
Méthodes itératives pour les fluides incompressibles

Sofiane Hadji* — Gouri Dhatt**

*Université de Compiègne (UTC), Génie des systèmes mécanique
Division M.N.M, Compiègne, France

**INSA de Rouen, France
(UTC également)

RÉSUMÉ. Cet article est consacré à une présentation détaillée des méthodes itératives Gradient conjugué (CG), Biconjugate gradient (BICG), conjugate gradient squared (CGS), Biconjugate gradient stabilized (BICGSTAB), transpose-free quasi minimal residual (TFQMR), full orthogonal method (FOM) et Generalized minimal residual (GMRES) adaptées à la résolution des systèmes éléments finis des équations de Navier-Stokes dans le cadre des fluides incompressibles. Afin d'accélérer le taux de convergence de ces méthodes, le préconditionnement basé sur la factorisation incomplète de Gauss (ILU) est couplé à celles-ci. Dans ce type de préconditionnement, l'accent est mis sur la nécessité de la renumérotation des variables, et enfin, l'importance d'un critère d'arrêt variable des méthodes itératives pour chaque itérations de Newton-Raphson est mis en évidence.

ABSTRACT. This paper presents an overview of different iterative methods Conjugate gradient (CG), Biconjugate gradient (BICG), conjugate gradient squared (CGS), Biconjugate gradient stabilized (BICGSTAB), transpose-free quasi minimal residual (TFQMR), full orthogonal method (FOM) et Generalized minimal residual (GMRES) for solving finite element systems of Navier-Stokes incompressible flows. To accelerate convergence of those methods, preconditionner based on incomplete Gauss factorisation (ILU) is used. The accent is put on the necessity to renumber the unknowns to guaranty the convergence of the iterative methods. The importance of a variable stop criterion of iterative methods for each Newton-Raphson step is underlined.

MOTS-CLÉS : modélisation, éléments finis, méthodes itératives, préconditionnement, fluides incompressibles.

KEY WORDS : modelling, finite element, iterative methods, preconditionner, incompressible flows.

1. Introduction

La méthode des éléments finis est considérée comme un outil fiable pour résoudre des problèmes industriels complexes découlant de la mécanique des fluides compressibles ou incompressibles. L'expérience acquise sur les modèles numériques en différences et volumes finis a contribué d'une façon significative au développement actuel de modèles éléments finis. La tendance actuelle est de modéliser des problèmes 2D et 3D utilisant les variables primitives (vitesses-pression) couplées à des méthodes de résolution itératives conduisant à plus de 50 000 variables.

Dans cette étude, on s'intéresse à l'évaluation des différentes méthodes itératives couplées à différents types de Préconditionnement pour la résolution des équations de Navier-Stokes incompressibles. Pour ce type d'équations, il est courant d'utiliser une formulation de type Galerkin (champ de vitesses plus riche que le champ de pression) et des méthodes robustes de type Newton couplées à une méthode de résolution directe. Néanmoins, la résolution de grands problèmes non linéaires par de tels solveurs devient très coûteuse en temps CPU et présente une limitation en termes de stockage. Il est donc indiqué d'utiliser les méthodes itératives, en particulier pour les problèmes de grandes tailles. Ces méthodes s'accompagnent bien sûr d'un choix adéquat pour le type de Préconditionnement utilisé, de la renumérotation des variables pour assurer la convergence, et enfin, du choix d'un 'bon' critère d'arrêt pour accélérer la convergence.

Dans la section suivante, nous présentons les modèles mathématiques et numériques des équations de Navier-Stokes. Une section est consacrée à la présentation des méthodes de linéarisation. Une autre section est consacrée à la description des différentes méthodes itératives suivie d'une section sur le Préconditionnement utilisé. Une autre section donne les différents types de renumérotation, et enfin une dernière section concerne le choix du critère d'arrêt. Des détails sur l'expérimentation numérique sont entrepris afin d'évaluer l'efficacité relative des différentes méthodes pour des nombres de Reynolds supérieurs à 400.

2. Modèle mathématique

Les problèmes de fluides sont caractérisés par les relations de la conservation de la quantité de mouvement et de la conservation de la masse.

$$\rho \frac{\partial \bar{u}}{\partial t} + \text{Div.}[\rho \bar{u} \otimes \bar{u} + p\mathbf{I}] - \mu \nabla^2 \bar{u} - \beta \nabla(\nabla \cdot \bar{u}) - \bar{f}_v = 0$$

[2.1]

$$\nabla \cdot \bar{u} + \alpha \nabla^2 p = 0$$

avec les conditions aux limites :

Paroi : condition d'adhérence $\bar{u} = 0$

Entrée : $\bar{u} = \bar{u}$

Sortie : $p = \bar{p}$, $\frac{\partial \bar{u}}{\partial \bar{n}} = 0$

\bar{u} est le vecteur vitesse, p la pression, ρ la densité, et μ la viscosité du fluide. α et β sont deux coefficients de stabilisation dépendant de la maille et du nombre de Peclet. Pour les écoulements stationnaires, le premier terme temporel de l'équation [2.1] est négligé. Les écoulements de Stokes sont les écoulements stationnaires où les termes non linéaires de convection sont négligés. Les relations classiques de Navier-Stokes sont définies par $\alpha = \beta = 0$.

REMARQUE. – Les expressions de α et β sont les suivantes : [STO].

$$\alpha = \frac{h}{2} \sqrt{\frac{\varepsilon}{\mu}} \quad \beta = \frac{h}{6} \sqrt{\frac{\mu}{\varepsilon}} \quad \bar{\varepsilon} = \frac{\varepsilon \mu}{h^2}$$

h est la taille de la maille. Dans notre cas : $\bar{\varepsilon} = 1$

2.1. Ecriture variationnelle

La forme variationnelle associée aux équations [2.1] s'écrit :

$$W = W_{N-S} + W_{inc}$$

$$W = \underbrace{\int_v \bar{\psi}_u \cdot \left[\rho \frac{\partial \bar{u}}{\partial t} + \text{Div}(\rho \bar{u} \otimes \bar{u} + p \mathbf{I}) - \mu \nabla^2 \bar{u} - \bar{f} \right] dv}_{W_{N-S}} + \underbrace{\int_v \psi_p \nabla \cdot \bar{u} dv}_{W_{Inc}} = 0 \quad [2.2]$$

L'écriture faible prend alors l'expression suivante :

$$W = \int_v \bar{\psi}_u \cdot \left[\rho \frac{\partial \bar{u}}{\partial t} + \text{Div}(\rho \bar{u} \otimes \bar{u} + p \mathbf{I}) + \text{Tr}(\nabla \bar{\psi}_u \cdot \nabla u) - \nabla \cdot \bar{\psi}_u \cdot p - \bar{f} \right] dv + \int_v \psi_p \nabla \cdot \bar{u} dv - \int_{S_f} \text{Tr}(\bar{\psi} \cdot \nabla u) ds + \int_{S_f} \bar{\psi}_u \cdot p \bar{n} ds = 0 \quad \text{avec : } \bar{u} = \bar{u}_s \text{ sur } S_u$$

[2.3]

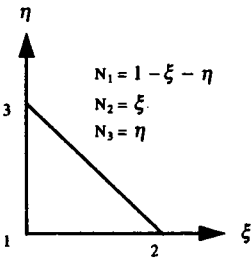
\bar{n} est la normale à la frontière dirigée vers l'extérieur de la frontière S .

REMARQUE. – Le nombre de Reynolds est défini par :

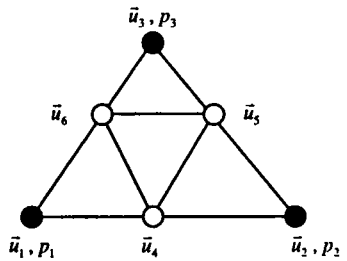
$$Re = \frac{\rho |u| L}{\mu} \text{ où } L \text{ est la longueur caractéristique}$$

2.2. Type d'élément

L'élément utilisé dans cette étude est l'élément P2-P1. C'est un triangle à six nœuds composé de quatre sous-triangles à trois nœuds (fig. 1), où l'approximation des vitesses est linéaire dans chaque sous-triangle, et l'approximation de la pression est linéaire dans le triangle principal.



Triangle de Référence pour les 4 sous-triangles



Triangle à six nœuds composé de 4 sous-triangles

Figure 2.1. Elément P2-P1

2.3. Structure matricielle générale

La forme variationnelle [2.3] discrétisée par éléments finis peut s'écrire :

$$W = \langle \psi \rangle ([M]\{\dot{U}\} + [K]\{U\} - \{F\}) = 0 \quad \forall \psi \tag{2.4}$$

$$\text{Le système à résoudre est alors : } [M]\{\dot{U}\} + [K]\{U\} = \{F\} \tag{2.5}$$

où {U} est le vecteur des variables globales, [M] et [K] sont la matrice masse globale et la matrice de rigidité globale respectivement. {F} est le vecteur global des sollicitations extérieures.

