

Travail en collaboration avec **C. Calgario** (Lille 1) et **Y. Saad** (U. Minnesota, Minneapolis, USA),

Enrichissements de factorisations LU incomplètes et applications

J-P. CHEHAB¹

¹LAMFA, UMR 7352, Université de Picardie Jules Verne, Amiens, France (jean-paul.chehab@u-picardie.fr)

ONERA Châtillon, le 29 avril 2014

- 1 Motivation
- 2 Formulation du problème
(*Enrichissement d'une factorisation LU incomplète*)
- 3 Méthodes itératives
 - Techniques de type "target matrix"
 - Itérations alternées
 - Flots de matrices
- 4 Mise en œuvre
- 5 Résultats numériques
- 6 Conclusion et perspectives

Résolution efficace de suites de problèmes linéaires

$$A_k x_k = b_k, k = 0, \dots$$

où les matrices A_k varient légèrement d'une itération à l'autre.

Une factorisation (LU, Cholesky), même précise, de A_0 peut ne plus être efficace pour A_k , $k \gg 1$. Il faut donc actualiser cette factorisation, sans avoir à tout recalculer !

- Schémas (semi)implicites pour les problèmes d'évolution
- Résolution de systèmes d'équations non linéaires par des méthodes de type Newton (actualisation de la factorisation de la jacobienne)

On désire améliorer la factorisation LU approchée

$$A = LU + R$$

L'idéal est de trouver des matrices X_L et X_U telles que

$$A = (L + X_L)(U + X_U) = LU + X_L U + L X_U + X_L X_U$$

$$R = X_L U + L X_U + X_L X_U$$

équ. de Riccati algébrique! approx. linéaire

$$R \simeq X_L U + L X_U$$

Le problème

Trouver X_L et X_U resp. triangulaire inférieure, supérieure, avec $(X_L)_{ii} = 1$ telles que

$$R = X_L U + L X_U$$

Rem On peut considérer Cholesky avec $X_L = (X_U)^T = X \rightarrow$ Eq de Lyapounov

Notations

- Parties triangulaires

X_{Δ} et X_{∇} désignent la *partie triangulaire inférieure stricte* et *partie triangulaire supérieure* de X ; $X_{\Delta} = Id + X_{\nabla}$

$$(X + Y)_{\Delta} = X_{\Delta} + Y_{\Delta}; \quad (X + Y)_{\nabla} = (X + Y)_{\nabla} \quad (1)$$

$$(X_{\Delta} Y_{\Delta})_{\nabla} = 0; \quad (X_{\nabla} Y_{\nabla})_{\Delta} = 0; \quad (2)$$

$$(X_{\nabla} Y_{\nabla})_{\Delta} = 0. \quad (3)$$

- Normes, produits scalaires

$\langle X, Y \rangle = \text{Tr}(Y^T X)$ La norme de Frobenius $\|\cdot\|_F$ est associé au pdt scalaire matriciel $\|X\|_F = [\text{Tr}(X^T X)]^{1/2}$

Méthodes basées sur les inverses approchées

Départ : calcul approché de l'inverse de A par minimisation de

$$F(M) = \|I - AM\|_F^2$$

Offre la possibilité de calculer des **inverses approchées creuses** : sparse Matrix correction en utilisant une méthode de descente. On cherche des solutions creuses approchées de

$$Am_j = e_j$$

Peu onéreux en termes de calculs et conjugable avec des techniques de seuillage de coefficients. La résolution s'effectue par une méthode de gradient :

$$M_{new} = M + \alpha G$$

avec $\alpha = \frac{\langle R, AG \rangle}{\langle AG, AG \rangle}$, $R = I - AM$, $G = -2A^T R$

Méthodes de descente alternées pour LU

On cherche à minimiser alternativement

$$\|A - (L + X_L)U\|_F \text{ et } \|A - L(U + X_U)\|_F$$

On définit

$$F(X_U) = \|A - L(U + X_U)\|_F^2 = \|R - LX_U\|_F^2 = \|R\|_F^2 - 2\text{Tr}[R^T LX_U] + \|LX_U\|_F^2$$

Et $X_U = X_U + \alpha_U G_U$, $\alpha_U = \frac{\langle R, LG_U \rangle}{\langle LG_U, LG_U \rangle}$, $G_U = L^T R$ Pour avoir une suite de matrices triangulaires sup.

$$G = [L^T R]_{\downarrow}$$

Lemma

Toute version creuse de G est une direction de descente.

