

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Méthodes Numériques Ecoulements Incompressibles

Pierre Jacquet Guillaume Ravel

Enseignant : Y. Coudière

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

troduction

Modélisation mathématique

Résolution numérique

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres

Ker(B^T)

Sommaire

Introduction

Modélisation mathématique

Schéma MAC

Résolution numérique

Code

Algorithme d'Uzawa

Cas tests

Résolution du Laplacien

Cas Test 0

Cas Test 0

Poiseuille

– Coiseuille

Erreurs et ordres

 $Ker(B^T)$

Influence de μ

Conclusion

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

troduction

Modélisation mathématique

Schéma MAC

Résolution umérique

Code Algorithme d'Uzawa

as tes

Résolution du Laplacien

Cas Test 0
Cas Test 1
Poiseuille

Poiseuille Erreurs et ordres

 $Ker(B^T)$



Introduction

Problème de Stokes

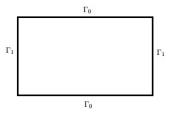


Figure – Domaine Ω

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

Introduction

Modélisation mathématique Schéma MAC

Résolution numérique Code Algorithme d'Uzawa

Cas tes

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres

 $Ker(B^T)$ Influence of

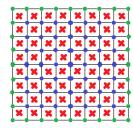
Conclusion

4□ > 4□ > 4□ > 4□ > 4□ > 4□

Modélisation mathématique

Schéma MAC

$$\left\{ \begin{array}{l} \frac{u_{1,n+1}^{1}-1,n}{u_{i,j}^{1}}-\nu_{i,j}^{1}-\nu_{i+1,j+1}^{1}+\nu_{i+1,j+1}^{1}-\nu_{i-1,j}^{1}-2\nu_{i,j}^{1,n+1}}{dx^{2}} + \frac{u_{i,j+1}^{1,n+1}+u_{i,j-1}^{1,n+1}-2u_{i,j}^{1,n+1}}{dy^{2}} \right) \\ \frac{p_{n+1}^{n+1}}{p_{i+\frac{1}{2},j+\frac{1}{2}}+p_{i+\frac{1}{2},j-\frac{1}{2}}} \frac{1-p_{n+1}^{n+1}}{j-\frac{1}{2},j+\frac{1}{2}} - \frac{1}{2},j-\frac{1}{2}}{j-\frac{1}{2},j-\frac{1}{2}} = f_{i,j}^{1,n} \\ + \frac{u_{i,j}^{2}+1-u_{i,j}^{2,n}}{j-\frac{1}{2},j-\frac{1}{2}} - \nu_{i+1,j}^{2,n+1}+u_{i-1,j}^{2,n+1}-2\nu_{i,j}^{2,n+1}} \\ \frac{u_{i,j}^{2,n+1}-u_{i,j}^{2,n}}{j-\frac{1}{2},j-\frac{1}{2}} - \nu_{i+1,j}^{2,n+1}-2\nu_{i,j}^{2,n+1}-2\nu_{i,j}^{2,n+1}-2\nu_{i,j}^{2,n+1}} \\ \frac{v_{i,j}^{2,n+1}-v_{i,j}^{2,n+1}-v_{i,j+1}^{2,n+1}-2\nu_{i,j}^{2,n+1}}{j-\frac{1}{2},j-\frac{1}{2}} - \frac{v_{i+1,j}^{2,n+1}-v_{i,j+1}^{2,n+1}-v_{i,j+1}^{2,n+1}-v_{i,j+1}^{2,n+1}}{j-\frac{1}{2},j+\frac{1}{2}} = f_{i,j}^{2,n} \\ -\frac{u_{i+1,j+1}^{2,n+1}+u_{i+1,j}^{2,n+1}-u_{i,j+1}^{2,n+1}-u_{i,j+1}^{2,n+1}-u_{i,j+1}^{2,n+1}-v_{i+1,j}^{2,n+1}-v_{i+1,j}^{2,n+1}-v_{i+1,j}^{2,n+1}}{2dx} = 0 \end{array} \right.$$