Utilisant une linéarisation de type Newton ou asymptotique et une discrétisation temporelle de type Euler implicite, le système [2.5] devient alors :

$$[A]\{x\} = \{b\} \quad [2.6]$$

où $\langle x \rangle = \langle \Delta u \ \Delta v \ \Delta p \rangle$ et $\{b\}$ le vecteur résidu.

3. Méthodes de linéarisation

Les relations aux dérivées partielles associées aux équations de Navier-Stokes peuvent s'écrire sous la forme :

$$Q(U, U) + L(U) - F_{\text{ext}} = 0 \quad [3.1]$$

avec

$$L(U) = \begin{Bmatrix} \frac{\partial p}{\partial x} - \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \\ \frac{\partial p}{\partial y} - \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \\ - \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right) \end{Bmatrix}, \quad Q(U, U) = \begin{Bmatrix} \rho \frac{\partial}{\partial x}(uu) + \rho \frac{\partial}{\partial y}(uv) \\ \rho \frac{\partial}{\partial x}(uv) + \rho \frac{\partial}{\partial y}(vv) \\ 0 \end{Bmatrix}$$

$$F_{\text{ext}} = \begin{Bmatrix} \rho f_x \\ \rho f_y \\ 0 \end{Bmatrix}$$

La forme variationnelle associée à l'équation [3.1] s'écrit :

$$W_h = W_{hL} + W_{hQ} + W_{h\text{ext}} = 0 \quad \forall \delta u_n, \delta v_n, \delta p_n$$

$$\text{ou :} \quad \{R_L(u_n)\} + \{R_Q(u_n, u_n)\} - \{F_n\} = 0 \quad [3.2]$$

avec :

$$W_{hL} = \langle \delta u_n \rangle \{R_L(u_n)\} \quad W_{hQ} = \langle \delta u_n \rangle \{R_Q(u_n, u_n)\} \quad W_{h\text{ext}} = - \langle \delta u_n \rangle \{F_n\}$$

3.1. Méthodes de type Newton

Pour résoudre le système d'équations non-linéaire [3.2], une étape de linéarisation est nécessaire. En effet, le problème non-linéaire est transformé en une série de problèmes linéaires de manière à ce que le sous-espace des solutions linéaires converge vers la solution non-linéaire désirée. Le succès d'une telle stratégie itérative dépend de la construction du problème linéarisé. Il est bien connu que les méthodes de type Newton ont de très bonnes propriétés de convergence si la solution initiale est proche de la solution exacte.

Les principales étapes d'une linéarisation de type Newton sont les suivantes :

- La solution $\{U_0\}$ est connue pour un niveau de chargement $\{F_0\}$ qui correspond à un nombre de Reynolds donné

$$\{R_0\} = \{R_L(U_0)\} + \{R_Q(U_0, U_0)\} - \{F_0\} = \{0\} \tag{3.3}$$

- Choisir un niveau de chargement ou un nombre de Reynolds

$$\{F\} = \{F_0\} + a \{F_1\} \tag{3.4}$$

on peut fixer l'amplitude 'a' ou la calculer en utilisant la norme du premier vecteur solution.

- Obtenir la solution en résolvant un certain nombre de problèmes linéaires. L'espace solution est constitué par $\{U^{(1)}\}, \{U^{(2)}\}, \dots, \{U^{(m)}\}$ obtenues par une série de résolution de systèmes linéaires.

On cherche alors la solution $\{U\}$ telle que :

$$\{U\} = \{U_0\} + a_1 \{U^{(1)}\} + a_2 \{U^{(2)}\} + \dots + a_m \{U^{(m)}\} \tag{3.5}$$

$$\{R(U)\} = \left\{ R(U_0 + \sum_{i=1}^m a_i U^{(i)}) \right\} = 0 \tag{3.6}$$

les vecteurs $\{U^{(i)}\}$ sont solutions des problèmes linéaires :

$$[K_T] \{U^{(i)}\} + \{\bar{R}^{(i)}\} = 0 \tag{3.7}$$

conduisant à :

$$\{R(U)\} = \sum ([K_T] \{U^{(i)}\} + \{\bar{R}^{(i)}\}) = \{0\} \tag{3.8}$$

La matrice $[K_T]$ est obtenue par une linéarisation de type Newton pour un $\{U\}$ donné. Le vecteur $\{\bar{R}\}$ peut correspondre à l'expression de $\{R\}$ ou à une représentation asymptotique. Les coefficients a_i représentent soit des paramètres de relaxation ou bien un développement asymptotique. La relation [3.8] doit être satisfaite de façon à obtenir la solution désirée.

Le choix de $[K_T]$ et de $\{\bar{R}\}$ conduit à une série de méthodes présentées dans les sections suivantes.

3.1.1. Méthode de Newton-Raphson

La matrice tangente $[K_T]$ est calculée pour une estimation de la solution $\{U\}$. Ceci est obtenu en discrétisant l'expression de ΔW linéaire en $\{\Delta U\}$:

$$W(U + \Delta U) = W(U) + \Delta W + \dots \quad [3.9]$$

$$\Delta W_h = \langle \delta u_n \rangle [K_T] \{\Delta U\}$$

Si on choisit $a_i = 1$, alors :

$$\{U\} = \{U_0\} + \{U^{(1)}\} + \{U^{(2)}\} + \dots \quad [3.10]$$

$$\left[K_T \left(U_0 + \sum_{j=1}^{i-1} U^{(j)} \right) \right] \{U^i\} = -\{\bar{R}\} \quad [3.11]$$

$$\text{avec : } \{\bar{R}\} = \left\{ R \left(U_0 + \sum_{j=1}^{i-1} U^j \right) \right\}$$

Les vecteurs $\{U^{(1)}\}$, $\{U^{(2)}\}$ représentent les vecteurs incrémentaux $\{\Delta U\}$ pour chaque itération.

L'incrément 'a' de l'équation [3.4] peut être choisi comme suit :

- L'utilisateur définit la valeur de a
- L'utilisateur définit la valeur de la norme s_0 telle que :

$$\| a \cdot U^{(1)} \| = s_0 \quad [3.12]$$

- Ajuster la valeur de 'a' à chaque niveau d'itération pour une norme s_0 donnée basée sur la notion de longueur d'arc [DHA 95].

OP₁ :

$$\| a \cdot U^{(1)} \| = s_0 \quad [3.13]$$

OP₂ :

Orthogonalisation :

$$(\Delta a U^{(1)} + U^{(2)}) \cdot U^{(1)} = 0 \quad a = a + \Delta a$$

$$\overline{OP}_2 = OP_1 + (\Delta a U^{(1)} + U^{(2)})$$

[3.14]

Ajustement :

$$OP_2 = (\overline{OP}_2 / |\overline{OP}_2|) s_0$$

OP₃ :

Orthogonalisation :

$$(\Delta a U^{(1)} + U^{(3)}) \cdot U^{(2)} = 0 \quad a = a + \Delta a$$

$$\overline{OP}_3 = OP_1 + (\Delta a U^{(1)} + U^{(3)})$$

[3.15]

Ajustement :

$$OP_3 = (\overline{OP}_3 / |\overline{OP}_3|) s_0$$

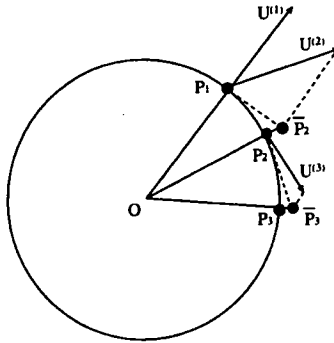


Figure 3.1. 'a' par la longueur d'arc

L'algorithme de Newton-Raphson est donné dans la figure 3.2.