De même $X_L = X_L + \alpha_L G_L$, $\alpha_L = \frac{\langle R, G_L U \rangle}{\langle G_L U, G_L U \rangle}$, $G_L = R U^T$

MERLU, Minimal Energy Residual descent for LU

1. Select an initial pair L, U (with $L = L_{\setminus}$, $U = U_{\setminus}$)
2. Until convergence Do
3. Compute $R := A - LU$
4. Compute $G = [L^T R]_{\setminus}$
5. Apply numerical dropping to G
6. Compute $\alpha = \langle R, C \rangle / \|C\|_F^2$, where $C = LG$.
7. Compute $U := U + \alpha G$
8. Compute $R := A - LU$
9. Compute $G := [RU^T]_{\setminus}$
10. Apply numerical dropping to G
11. Compute $\alpha = \langle R, C \rangle / \|C\|_F^2$ where $C = GU$.
12. Compute $L := L + \alpha G$
13. EndDo

Itérations creuses alternées

$$X_L U = R \rightarrow X_L = R U^{-1}$$

$$U_{k+1} = U_k + [L_k^{-1}(A - L_k U_k)]_{\nabla} \quad (4)$$

$$L_{k+1} = L_k + [(A - L_k U_{k+1})U_{k+1}^{-1}]_{\Delta}. \quad (5)$$

Les équations (4–5) se simplifient.

$$U_{k+1} = U_k + [L_k^{-1}(A - L_k U_k)]_{\nabla} = [U_k + L_k^{-1}(A - L_k U_k)]_{\nabla} = [L_k^{-1}A]_{\nabla}$$

Idem pour L_{k+1} , on en tire la méthode itérative

$$U_{k+1} = [L_k^{-1}A]_{\nabla} \quad (6)$$

$$L_{k+1} = [A U_{k+1}^{-1}]_{\Delta}. \quad (7)$$

Les parties triangulaires sup. et inf. peuvent être seuillées (*numerical dropping*).

Exemple

$$A = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \rightarrow L_0 = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/4 & 1/4 & 1 \end{pmatrix}, U_0 = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 4 & 1 \\ 0 & 0 & 4 \end{pmatrix}$$

$$\rightarrow L_0 U_0 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 17/4 & 5/4 \\ 1 & 5/4 & 9/2 \end{pmatrix}$$

$$L_0^{-1} A = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 15/4 & 3/4 \\ 0 & -3/16 & 57/16 \end{pmatrix} \rightarrow U_1 = \begin{pmatrix} 4 & 1 & 1 \\ 0 & 15/4 & 3/4 \\ 0 & 0 & 57/16 \end{pmatrix}.$$

$$AU_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/4 & 1/5 & 95/96 \end{pmatrix} \rightarrow L_1 = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/4 & 1/5 & 1 \end{pmatrix}$$

$$L_1 U_1 = \begin{pmatrix} 4 & 1 & 1 \\ 1 & 4 & 1 \\ 1 & 1 & 317/80 \end{pmatrix}$$

ITALU, Iterative Threshold Alternating Lower-Upper correction

1. Given : A, U_0, L_0 (with $U_0 = [U_0]_{\nabla}; L_0 = [L_0]_{\blacktriangleleft}$)
2. For $k = 1, \dots$, Do :
3. Compute $R_k = A - L_k U_k$
4. Compute $X_U = [L_k^{-1} R_k]_{\nabla}$
5. Apply numerical dropping to X_U
6. $U_{k+1} = U_k + X_U$
7. If $\det(U_{k+1}) == 0$ Abort "Singular U reached"
8. Compute $R_{k+1/2} = A - L_k U_{k+1}$
9. Compute $X_L = [R_{k+1/2} U_{k+1}^{-1}]_{\blacktriangleleft}$
10. Apply numerical dropping to X_L
11. $L_{k+1} = L_k + X_L$
12. EndDo

Lemmes de filtrage

Lemma

Soient X une matrice carrée, P une matrice triangulaire inférieure $P_{ii} = 1$ et Q une matrice triangulaire supérieure. Alors

$$[PX]_{\nabla} = [PX]_{\nabla} ; \quad [XQ]_{\Delta} = [X]_{\Delta} Q_{\Delta} .$$