- Valeurs connues en vitesse (conditions limites)
- Inconnues en vitesse
- Inconnues en Pression

Figure – Illustration du maillage - schéma MAC

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

Résolution numérique Code Algorithme

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres Ker(B^T)

influence de μ

Modélisation mathématique

Système matriciel

D'où le système matriciel suivant :

$$\begin{pmatrix} A & 0 & B_1^T \\ 0 & A & B_2^T \\ B1 & B2 & 0 \end{pmatrix} \cdot \begin{pmatrix} U_x^{n+1} \\ U_y^{n+1} \\ P^{n+1} \end{pmatrix} = \begin{pmatrix} F_x^n \\ F_y^n \\ 0 \end{pmatrix} + \frac{1}{dt} \begin{pmatrix} U_x^n \\ U_y^n \\ 0 \end{pmatrix} + CL_{\text{Modélisation mathématique Schéma MAC}}$$

$$B1 = \begin{pmatrix} S & 0 & 0 \\ S & \ddots & 0 \\ 0 & \ddots & S \\ 0 & 0 & S \end{pmatrix} S = \frac{1}{2dx} \begin{pmatrix} -1 & 0 & 0 \\ 1 & \ddots & 0 \\ 0 & \ddots & -1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$B2 = \begin{pmatrix} -R & 0 & 0 \\ R & \ddots & 0 \\ 0 & \ddots & -R \end{pmatrix} R = \frac{1}{2dy} \begin{pmatrix} 1 & 0 & 0 \\ 1 & \ddots & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R = \frac{1}{2dy} \begin{pmatrix} 1 & 0 & 0 \\ 1 & \ddots & 0 \\ 0 & \ddots & 1 \end{pmatrix}$$

$$R = \frac{1}{2dy} \begin{pmatrix} 1 & 0 & 0 \\ 1 & \ddots & 0 \\ 0 & \ddots & 1 \end{pmatrix}$$

$$R = \frac{1}{2dy} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \ddots & 1 \end{pmatrix}$$

$$R = \frac{1}{2dy} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \ddots & 1 \end{pmatrix}$$

Résolution de l'équation de Stokes en schéma MAC. par la méthode d'Uzawa et solveur itératif. en Fortran90

Pierre Jacquet. Guillaume Ravel

Schéma MAC

Code Algorithme

d'Uzawa

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres

 $Ker(B^T)$

Résolution numérique

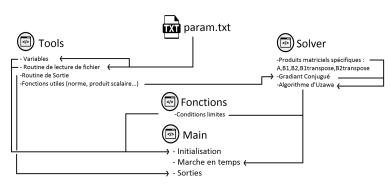


Figure – Organisation du code

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique Schéma MAC

Résolution

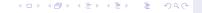
Code

Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille

Erreurs et ordres $Ker(B^T)$



Résolution numérique

Algorithme d'Uzawa

$$\left(\begin{array}{cc} A & B^{T} \\ B & 0 \end{array}\right) \cdot \left(\begin{array}{c} U \\ P \end{array}\right) = \left(\begin{array}{c} F \\ 0 \end{array}\right) \tag{4}$$

U réalise aussi le minimum de $J(U) = \frac{1}{2}U^TAU - U^TF$ sous la contrainte BU = 0

P est un multiplicateur de Lagrange : $\langle U, B^T P \rangle = 0$

U réalisant le minimum de J convexe, on a alors la relation :

$$\nabla J = -B^T P$$

Lors d'un pas de temps, le système (4) peut alors être résolu avec l'algorithme d'Uzawa :

- ▶ Initialisation : on fixe μ et P^0
- ▶ Tant que : $||P^{n+1} P^n|| = \mu ||BU^n|| > \epsilon$ Faire
 - ► Résolution de $AU^n = F B^T P^n$ par l'algorithme du gradient conjugué
 - ► Mise à jour de P : $P^{n+1} = P^n + \mu BU^n$