- Choix de la norme de convergence ϵ
- Solution $\{U_0\}$ pour un niveau de chargement F_0
- Choix de la norme de l'incrément 'a'

ITERATION : ITER = 1, 2,....., NITER

Calculer $[K_T]$

Calculer $\{R\}$

Résoudre : $[K_T]\{\Delta U\} = \{R\}$

Pour un 'a' fixé ou par (12) :

Solution : 1 - Pour un a fixé :
 $\{U\} = \{U\} + \{\Delta U\}$
 2 - Utilisant (12), $\{\Delta U\}$
 $a = a + \Delta a$
 $U = O P^{ITER}$

Calculer la norme relative :

$$\begin{aligned} & \| \{\Delta U\} \| / \| \{U\} \| \\ & \| \{R\} \| / \| a \{F_1\} \| \end{aligned}$$

Convergence :

$$\text{Norme} < \epsilon$$

FIN DE BOUCLE ITER

Figure 3.2. Algorithme de Newton-Raphson

3.1.2. Méthode de Newton-Raphson modifiée

Cette méthode est identique à celle de Newton-Raphson excepté que la matrice tangente est calculée puis factorisée une seule fois au début des itérations ou maintenue constante pour un nombre donnée de pas de niveau de chargement. La factorisation de la matrice $[K_T]$ est donc effectuée occasionnellement ce qui rend la méthode relativement performante. Si le comportement de convergence de la méthode n'est plus satisfaisant, on recalcule $[K_T]$ et on relance la procédure pour un certain nombre d'itérations jusqu'à obtenir la solution désirée.

3.2. Méthode de Newton-Asymptotique

Comme présentée au début de la section (équations [3.6]), l'aspect essentiel de toute méthode de linéarisation est le choix de la matrice tangente ou matrice Jacobienne $[K_T]$ et le choix du vecteur résidu $\{\bar{R}\}$ pour chaque itération. Dans la méthode asymptotique [COC 94], [DAM 90], [HAD 97], nous avons :

- La matrice tangente est calculée une seule fois au début des itérations ou à chaque cycle de chargement. Cette étape est donc identique à celle de la méthode de Newton-Raphson modifiée.

- L'évaluation du vecteur résidu est obtenue d'une manière spéciale qui diffère de celles utilisées par les méthodes de Newton-Raphson ou Newton-Raphson modifiée.

Un problème non-linéaire est représenté par une suite de problème linéaire. La solution est écrite sous forme de série en fonction d'un paramètre de chargement 'a' :

$$\{F\} = \{F_0\} + a \{F_1\} \tag{3.16}$$

$$\{U\} = \{U_0\} + a \{U^{(1)}\} + a^2 \{U^{(2)}\} + a^3 \{U^{(3)}\} + \dots + a^m \{U^{(m)}\}$$

L'expression du vecteur résidu devient :

$$\{R(U)\} = \{R_0\} + a \{R^{(1)}\} + a^2 \{R^{(2)}\} + a^3 \{R^{(3)}\} + \dots + a^m \{R^{(m)}\} \tag{3.17}$$

qui conduit à un certain nombre de problèmes linéaires associé à chaque puissance de 'a' :

$$\begin{aligned} \{R_0\} &= \{0\} \\ \{R^{(1)}\} &= [K_{T0}] \{U^{(1)}\} - \{F_1\} = 0 \\ \{R^{(2)}\} &= [K_{T0}] \{U^{(2)}\} + \{\bar{R}^{(2)}\} = 0 \\ &\vdots \\ \{R^{(m)}\} &= [K_{T0}] \{U^{(m)}\} + \{\bar{R}^{(m)}\} = 0 \end{aligned} \tag{3.18}$$

La caractéristique typique de la méthode asymptotique est liée à l'évaluation du vecteur résidu $\{\bar{R}^{(i)}\}$. Le vecteur résidu dans la méthode de Newton correspond à l'expression de W pour une solution estimée $\{U\}$. Pour la méthode asymptotique, le vecteur résidu correspond à l'expression des termes liés à a, a^2, \dots, a^m . Prenons une équation algébrique non-linéaire à un seul degré de liberté pour expliquer la méthode.

Soit l'équation non-linéaire quadratique définie par :

$$L(U) + Q(U, U) - F = 0$$

ou

$$k_0 U + k_1 U^2 - (F_0 + a F_1) = 0 \quad [3.19]$$

supposons que la solution u_0 pour un F_0 donné est connue :

$$R_0 = k_0 U_0 + k_1 U_0^2 - F_0 = 0$$

Pour la méthode de Newton-Raphson modifiée, la série de problèmes linéaires est (U_m correspond à $U^{(m)}$) :

$$K_{T0} = (k_0 + 2 k_1 U_0) \quad [3.20]$$

$$R_{1L} = K_{T0} U_1 + (k_0 U_0 + k_1 U_0^2 - F_0 - a F_1) = K_{T0} U_1 - a F_1 = 0 (\Delta U = U_1)$$

$$\underbrace{\hspace{10em}}_{R_1} \quad \underbrace{\hspace{2em}}_{R_1}$$

$$R_{2L} = K_{T0} U_2 + (k_0(U_0 + U_1) + k_1 (U_0 + U_1)^2 - F_0 - a F_1) = K_{T0} U_2 + k_1 U_1^2 = 0$$

$$\underbrace{\hspace{10em}}_{R_2} \quad \underbrace{\hspace{2em}}_{R_2}$$

$$U = U_0 + U_1 + U_2$$

$$R_{3L} = K_{T0} U_3 + (k U + k U^2 - F_0 - a F_1) = K_{T0} U_3 + k_1 (2U_1 U_2 + U_1^2) = 0$$

$$\underbrace{\hspace{10em}}_{R_3} \quad \underbrace{\hspace{10em}}_{R_3}$$

[3.21]

$$U = U_0 + \sum_{j=1,3} U_j$$

$$R_{mL} = K_{T0} U_m + (kU + kU^2 - F_0 - aF_1) = K_{T0} U_m + k_1 \underbrace{\left(2 \sum_1^{m-2} U_j \sum_{j+1}^{m-1} U_j + U_{m-1}^2 \right)}_{R_m} = 0$$

$$U = U_0 + \sum_{j=1,m} U_j$$

Pour la méthode Newton-Asymptotique (U_m défini $U^{(m)}$), nous avons :

$$U = U_0 + a U^{(1)} + a^2 U^{(2)} + \dots + a^m U^{(m)} \tag{3.22}$$

$$F = F_0 + a F_1$$

la série de problèmes linéaires est :

$$R = R_0 + a R_1 + a^2 R_2 + \dots + a^m R_m \tag{3.23}$$

avec :

$$R_0 = k_0 U_0 + k_1 U_0^2 - F_0 = 0$$

$$R_{1L} = K_{T0} U_1 + \dots = 0, \quad \bar{R}_1 = -F_1 \quad (\text{identique à N-R Modifiée})$$

$$R_{2L} = K_{T0} U_2 + \bar{R}_2 = 0, \quad \bar{R}_2 = k_1 U_1^2 \quad (\text{identique à N-R Modifiée})$$

$$R_{3L} = K_{T0} U_3 + \bar{R}_3 = 0, \quad \bar{R}_3 = 2 k_1 U_1 U_2 \quad (\text{différent de N-R Modifiée})$$

⋮
⋮
⋮

$$R_{mL} = K_{T0} U_m + \bar{R}_m = 0, \quad \bar{R}_m = k_1 \sum_{j=1}^{m-1} U_j U_{m-1-j}$$

4. Méthodes itératives pour les systèmes linéaires

Nous avons montré dans la section précédente que l'étape de linéarisation conduisait à la résolution d'un système linéaire de la forme $[A]\{x\} = \{b\}$. Lorsque le problème est de grande taille, nous pouvons exploiter la structure particulière de la matrice en utilisant par exemple un stockage en ligne de ciel couplé à la méthode de résolution directe de Gauss. Néanmoins, dans la majorité des cas, spécialement en 3D, l'approche directe est trop coûteuse en espace mémoire et en temps CPU. La seule alternative est donc d'utiliser les méthodes itératives.

4.1. Présentation générale

La linéarisation d'un problème non linéaire conduit à un système de relations linéaires non symétrique suivant :

$$[A]\{x\} = \{b\} \text{ ou } \{r\} = \{b\} - [A]\{x\} = \{0\} \quad [4.1]$$

où $[A]$ est une matrice non symétrique de taille $(n \times n)$

$\{b\}$ est le vecteur des sollicitations

$\{x\}$ est le vecteur solution

Pour des systèmes de grandes tailles (20.000 ou plus), nous utilisons souvent les méthodes itératives pour obtenir la solution $\{x\}$ de [4.1]. La démarche de résolution par les méthodes itératives contient les aspects suivants :

- Nous transformons le système [4.1] par un autre système équivalent par le choix d'un préconditionnement (voir section 5). Dans cette section, le système à résoudre est défini par [4.1].