Lemma

Avec les mêmes hypothèses et Q inversible, on a pour toute matrice M carrée

$$[M]_{\Delta} = [[MQ^{-1}]_{\Delta} Q]_{\Delta} \quad \text{et} \quad [M]_{\nabla} = [P[P^{-1}M]_{\nabla}]_{\nabla} .$$

Soit A une matrice admettant une décomposition LU $A = LU$, avec $L_{ii} = 1$. Alors les suites U_k et L_k définies par 6 et 7 en partant de L_0 triangulaire inférieure avec $(L_0)_{ii} = 1$, satisfont les propriétés suivantes :

- (i) $[A - L_k U_{k+1}]_{\nabla} = 0$, $[A - L_{k+1} U_{k+1}]_{\Delta} = 0$, $k \geq 0$.
- (ii) Si L_k et U_k convergent, alors $\lim_{k \rightarrow +\infty} L_k = L$ et $\lim_{k \rightarrow +\infty} U_k = U$.
- (iii) L'algorithme converge en une étape si $L_0 = L$.

Theorem

Soit L_0 donné tel que les suites L_k, U_k sont définies pour $k = 1, 2, \dots, n$. Alors U_k et L_k convergent vers U et L respectivement, en au plus n étapes.

Cas des filtres creux

On note \mathcal{F} la matrice filtre et \odot le produit d'Hadamard. On a

$$U_{k+1} = [(L_k^{-1}A) \odot \mathcal{F}]_{\nabla} = [(L_k^{-1}A)]_{\nabla} \odot \mathcal{F} \quad (8)$$

$$L_{k+1} = [(AU_{k+1}^{-1}) \odot \mathcal{F}]_{\blacktriangleleft} = [(AU_{k+1}^{-1})]_{\blacktriangleleft} \odot \mathcal{F}. \quad (9)$$

Corollary

Supposons que chaque matrice filtre \mathcal{F}_k , $k \geq 0$ contienne le "cannevas de creusitude" de L et de U . Soit L_0 tel que $L_0 \odot \mathcal{F}_0 = L_0$, on suppose que L_k et U_k existent pour $k = 1, \dots, n$. Alors les suites U_k et L_k convergent vers U et L , respectivement, en au plus n étapes.

Supposons que le filtre \mathcal{F} soit constant. Soient $S_k = U_k - \mathcal{F} \odot U_k$ et $T_k = L_k - \mathcal{F} \odot L_k$. Supposons que L_k et U_k convergent vers \bar{L} , \bar{U} , respectivement et que \bar{U} est inversible. Alors

- (i) La suite S_k converge vers $S = [\bar{L}^{-1}A]_{\nabla} - \bar{U}$.
- (ii) La suite T_k converge vers $T = [A\bar{U}^{-1}]_{\Delta} - \bar{L}$.
- (iii) Les suites $(A - L_k U_{k+1})_{\Delta}$ et $(A - L_k U_k)_{\nabla}$ convergent vers $[\bar{L}S]_{\nabla}$ et $[T\bar{U}]_{\Delta}$, respectivement.

Flots de factorisations de matrices

Obtention du flot de Cholesky

Soit A SDP, $A = S^T S$. On définit

$$\frac{dX}{dt} = S - X, \text{ ainsi } \frac{dX}{dt} = (S^T)^{-1} (S^T S - S^T X)$$

Si X est assez proche de S alors

$$\frac{dX}{dt} \simeq (X^T)^{-1} (A - X^T X)$$

$$\text{On définit } X \text{ par } \frac{dX}{dt} = [(X^T)^{-1} (A - X^T X)]_{\nabla}$$

Schéma en temps

Euler explicite ($\Delta t = 0.8$)

Obtention du flot LU

$$\begin{cases} \frac{dP}{dt} = [(A - PQ)Q^{-1}]_{\nabla}, & t > 0 \\ P_{i,i}(t) = 1, \\ P(0) = P_0 \end{cases}$$

et

$$\begin{cases} \frac{dQ}{dt} = [P^{-1}(A - PQ)]_{\Delta}, & t > 0 \\ Q(0) = Q_0 \end{cases}$$

