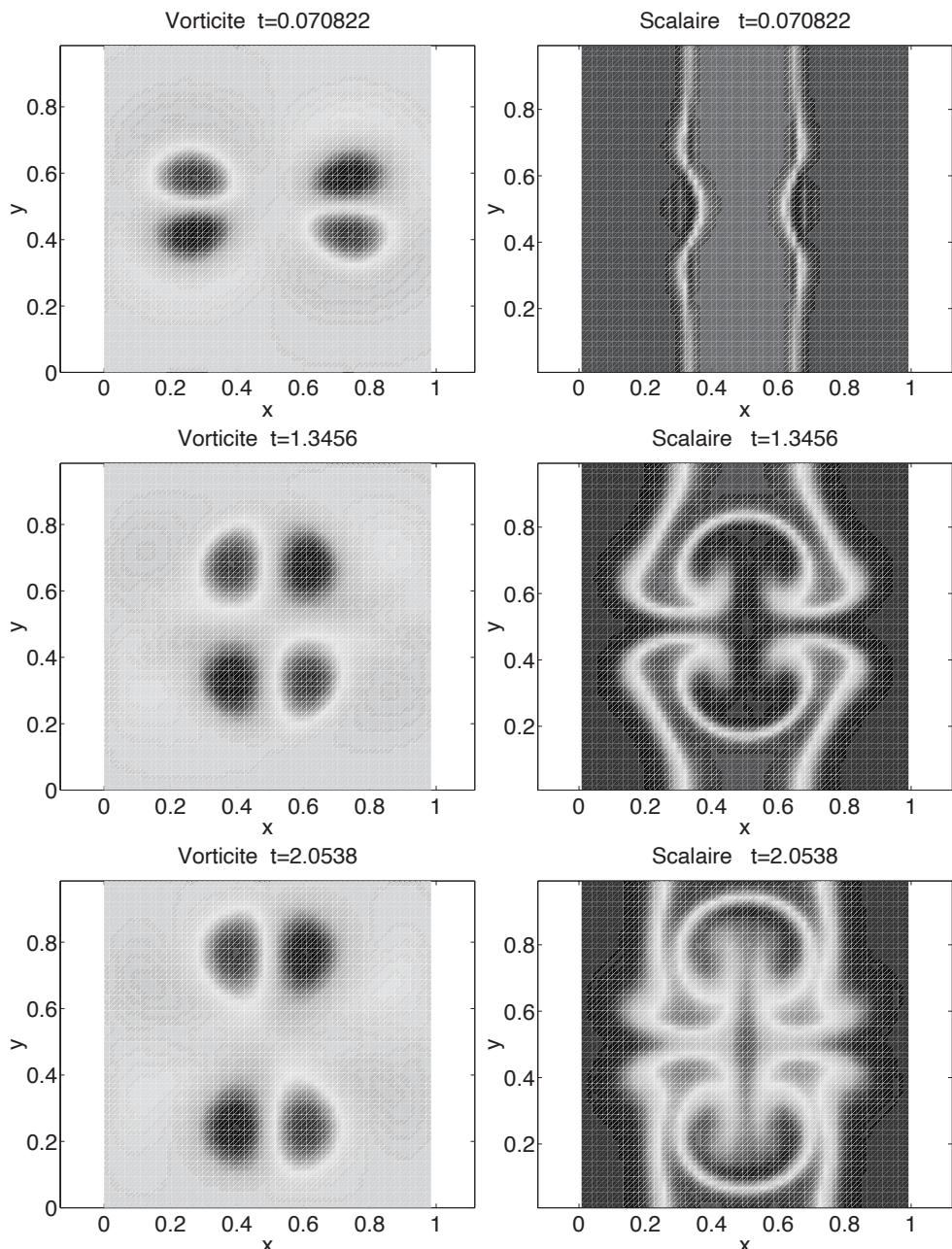


Figure 12.6 Cas de calcul 2. Développement de l'instabilité de Kelvin-Helmholtz pour  $\lambda_x/L_x = 0.25$ .



**Figure 12.7** Cas de calcul 3. Evolution d'un dipôle de vorticité.



**Figure 12.8** Cas de calcul 4. Interaction de deux dipôles de vorticité .