- L'espace solution K_m est défini par sa base $\{v_1\}, \{v_2\}, \{v_3\}, \dots, \{v_m\}$ tel que :

$$[V]_m = [\{v_1\}, \{v_2\}, \{v_3\}, \dots, \{v_m\}] \text{ base de } K_m \quad [4.2]$$

La projection de la solution $\{x\}$ dans cet espace de dimension m s'écrit :

$$\{x\} \xrightarrow{K_m} \{x_m\} = \{x_1\} + \sum_{i=1}^m \{v_i\} y_i = \{x_1\} + [V]_m \{y\}_m$$

ou encore

$$\{x_m\} = \{x_1\} + \{z_m\} \quad [4.3]$$

$\{x_1\}$ est la solution initiale

$\{x_m\}$ est l'approximation de la solution

$\{z_m\}$ est la projection de $\{x\}$ dans l'espace K_m

Le vecteur résidu associé à une solution approximative est :

$\{r_1\} = \{b\} - [A]\{x_1\}$: résidu initial

$\{r_m\} = \{b\} - [A]\{x_m\}$: résidu de la solution obtenue

ou

$$\{r_m\} = \{r_1\} - \sum_{i=1}^m [A]\{v_i\}y_i = \{r_1\} - [A][V]_m \{y\}_m \quad [4.4]$$

L'espace test L_m est défini par sa base $\{w_1\}, \{w_2\}, \{w_3\}, \dots, \{w_m\}$

$$[W]_{m \times m} = [\{w_1\}, \{w_2\}, \{w_3\}, \dots, \{w_m\}] \text{ base de } L_m \quad [4.5]$$

La dimension de L_m est égale à celle de K_m .

Les coefficients y_i de la solution sont obtenus en imposant la projection du résidu dans L_m nulle. Il est courant d'appeler cette technique comme critère d'orthogonalité, critère du résidu pondéré, critère de recherche optimale, etc. Nous cherchons y_i tel que le résidu $\{r_m\}$ (éq. [4.4]) soit nul dans l'espace L_m :

$$\langle w_i | \{r_i\} \rangle = 0 \quad i = 1, 2, \dots, m.$$

Sous forme matricielle, cette équation s'écrit :

$$[W]_m^T (\{r_1\} - [A][V]_m \{y\}_m) = 0 \quad [4.6]$$

Soit :

$$[\bar{A}] \{y\}_m = \{\bar{r}\}_m \quad [4.7]$$

où

$$[\bar{A}]_{m \times m} = [W]_m^T [A][V]_m \text{ est la projection de } [A] \text{ dans les sous-espaces } (K_m, L_m)$$

$$\{\bar{r}\}_m = [W]_m^T \{r_1\} \text{ est la projection du résidu } \{r_1\} \text{ dans } L_m$$

La solution est obtenue par la résolution de [3.7].

$$\begin{aligned}
 \{y\}_m &= [\bar{A}]^{-1} \{\bar{r}\}_m \\
 \{x_m\} &= \{x_1\} + [V]_m \{y\}_m \\
 \{r_m\} &= \{b\} - [A]\{x_m\}
 \end{aligned}
 \tag{4.8}$$

La qualité de la solution $\{x_m\}$ est contrôlée par la norme $\|r_m\|$

Une méthode itérative est caractérisée par :

1. Le choix de la base $[V]_m$ de l'espace solution et la technique de construction de cette base.
2. Le choix de la base $[W]_m$ de l'espace test et la technique de construction de cette base.
3. La stratégie de l'addition de nouveaux vecteurs dans les espaces solution et test pour maintenir une meilleure efficacité numérique. La technique souvent utilisée est la suivante :

Supposant que nous ayons obtenu la solution avec $\{v_1\}, \dots, \{v_i\}$. Le résidu $\{r_{i+1}\}$ est :

$$\{r_{i+1}\} = \{r_i\} - [A]\{v_i\}y_i \tag{4.9.a}$$

Le vecteur $\{v_{i+1}\}$ est obtenu par orthogonalisation :

$$\{v_{i+1}\} = \{r_{i+1}\} + \sum_{j=1}^i \{v_j\} h_{ij} \tag{4.9.b}$$

L'espace de Krylov peut être :

$$K_m = \text{esp}[\{r_1\}, \{r_2\}, \dots, \{r_{m-1}\}]$$

ou

$$K_m = \text{esp}[\{r_1\}, [A]\{r_1\}, \dots, [A]^{m-1}\{r_1\}] \tag{4.9.c}$$

Les coefficients h_{ij} sont calculés en imposant la condition que $\{v_{i+1}\}$ soit orthogonal à $\{v_i\}$ (GMRES, FOM), à $[A]\{v_i\}$ (GC) et à $[A]^T\{v_i\}$ (BICG, BICGSTAB, CGS, TFQMR) selon la méthode itérative choisie. Nous pouvons écrire [4.9.c] sous la forme matricielle suivante :

$$[\{v_2\}, \{v_3\}, \dots, \{v_{i+1}\}] = [\{r_2\}, \{r_3\}, \dots, \{r_{i+1}\}] \oplus [\{v_2\}, \{v_3\}, \dots, \{v_i\}] \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1i} \\ 0 & h_{22} & \dots & h_{2i} \\ \dots & 0 & \dots & \dots \\ 0 & \dots & 0 & h_{ii} \end{bmatrix} \quad [4.10.a]$$

avec $\{v_1\} = \{r_1\}$

ou bien
$$[V] = [R] + [V]_i [H] \quad [4.10.b]$$

$$\begin{matrix} & n_{xi+1} & & \\ & n_{xi+1} & & \\ & n_{xi} & & i_{xi} \end{matrix}$$

Pour la méthode du gradient conjugué, seuls h_{ii} sont non nuls.

Si l'espace K_m est composé de $(A^{m-1} v_1)$, le vecteur $\{r_{i+1}\}$ est remplacé par $\{Av_i\}$.

Le choix de K_m et L_m conduit à différentes méthodes :

- L'espace test est de type Galerkin si : $[W]_m = [V]_m \quad [4.11]$

La méthode du gradient conjugué est de type Galerkin où les vecteurs $\{v_i\}$ sont A-orthogonaux ou conjugués.

- Le critère d'orthogonalité est de type moindres-carrés si :

$$[W]_m = [A][V]_m \quad [4.11.a]$$

Dans ce cas, nous cherchons la solution ayant la plus petite norme Euclidienne.

- Dans le cas d'une résolution directe, nous avons :

$$[W]_m = [V]_m = [I]_{n \times n} \text{ avec } m = n \quad [4.11.b]$$

- L'espace K_m est en général de type Krylov :

$$K_m = \text{Esp}[\{v_1\}, [A]\{v_1\}, \dots, [A]^{m-1}\{v_1\}]$$

Par contre, la base $\{v_i\}$ de cet espace est différente selon la méthode itérative.

- L'espace L_m peut être de type Galerkin, moindres-carrés ou Krylov utilisant la matrice $[A]^T$.

L'algorithme général des méthodes itératives est le suivant :