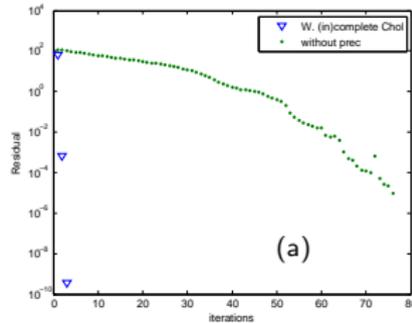
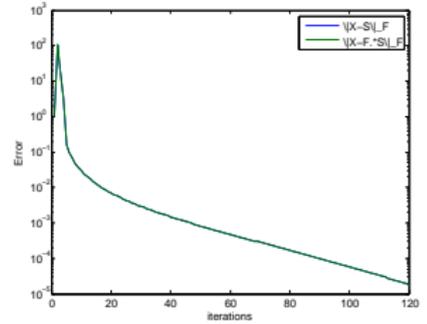
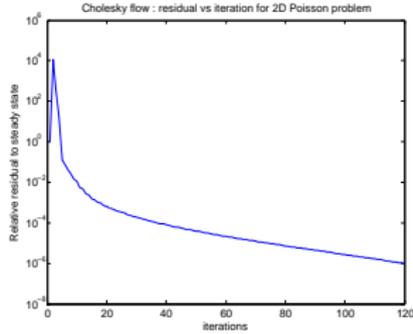
Schéma en temps

Euler explicite ($\Delta t = 0.8$) ou bien

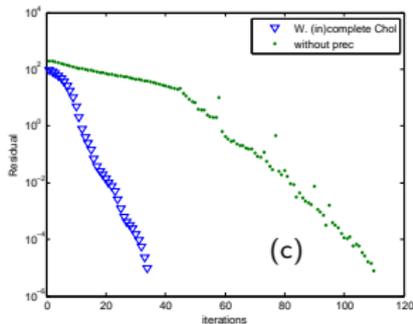
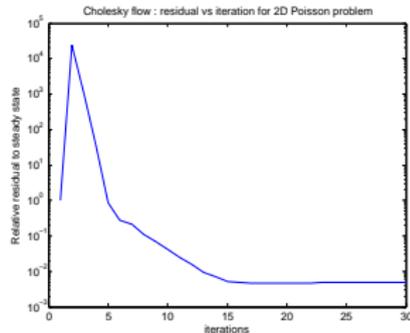
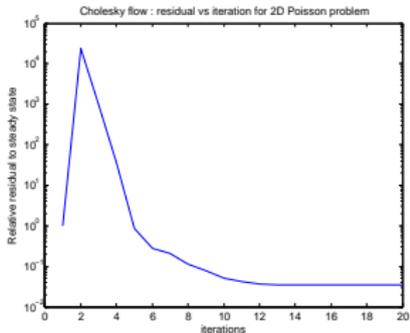
$$\begin{aligned} Q^{k+1} &= Q^k + \Delta t [(P^k)^{-1} A]_{\nabla} \\ P^{k+1} &= P^k + \Delta t [A(Q^{k+1})^{-1}]_{\Delta} \end{aligned}$$

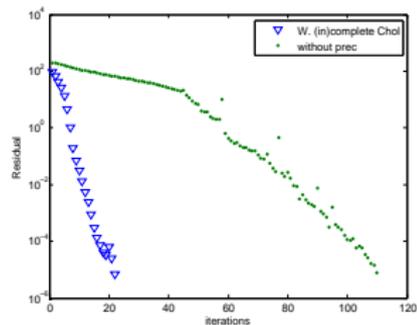
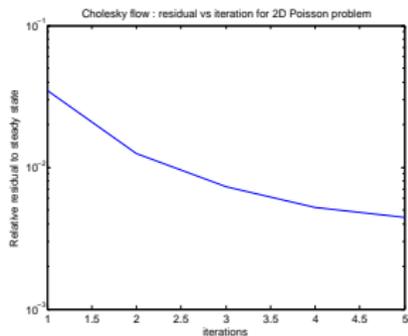
c'est ITALU avec $\Delta t = 1$