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modelisation mathématique Schéma MAC

Résolution numérique Code

> Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres Ker(B^T)

Influence de μ

Résolution du Laplacien

Cas test 1:

$$\begin{cases} u(x, y) = x(1 - x)y(1 - y) \\ f(x, y) = 2(y - y^{2} + x - x^{2}) \\ g = h = 0 \end{cases}$$
 (5)

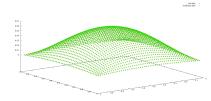


Figure – Visualisation des solutions numérique et exacte - Cas test 1

Cas test 2:

$$\begin{cases} u(x, y) = \sin(x) + \cos(y) \\ f(x, y) = \sin(x) + \cos(y) \\ g = h = f \end{cases}$$
 (6)

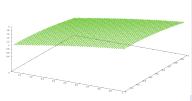


Figure – Visualisation des solutions numérique et exacte – Cas test 2

 $\mathsf{Ker}(\mathsf{B}^{\mathsf{T}})$ Influence de μ

l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Résolution de

Pierre Jacquet, Guillaume Ravel

itroduction

Modélisation mathématique Schéma MAC

ésolution

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien

Cas Test 0
Cas Test 1
Poiseuille
Erreurs et ordres

Influence de μ

Rsolution du Laplacien

Cas test 1:

$$\begin{cases} u(x, y) = x(1 - x)y(1 - y) \\ f(x, y) = 2(y - y^2 + x - x^2) \\ g = h = 0 \end{cases}$$
 (7)

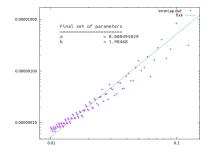


Figure – Comparaison de l'erreur en norme 2 suivant dx avec $f(x) = ax^b$ (échelle log-log)

Cas test 2:

$$\begin{cases} u(x, y) = \sin(x) + \cos(y) \\ f(x, y) = \sin(x) + \cos(y) \\ g = h = f \end{cases}$$
 (8)

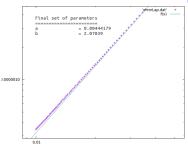


Figure – Comparaison de l'erreur en norme 2 suivant dx avec $f(x) = ax^b$ (échelle log-log)

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

roduction

Modélisation mathématique

ésolution

Code Algorithme d'Uzawa

Cas tests

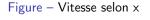
Résolution du Laplacien

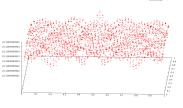
Cas Test 0
Cas Test 1
Poiseuille
Erreurs et ordres
Ker(B^T)

Influence

Cas Test 0a

$$\left\{ \begin{array}{l} U=(u_1,u_2) \text{ avec } u_1,u_2 \text{ r\'eels} \\ F=(0,0) \end{array} \right.$$





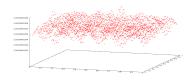


Figure – Vitesse selon y

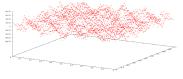


Figure - Pression

Résolution de l'équation de Stokes en schéma MAC. par la méthode d'Uzawa et solveur itératif. en Fortran90