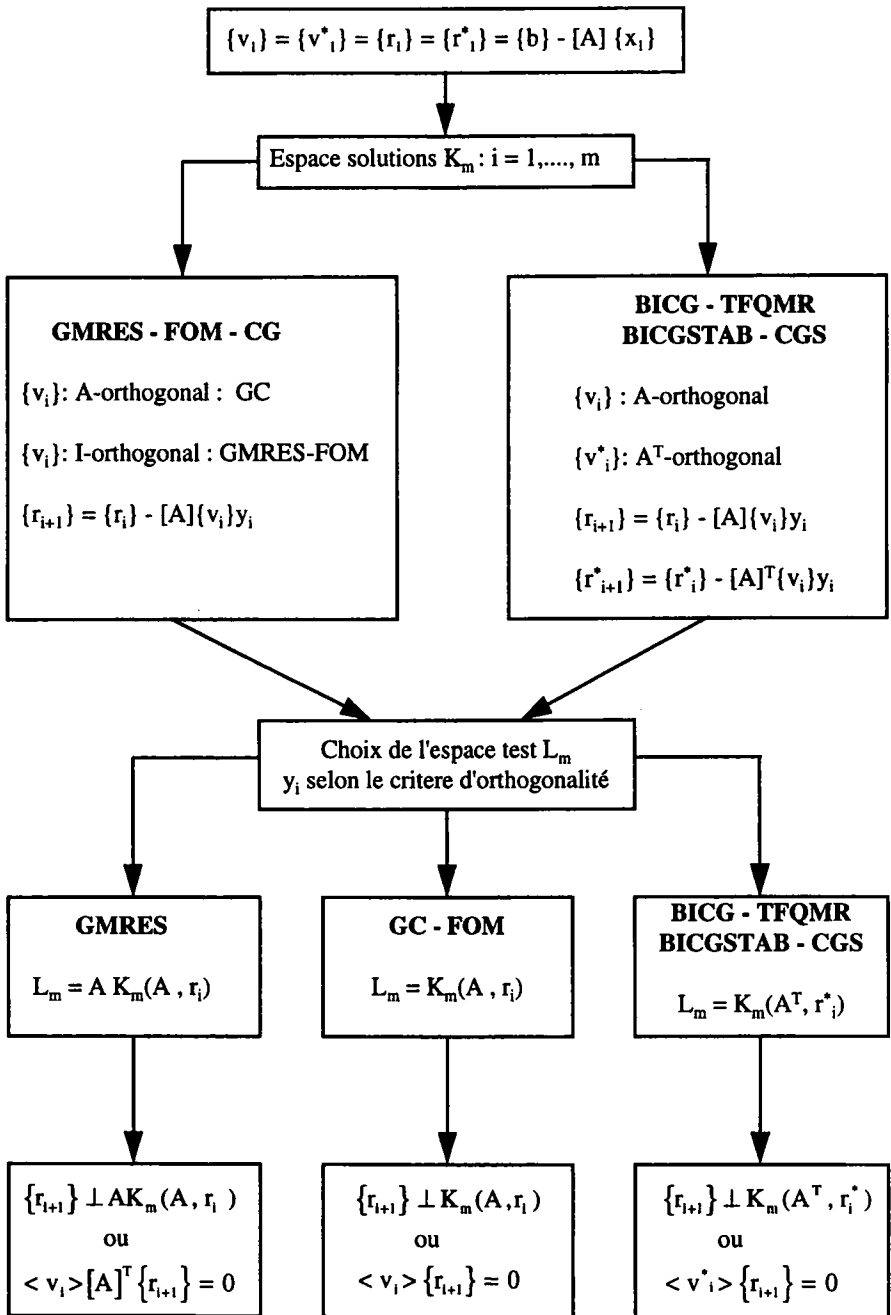


Figure 4.1. Algorithme général des méthodes itératives

L'algorithme pour la construction de la base $\{v_i\}$ est donné dans la figure suivante :

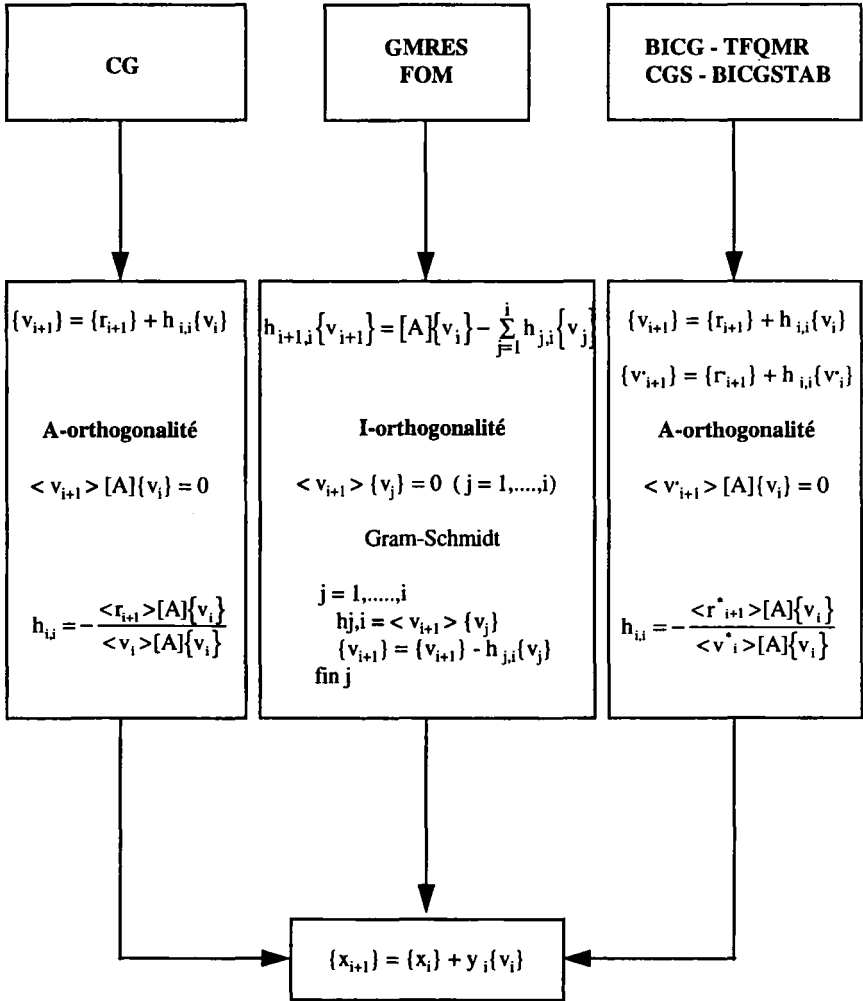


Figure 4.2. Construction de la base $\{v_i\}$

En respectant les deux algorithmes proposés ci-dessus, nous pouvons déduire tous les algorithmes des différentes méthodes itératives.

5. Technique de préconditionnement

5.1. Généralités

La rapidité de convergence des méthodes itératives dépend du nombre conditionnement de la matrice $[A]$. Plus $\text{cond}(A)$ est proche de 1, plus vite convergera l'algorithme. Le principe du préconditionnement d'une matrice $[A]$ consiste à remplacer la résolution de l'équation $[A]\{x\} = \{b\}$ par celle du système équivalent $[A']\{x'\} = \{b'\}$ avec l'objectif que $\text{cond}(A')$ soit beaucoup plus petit que $\text{cond}(A)$.

Soit $[M]$ la matrice de préconditionnement de dimension $(n \times n)$. Supposant que $[M]$ est décomposée sous la forme :

$$[M] = [M_1][M_2]$$

Le système à résoudre devient alors :

$$[A']\{x'\} = \{b'\}$$

avec $[A'] = [M_1^{-1}][A][M_2^{-1}]$, $\{x'\} = [M_2]\{x\}$ et $\{b'\} = [M_1^{-1}]\{b\}$.

En théorie, le meilleur choix est de prendre $[M] = [A]$ car alors $\text{cond}(A') = 1$. En pratique, il faudra trouver $[M]$ le plus proche de $[A]$ sans que les calculs pour $[M]$ soient trop coûteux. Les matrices $[M_1]$ et $[M_2]$ sont calculées de telle manière à préserver la structure de la matrice $[A]$. Si par exemple $[A]$ est symétrique définie positive, on prendra $[M_2] = [M_1^T]$ de telle manière que $[A']$ reste symétrique.

Si $[M_1] = [I]$: Préconditionnement à droite et dans ce cas $\{b'\} = \{b\}$.

Si $[M_2] = [I]$: Préconditionnement à gauche et dans ce cas $\{x'\} = \{x\}$.

Si $[M_1] \neq [I]$ et $[M_2] \neq [I]$, on parlera alors d'un reconditionnement partagé.

5.2. Préconditionnement par factorisation incomplète.

5.2.1. Définitions

Avant de décrire ce qu'est la factorisation incomplète, examinons ce qui se passe lorsqu'on effectue une factorisation complète de la matrice $[A]$ par la méthode d'élimination de Gauss. Lorsque l'on factorise une matrice inversible $[A]$, on obtient deux matrices $[L]$ et $[U]$ respectivement triangulaire inférieure et triangulaire supérieure telle que :

$$[A] = [L][U]$$

Cependant lors d'une factorisation complète, un problème sérieux apparaît, celui du remplissage. Les matrices qui nous intéressent proviennent de la méthode des éléments finis, ces matrices ont la particularité et l'intérêt d'être très creuses, mais hélas très grandes. La factorisation produit sur celles-ci un remplissage considérable et le temps de l'opération devient alors astronomique. L'idée de la factorisation incomplète est d'effectuer les opérations seulement sur un certain nombre de termes et de refuser que tout le remplissage ne se produise. Evidemment, la factorisation incomplète ne produit pas des facteurs $[L]$ et $[U]$ tels que $[L][U] = [A]$, le résultat est plutôt deux matrices $[L']$ et $[U']$ triangulaires elles-aussi et telles que :

$$[L'][U'] = [A] + [R]$$

$[R]$ étant l'erreur commise lors de la factorisation incomplète. L'objectif est d'utiliser $[L'][U']$ comme matrice de préconditionnement.

On peut refuser le remplissage durant la factorisation suivant deux critères :

- Soit par rapport à la position du terme dans la matrice et dans ce cas, on n'effectue les opérations que sur un ensemble F d'indices définis à l'avance.
- Soit par rapport à la valeur numérique du terme et dans ce cas, on ne garde que les termes supérieurs à un certain coefficient de tolérance.