Flot de Cholesky / Pb de Poisson 2D

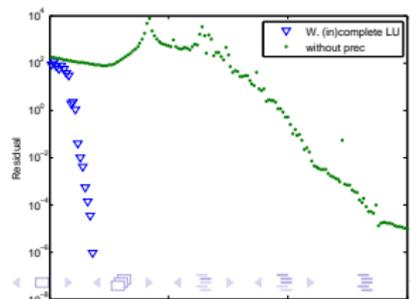
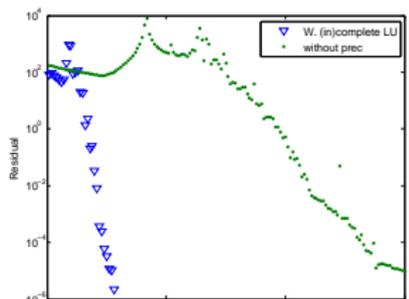
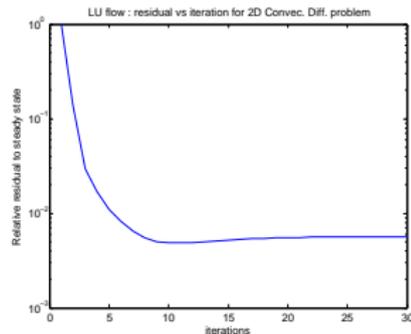
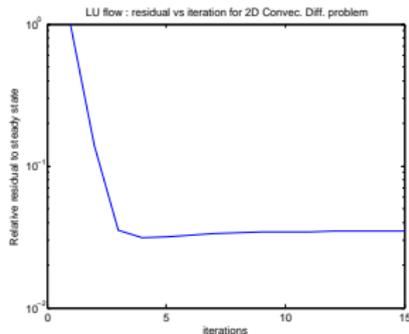


Flot de Cholesky / Pb de Poisson 2D





Flot LU / Pb convection diffusion $-\Delta u + 20u_x + 10u_y = f$



(a)

Ingrédients pour la mise en œuvre

Deux ingrédients principaux

- Produits de tableaux creux (Chow-Saad, SIAM JSC 1998)
- Résolutions de systèmes triangulaires creux avec seconds membre creux (Gilbert-Peierls, Tinney-Brandwajn-Chan, Chan-Brandwajn); DAG techniques; approximate solutions of such systems (thresholding)
- Simplified iterations : basically replace the matrices by their diagonal part in the implicit parts of ITALU

On comparera avec d'autres factorisations incomplètes

- Le remplissage (*fill-in*)
- L'efficacité du préconditionneur obtenu (itérations)
- La robustesse

Particularité : on pourra faire varier (enrichir) le préconditionneur au cours des itérations (avec FGMRES par exemple)

Problèmes linéaires

Convection-Diffusion 3D + shift (matrice indéfinie) et Matrices de Harwell Boeing

- Avec ITALU et GMRES(30)
 - SSOR $\omega = 1$ $L_0 = I + A_{\setminus} D_A^{-1}$, $U_0 = A_{\nabla}$
 - $A \simeq L_1 U_1$ (avec 1 itération de ITALU) : ITALU(1)
 - ITALU(2)
 - ITALU(3)
- Avec MERLU et GMRES(30) (drop et lfil variant)
 - SSOR $\omega = 1$ $L_0 = I + A_{\setminus} D_A^{-1}$, $U_0 = A_{\nabla}$
 - $A \simeq L_1 U_1$ (avec 1 itération de ITALU) : ITALU(1)
 - ITALU(2)
 - ITALU(3)