Pierre Jacquet, Guillaume Ravel

Schéma MAC

Code Algorithme

d'Uzawa

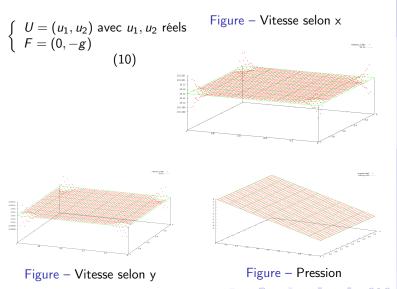
Résolution du Laplacien

Cas Test 0

Cas Test 1 Poiseuille

Erreurs et ordres $Ker(B^T)$

Cas Test 0b



Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modelisation mathématique

Schéma MAC

lésolution umérique

Code Algorithme d'Uzawa

d Uzawa

Cas te

Résolution du Laplacien

Cas Test 0

Cas Test 1
Poiseuille
Erreurs et ordres

 $Ker(B^T)$ Influence

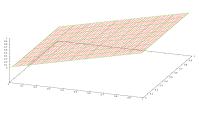
onclusion

4 U P 4 DF P 4 E P 4 E P E *) 4 (*

Cas Test 1a

$$\begin{cases}
U = (x, -y) \\
F = (1, 1) \\
G = H = U \\
P = x + y + c
\end{cases}$$
(11)

Figure – Vitesse selon x



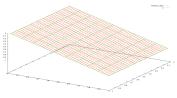


Figure – Vitesse selon y

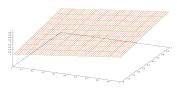


Figure – Pression

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

ésolution umérique

Code Algorithme d'Uzawa

Cas tests

Résolution du

Laplacien

Cas Test 1

Poiseuille

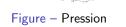
Erreurs et ordres $Ker(B^T)$

Influence

Cas Test 1b

$$\begin{cases}
U = (x, y) \\
F = (1, 1) \\
G = H = U \\
P = x + y + c
\end{cases}$$
(12)

Figure - Vitesse selon y



Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

ésolution umérique

Code Algorithme d'Uzawa

Cas tests

.

Résolution du Laplacien

Cas Test 0

Cas Test 1

Poiseuille Erreurs et ordres

Erreurs et ord Ker(B^T)

Influence

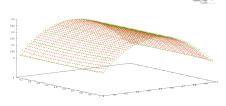
onclusion

◆ロ → ◆昼 → ◆ き → ● ● りへで

Poiseuille

$$\begin{cases}
U = (L_y^2(1 - \frac{y}{L_y})\frac{y}{L_y}, 0) \\
F = (0, 0) \\
G = (1, 0) \text{ et } H = (0, 0) \\
P = -2x + c
\end{cases}$$
(13)

Figure - Vitesse selon x



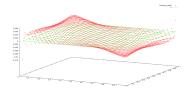


Figure – Vitesse selon y

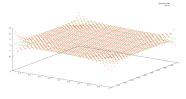


Figure - Pression

Résolution de l'équation de Stokes en schéma MAC. par la méthode d'Uzawa et solveur itératif. en Fortran90

Pierre Jacquet, Guillaume Ravel

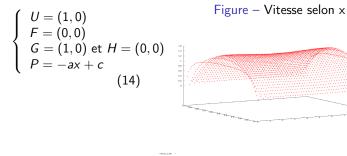
Schéma MAC

Code Algorithme

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille

Erreurs et ordres $Ker(B^T)$

Génération d'écoulement de Poiseuille



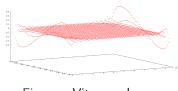


Figure – Vitesse selon y

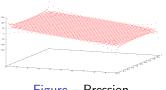


Figure - Pression

Résolution de l'équation de Stokes en schéma MAC. par la méthode d'Uzawa et solveur itératif. en Fortran90

Pierre Jacquet, Guillaume Ravel

Schéma MAC

Code Algorithme

Résolution du

Laplacien Cas Test 0 Cas Test 1

Poiseuille Erreurs et ordres

 $Ker(B^T)$

Erreurs et ordres

Figure – TC0 : Erreur=f(dx)

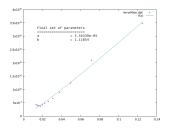


Figure - TC1a : Erreur=f(dx)

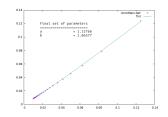
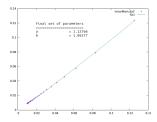
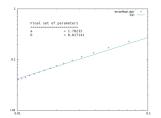


Figure – TC1b : Erreur=f(dx) Figure – Poiseuille : Erreur=f(dx)





Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique Schéma MAC

o Caralandan

numérique Code

Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille

Erreurs et ordres $Ker(B^T)$

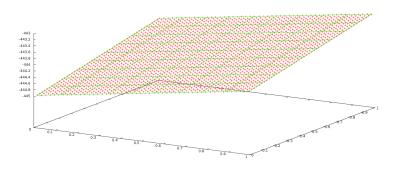
Influence de

Conclusion

16/23

$Ker(B^T)$ Observation Cas TC1b





Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

Résolution

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille

Erreurs et ordres

 $Ker(B^T)$

$Ker(B^T)$

Introduction à la fonction Moy_0

```
FUNCTION Moy 0(V)
  IMPLICIT NONE
  REAL*8, DIMENSION(:), INTENT(IN)::V
  REAL*8, DIMENSION(size(V)):: Moy 0
  REAL*8::moy
  INTEGER::i
  moy=SUM(V)/size(V)
  DO i=1,size(V)
    Mov O(i)=V(i)-mov
  END DO
END FUNCTION
```

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

Introduction

Modélisation mathématique

Résolution

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres

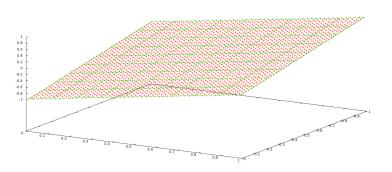
 $Ker(B^T)$

. . .

◆ロト ◆部ト ◆差ト ◆差ト 差 めので

$Ker(B^T)$ Après Moy_0

oression.dat' + x+y-1 ----



Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

itroduction

Modélisation mathématique

Schéma MAC

Résolution Jumérique

Code Algorithme d'Uzawa

Cas tests

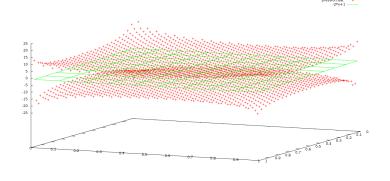
Résolution du Laplacien Cas Test 0 Cas Test 1

Poiseuille Erreurs et ordres

 $Ker(B^T)$



$Ker(B^T)$ Poiseuille



Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

Résolution numérique

Code Algorithme d'Uzawa

Cas tes

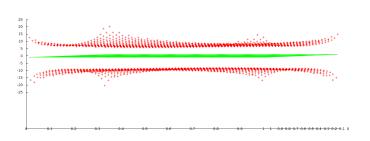
Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres

Ker(B^T)

. . .

$Ker(B^T)$ Poiseuille





Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

lésolution

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille

Erreurs et ordres $Ker(B^T)$

minderide de



Influence de μ

μ	Cas test 0b		Cas test 1a	
-	Nb Iter	Résidu	Nb Iter	Résidu
1	749	10-4	223	10-4
2	1000	2.10-3	1000	3.10-4
3	-	NaN	-	NaN
0.5	970	10-4	292	10-4
0.1	1000	2.10-3	592	10-4
1.5	641	10-4	191	10-4

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique Schéma MAC

Résolution numérique

Code Algorithme d'Uzawa

Cas tests

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres $Ker(B^T)$

Influence de μ



Conclusion

- ► Implémentation d'Uzawa et du gradient conjugué
- Gradient conjugué validé sur un problème de Poisson
- Résultats et cas tests probants
- ▶ Analyse des phénomènes issus de Ker(B^T)
- ightharpoonup Etude sur le critère μ de l'aglorithme d'Uzawa

Résolution de l'équation de Stokes en schéma MAC, par la méthode d'Uzawa et solveur itératif, en Fortran90

Pierre Jacquet, Guillaume Ravel

ntroduction

Modélisation mathématique

Schéma MAC

Résolution numérique Code

Algorithme d'Uzawa

Cas te

Résolution du Laplacien Cas Test 0 Cas Test 1 Poiseuille Erreurs et ordres Ker(B^T)

Conclusion