Il est clair que lorsqu'on effectue une décomposition incomplète par position, l'espace mémoire pour la matrice décomposée est connue à l'avance suivant l'ensemble d'indices F , par contre lors d'une décomposition incomplète par valeur, l'espace mémoire alloué dépendra du coefficient de tolérance et des valeurs de la matrice $[A]$.

5.2.2. Factorisation incomplète par position : $ILU(\alpha)$

Le problème qui se pose maintenant est de savoir comment choisir l'ensemble F . La réponse à cette question dépend de notre tolérance au remplissage ' α '. Si le degré de tolérance ' α ' face au remplissage est élevé, la matrice d'erreur $[R]$ sera moins importante et dans ce cas $[M] = [L'][U']$ sera un meilleur préconditionneur, cependant le temps de factorisation et l'espace mémoire deviennent trop grand. D'un autre côté, un degré de tolérance faible facilite le calcul mais éloigne $[M]$ de la matrice $[A]$. Ce conflit tend à rendre difficile un choix optimal pour l'ensemble F .

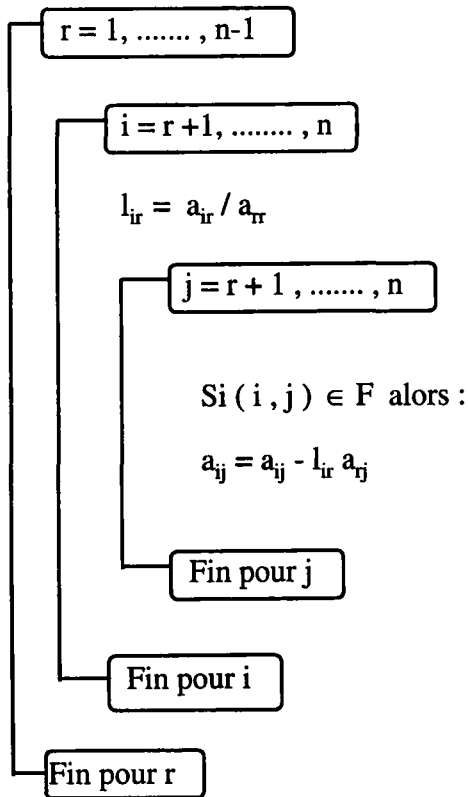


Figure 5.1. *Algorithme de factorisation incomplète ILU*

Le choix le plus courant pour F est de ne considérer que les termes non nuls de $[A]$, ce qui revient à dire qu'il n'y a pas de remplissage lors de la factorisation incomplète. Dans ce cas, $\alpha = 0$ et $F(0) = \{(i,j) / a_{ij} \neq 0\}$. Cette décomposition est notée $ILU(0)$. En additionnant à l'ensemble $F(0)$ d'autres paires d'indices, cette décomposition est notée $ILU(n)$, 'n' étant le paramètre de remplissage. Il existe une relation entre les ensembles $F(n)$ et $F(n-1)$. En effet, soit $F(n-1)$ l'ensemble d'indices de niveau $(n-1)$, en effectuant la décomposition de Gauss sur cet ensemble, nous obtenons deux matrices $[L]$ et $[U]$ correspondant à $ILU(n-1)$. Si le produit $[L][U]$ est effectué, d'autres paires d'indices apparaîtront en plus des indices appartenant à $F(n-1)$, ces nouvelles paires d'indices sont rajoutées à cet ensemble pour constituer l'ensemble $F(n)$.

Notons que pour chercher cet ensemble $F(n)$, aucune multiplication de matrices n'est effectuée, l'algorithme suivant est utilisé :

Initialement :

$$\text{Re mpl}(a_{ij}^{(0)}) = \begin{cases} 0 & \text{si } a_{ij} \neq 0 \\ \infty & \text{si } a_{ij} = 0 \end{cases}$$

L'élimination de la ligne 'k' dans le processus de Gauss est :

$$a_{ij} = a_{ij}^{(k-1)} - \frac{a_{ik}^{(k-1)} - a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}$$

En effectuant la décomposition de Gauss pour tous les termes, on calculera les niveaux de remplissage comme suit :

$$\text{Re mpl}(a_{ij}^{(k)}) = \text{Min} \left[\text{Re mpl}(a_{ik}^{(k-1)}) + \text{Re mpl}(a_{kj}^{(k-1)}) + 1, \text{Re mpl}(a_{ij}^{(k-1)}) \right]$$

Dans notre étude, le préconditionnement ILU(0) est uniquement utilisé.

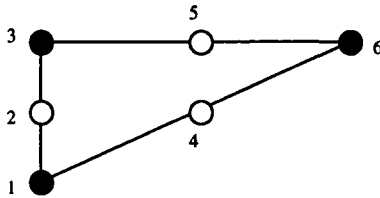
6. Numérotation des variables

La résolution d'un système d'équations $[A]\{x\} = \{b\}$ se fait soit par la méthode directe (méthode de Gauss) avec un stockage 'ligne de ciel', soit par une méthode itérative avec un stockage morse. En ce qui concerne la méthode directe, même si la renumérotation des variables est importante en termes de stockage (réduction de la largeur de bande), elle n'a aucune influence sur la convergence de la méthode. Par contre, une 'bonne' renumérotation améliore sensiblement le taux de convergence des méthodes itératives.

Notre objectif principal est d'étudier les écoulements incompressibles. Le modèle discret est représenté par des éléments finis où 3 degrés de liberté sont associés aux nœuds sommets et 2 degrés de liberté aux nœuds milieux. Dans la démarche éléments finis classique, les nœuds sont numérotés et les degrés de liberté de chaque nœud sont alors regroupés. Dans notre cas, chaque degré de liberté est numéroté, puis la matrice globale par blocs de degré de liberté est assemblée. Cette manière de procéder nous permet de remplir la sous-matrice de pression lors de la décomposition de Gauss, et donc d'améliorer son conditionnement.

A titre d'exemple, nous donnons la topologie de trois matrices pour trois numérotations différentes d'un seul élément fini. La première consiste à numéroté les nœuds puis à regrouper les degrés de liberté par blocs de 3x3 ou 2x2, c'est la démarche élément finis classique. La deuxième consiste à numéroté par blocs de

degrés de liberté de la manière suivante : $(u_1, u_2, \dots) + (v_1, v_2, \dots) + (p_1, p_2, \dots)$. La troisième consiste à numéroté par blocs de degrés de liberté de la manière suivante : $(u_1, v_1, u_2, v_2, \dots) + (p_1, p_2, \dots)$. L'élément fini est l'élément P2-P1 présenté dans le paragraphe 2. Les nœuds sommets possèdent 3 degrés de liberté (u, v, p) , les nœuds milieux possèdent 2 degrés de liberté (u, v) .



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_1	v_1	p_1	u_2	v_2	u_3	v_3	p_3	u_4	v_4	u_5	v_5	u_6	v_6	p_6

Figure 6.1. Numérotation classique élément finis : blocs 3×3 ou 2×2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_1	u_2	u_3	u_4	u_5	u_6	v_1	v_2	v_3	v_4	v_5	v_6	p_1	p_2	p_3

Figure 6.2. Numérotation par groupe de DDL $(u_1, u_2, \dots) + (v_1, v_2, \dots) + (p_1, p_2, \dots)$

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_1	v_1	u_2	v_2	u_3	v_3	u_4	v_4	u_5	v_5	u_6	v_6	p_1	p_2	p_3

Figure 6.3. Numérotation par groupe de DDL $(u_1, v_1, u_2, v_2, \dots) + (p_1, p_2, \dots)$

Dans nos tests numériques, nous comparerons ces trois types de numérotation.

7. Choix du critère d'arrêt

A l'inverse de la méthode directe, un avantage des méthodes itératives est de pouvoir contrôler la précision de la solution. Un critère d'arrêt sévère nous donne une bonne approximation de la solution mais augmente le temps CPU. Si la précision demandée est petite, le temps CPU devient petit, mais il existe un risque de divergence. L'efficacité de la méthode de Newton-Raphson couplée aux méthodes itératives est très liée au choix du critère d'arrêt des méthodes itératives Eps_n . Dans notre cas, le critère de convergence est tel que :

$$\frac{\| [K_T]_n \{ \Delta u_n \} - \{ R_n \} \|}{\| \{ R_n \} \|} < Eps_n$$

n étant la n^{ème} itération de Newton-Raphson.

Le choix optimal de Eps_n est très empirique. Dans notre cas, deux options ont été choisies : dans la première option, Eps_n est égal à une valeur constante. Dans la deuxième option, Eps_n varie en fonction des itérations de Newton-Raphson.

8. Résultats numériques

Les objectifs de cette section sont :

- 1- Tester la numérotation proposée qui consiste à renuméroter les degrés de liberté par blocs.
- 2- Tester les différents critères d'arrêts des méthodes itératives pour le pilotage du modèle.