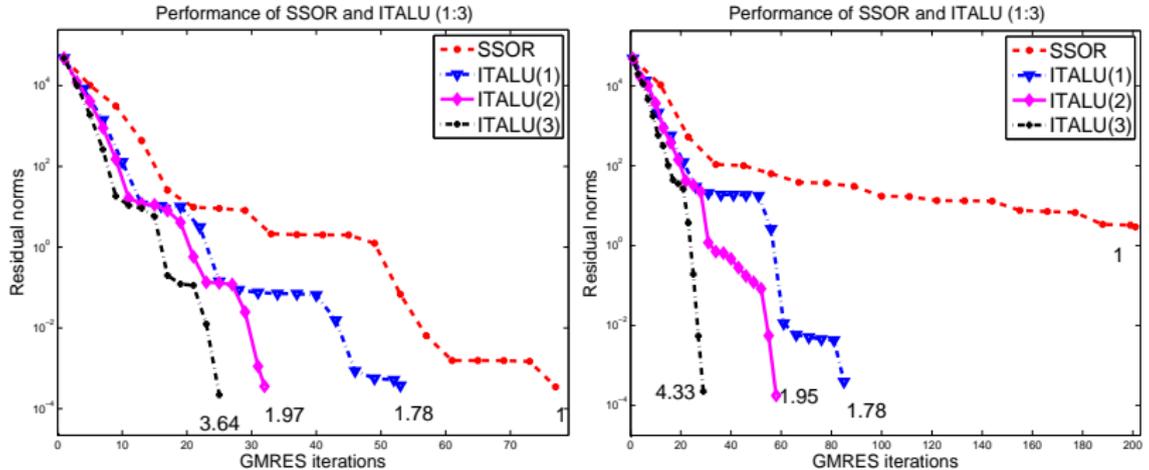


Figure : Performance of GMRES(30) for a convection-diffusion problem with the SSOR and 3 levels of improvements of the ITALU algorithm. Left and right plot are for two different problems.

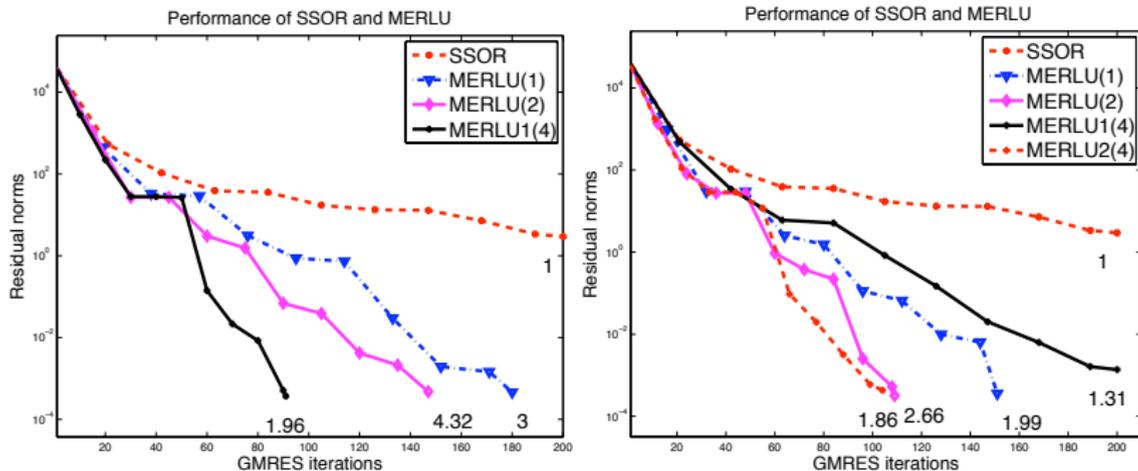


Figure : Performance of GMRES(30) for a convection-diffusion problem with the SSOR and a few levels of improvements using the Frobenius-based approach. The test matrix is the same as the second one of the previous section. The right plot shows tests with different dropping strategies.