Dans tous les exemples qui suivent, seul le préconditionnement ILU(0) est utilisé. Notons que les méthodes BICG et FOM ne sont pas étudiées. La première n'a que peu d'intérêt (produit par la matrice [A^T]), la deuxième appartenant à la même classe que GMRES est beaucoup moins performante que celle-ci.

Problème. Chambre

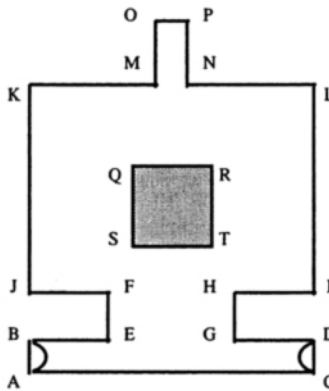
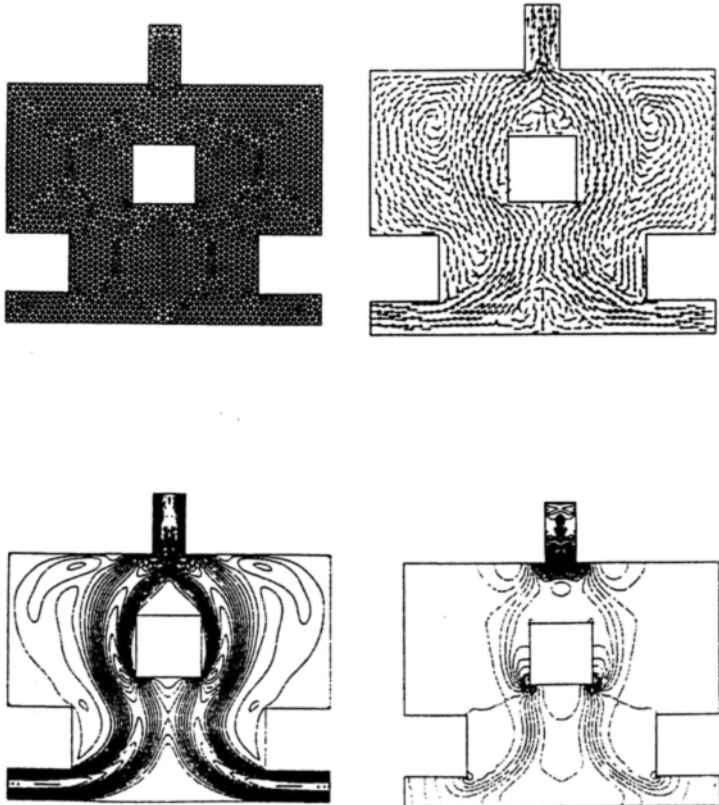


Figure 8.1. Géométrie du problème

|AB| = |CD| = |OP| = 0.1m, |EF| = |HG| = 0.2m, |JK| = |IL| = 0.5m, |KM| = |LN| = 0.45m
 |OM| = |NP| = 0.2m, |SQ| = |QR| = |RT| = |ST| = 0.2m, |BE| = |GD| = |JF| = |HI| = 0.2m

Caractéristiques du problème

NEQ	NELT	Elément	Largeur de bande	Base de Krylov
21.000	4300	P2-P1	251	40

Tableau 8.1. *Caractéristiques du problème*

Maillage, vecteurs vitesses, isolignes et pressions pour $Re = 200$ (viscosité = 0.0005)

	<u>Nbre de termes stockés dans la matrice [K]</u>	<u>Rapport (Itératives/Directe)</u>
Directe	10.197.282	4.97 %
Itérative	507.082	

Tableau 8.2. Espace mémoire occupé par la matrice tangente globale [K]

Equation de Stokes

Afin de comparer les différentes numérotations de variables, nous traitons en premier l'équation de Stokes. La viscosité du fluide est prise égale à 10^{-02} . La précision pour les méthodes itératives est égale à $Eps = 10^{-04}$

Effet de la numérotation

Comme nous l'avons proposé dans la section 6, la numérotation adoptée est celle qui numérote en premier les vitesses suivies de la pression. Dans ce test, les trois numérotations décrites ont été testées. La première consiste à numérotter les vitesses u suivies des vitesses v et en dernier la pression. Cette numérotation sera notée numérotation 1. La deuxième numérotation consiste à prendre pour chaque nœud n , les vitesses u_n et v_n , puis, lorsque toutes les vitesses ont été numérotées, les termes de pression sont à leur tour numérotés. Cette numérotation sera notée numérotation 2.

Méthode	Numérotation 1 Itération / CPU (Sec)	Numérotation 2 Itérations / CPU (Sec)	Numérotation éléments finis
CGS	105 / 67	90 / 61	Diverge
BICGSTAB	158 / 92	148 / 88	Diverge
TFQMR	151 / 88	148 / 80	Diverge
GMRES	735 / 320	1187 / 520	Diverge

Tableau 8.3. Différentes numérotations avec le préconditionnement ILU(0)

Comme nous pouvons le constater dans le tableau 8.3, la numérotation 2 donne de meilleurs résultats sauf pour la méthode GMRES. D'autre part, toutes les méthodes itératives divergent avec la numérotation standard de la méthode des éléments finis. Ceci est dû aux valeurs des termes de pressions situés sur la diagonale principale. Leurs valeurs restent très petites par rapport aux termes de vitesses même après la décomposition ILU. Cet état de fait nous conduit alors à une matrice très mal conditionnée. Dans tous les autres exemples traités, la numérotation 1 sera uniquement utilisée.

Equation de Navier-Stokes

Afin de voir l'importance du critère d'arrêt sur le gain de temps CPU, nous traitons cette fois-ci les équations de Navier-Stokes. Nous comparons les résultats obtenus par les différentes méthodes itératives pour différents critères d'arrêt avec le temps CPU donné par la méthode directe. Le pilotage en viscosité se fait de la manière suivante : nous prenons la solution de Stokes comme solution initiale puis nous diminuons la viscosité μ de manière à obtenir les différents nombres de Reynolds ci-dessus.

Re = 0 ----->	Re = 100 ----->	Re = 220 ----->	Re = 350 ----->	Re = 400
($\mu = 0.1$)	($\mu = 0.001$)	($\mu = 0.00045$)	($\mu = 0.00029$)	($\mu = 0.00025$)

La précision du schéma de Newton-Raphson est telle que : $\| \Delta u \| / \| u \| < 10^{-04}$
 Notons que dans les tests suivants, la méthode GMRES stagne.

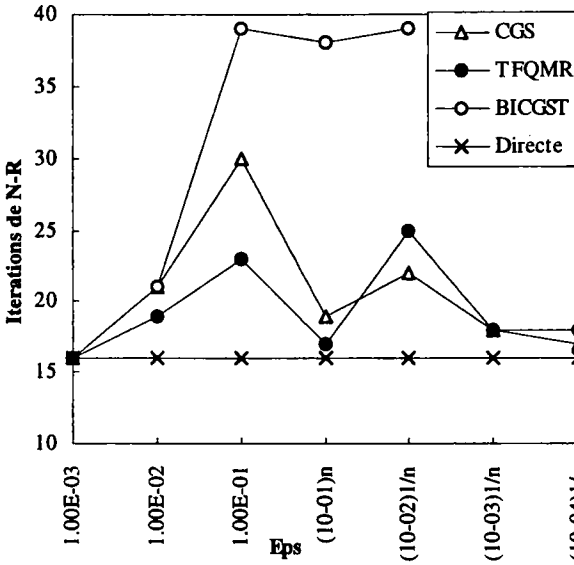


Figure 8.2. Nombre total d'itérations de Newton-Raphson en fonction du paramètre d'arrêt Eps_n pour $Re = 400$. n est le nombre d'itération de Newton-Raphson

Les figures 8.2 et 8.3 donnent respectivement le nombre total d'itérations de Newton-Raphson et le temps CPU en seconde en fonction de Eps_n pour un nombre de Reynolds égal à 400. Les résultats montrent que la condition $Eps_n < 10^{-03}$ est trop restrictive. Il est vrai qu'avec cette condition, nous obtenons le nombre minimum d'itérations de Newton-Raphson, c'est-à-dire celui de la méthode directe, mais par contre, le temps CPU est grand comparé à celui donné par les autres critères d'arrêt. A l'inverse, la condition $Eps_n > 10^{-01}$ n'est pas assez restrictive et nous obtenons alors un grand nombre d'itérations de Newton-Raphson.

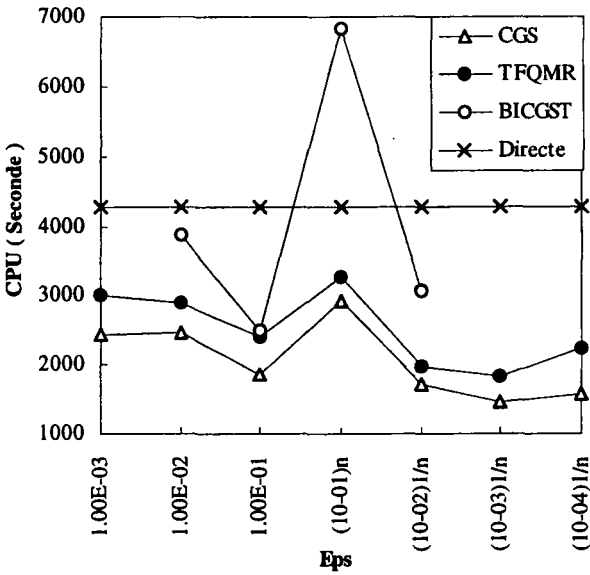


Figure 8.3. Temps CPU en seconde en fonction du paramètre d'arrêt Eps_n pour $Re = 400$. n est le nombre d'itération de Newton-Raphson

	Nombre moyen d'itérations pour les méthodes itératives						
	10^{-01}	10^{-02}	10^{-03}	$(10^{-01})^n$	$(10^{-02})^{1/n}$	$(10^{-03})^{1/n}$	$(10^{-04})^{1/n}$
CGS	92	194	265	262	121	126	150
TFQMR	165	268	314	322	118	160	202
BICGSTAB	96	316	Diverge	346	128	Diverge	Diverge

Tableau 8.4. Nombre moyen d'itérations en fonction du paramètre d'arrêt Eps_n

La question qui se pose alors est comment choisir Eps_n en fonction des itérations de Newton-Raphson. Est-ce que Eps_n doit être très petit au début et augmenter au fur et à mesure des itérations de Newton-Raphson, ou bien l'inverse, c'est-à-dire, grand au début et diminuer au fur et à mesure des itérations. Les résultats obtenus montrent clairement que la première option est celle qui donne les meilleurs résultats. En effet, pour $Eps_n = (10^{-03})^{1/n}$, le temps CPU est minimum et le nombre d'itérations de Newton-Raphson représente que 1.2 fois celui de la méthode directe pour CGS, et 1.13 fois pour TFQMR. Notons que pour cette valeur de Eps_n , BICGSTAB ne converge pas à cause de la restriction du critère d'arrêt imposé au début pour $n = 1$. Il est intéressant aussi de mentionner que quel que soit Eps_n choisi, les performances en temps CPU des méthodes itératives sont toujours supérieures à celles de la méthode directe.

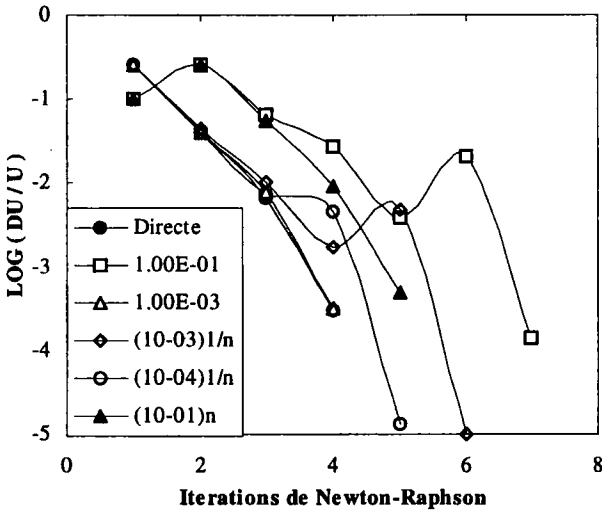


Figure 8.4. Variation de $\| \Delta u \| / \| u \|$ en fonction des itérations de Newton-Raphson lorsque le critère d'arrêt change. Dans ce cas, la méthode de résolution est CGS et $Re = 220$

La figure 8.4 donne l'évolution de la norme $\| \Delta u \| / \| u \|$ pour chaque itération de Newton-Raphson lorsque le critère d'arrêt varie. La méthode itérative choisie est CGS. Pour $Eps_n = 10^{-03}$, la variation de $\| \Delta u \| / \| u \|$ coïncide exactement avec celle donnée par la méthode directe. Lorsque $Eps_n = (10^{-03})^{1/n}$ ou $(10^{-04})^{1/n}$, cette variation est semblable à celle de la méthode directe pour les premières itérations de Newton-Raphson, puis s'écarte de celle-ci au fur et à mesure. A l'inverse, pour $Eps_n = (10^{-01})^n$, les valeurs de $\| \Delta u \| / \| u \|$ sont très différentes de celles de la méthode directe pour les premières itérations de Newton-Raphson, mais par contre garde la même allure que celle de la méthode directe pour les itérations qui suivent.

9. Conclusion

les résultats numériques obtenus lors de cette étude conduisent aux remarques suivantes :

- - Le préconditionnement basé sur la factorisation incomplète ILU est très efficace pour les méthodes itérative étudiées.
- - La renumérotation des variables est capitale pour ce type de préconditionnement. La technique basée sur le regroupement des variables par nœuds n'est pas très conseillée. On suggère un schéma de renumérotation groupant d'abord les composantes des vitesses suivies des variables de pression.
- - Nous avons montré que le critère d'arrêt des méthodes itératives avait son importance. En effet, pour différents exemples numériques, le critère d'arrêt variable $Eps_n = (10^{-03})^{1/n}$, (n étant la n^{ieme} itération du schéma non linéaire global) était le meilleur compromis entre le temps CPU et le nombre d'itérations total du schéma non linéaire.
- - Parmi les quatre méthodes itératives utilisées, CGS semble être la plus efficace.
- - Pour les problèmes étudiés, en plus des avantages en termes de mémoire, les méthodes itératives sont en général plus efficaces que la méthode directe. L'efficacité de telles méthodes sera plus accentuée lors de l'utilisation de machine parallèle.

10. Bibliographie

- [COC 94] B. COCHELIN, "Méthodes Asymptotiques-Numériques pour le calcul non-linéaire géométrique des structures élastiques", Habilitation à diriger des recherches, Université de Metz, 1994.
- [DAM 90] N. DAMIL, M. POTIER-FERRY, "A new method to compute perturbed bifurcations : Application to the buckling of imperfect elastic structures", *Int. J. Engineering Sciences*, vol. 28, 704-719, (1990).
- [DHA 84] G. DHATT, G. TOUZOT, *Une présentation de la méthode des éléments finis*. Maloine, 1984.
- [DHA 95] G. DHATT, M. FAFAR, "Mécanique non-linéaire", Cours IPSI, Paris, 1995.
- [FRE 93] R. W. FREUND, "A Transpose-Free quasi-minimal residual algorithm for the non-hermitian linear systems", *SIAM J. Sci. Statist. Comput.*, vol. 14, pp. 470-482, 1993.
- [HAD 95] S.HADJI, "Méthode de résolution pour les fluides incompressibles", thèse de doctorat, Université de Technologie de Compiègne, 1995.
- [HAD 97] S.HADJI, G. DHATT, "Asymptotic-Newton method for solving incompressible flows", *Int. J. Numer. Methods fluids*, à paraître.

- [HOW 90] D. HOWARD, W. M. CONNOLLEY, J. S. ROLLETT, "Unsymmetric conjugate gradient methods and sparse direct methods in finite element flow simulation", *Int. J. Num. Methods fluids*, vol. 10, 925-945, 1990.
- [MEI 81] J. A. MEIJERINK, H. A. VAN DER VORST, "Guidelines for the usage of incomplete decompositions in solving sets of linear equations as they occur in practical problems", *Journal of Computational Physics*, 44, 134-155, 1981.
- [SAA 86] Y. SAAD, M. H. SCHULTZ, "GMRES : a generalized minimal residual algorithm for solving nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.*, vol. 7, pp. 856-869, 1986.
- [SON 89] P. SONNEVELD, "CGS, a fast Lanczos-type solver for nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.*, vol. 10, pp. 36-52, 1989.
- [STO] M. STORI, N. NIGROT, S. IDELSOHN, "Stabilising equal-order interpolations for mixed formulations of Navier-Stokes equations via SUPG method", Research Report, Universidad Nacional del litoral and CONICET, Argentina.
- [VAN 92] H. A. VAN DER VORST, "BI-CGSTAB : a fast and smoothly converging variant of BI-CG for the solutions of nonsymmetric linear systems", *SIAM J. Sci. Statist. Comput.*, vol. 13, pp. 631-644, 1992.