Factorisation enrichie au cours des itérations de GMRES

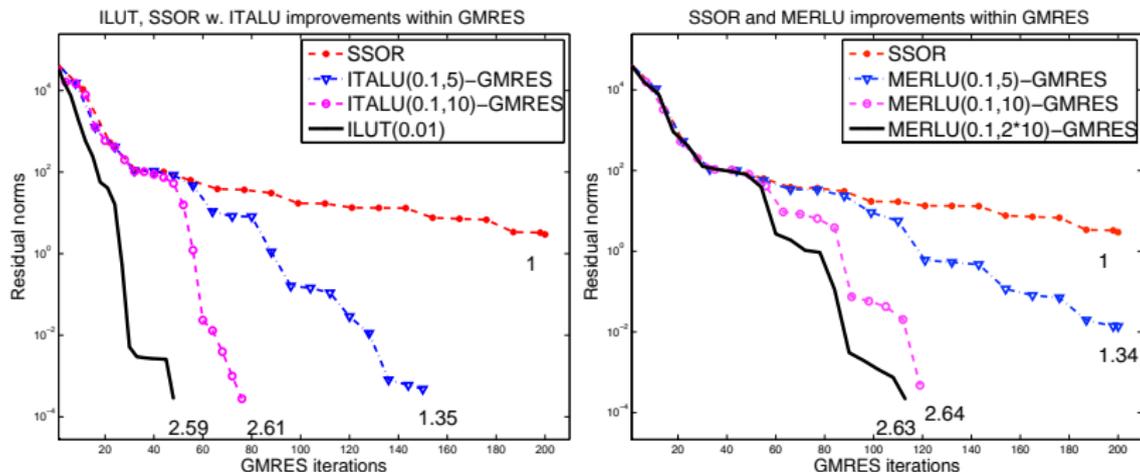


Figure : Performance of GMRES(30) for a convection-diffusion problem with the SSOR and ILUT preconditioners and incremental ILU procedures implemented within GMRES with 2 different parameters. Left and right plot are for the alternating LU correction and the Frobenius approach.

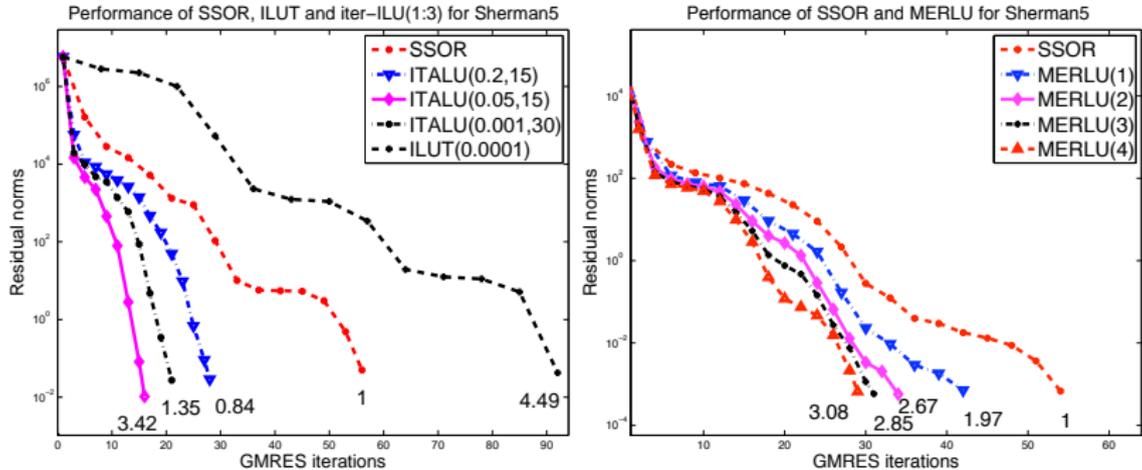


Figure : Performance of GMRES(30) for Sherman-5 problem with the SSOR and ILUT preconditioners and incremental ILU procedures. The left side plot uses the alternating lower-upper correction and the right one the Frobenius approach.

Un problème non linéaire évolutif : Navier-Stokes à densité variable

Soit $\Omega \subset \mathbb{R}^2$.

$$\partial_t \rho + \operatorname{div}_{\mathbf{x}}(\rho \mathbf{u}) = 0, \quad (10)$$

$$\rho(\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla_{\mathbf{x}}) \mathbf{u}) + \nabla_{\mathbf{x}} p - \mu \Delta_{\mathbf{x}} \mathbf{u} = \mathbf{f}, \quad (11)$$

$$\operatorname{div}_{\mathbf{x}} \mathbf{u} = 0. \quad (12)$$

Ici $\rho(t, \mathbf{x}) \geq 0$ est la densité du fluide de vitesse $\mathbf{u}(t, \mathbf{x}) \in \mathbb{R}^2$, $\mathbf{f}(t, \mathbf{x})$ est la force extérieure et $\mu > 0$ la viscosité, $p(t, \mathbf{x}) \in \mathbb{R}$ la pression (Calgaro-Creusé-Goudon, J.Comp.Phys. 2008)

Discrétisation en temps

$$\begin{cases} \rho^* \left(\frac{3\mathbf{u}^{n+1} - 4\mathbf{u}^n + \mathbf{u}^{n-1}}{2\Delta t} \right) + (\bar{\mathbf{u}}^{n+1} \cdot \nabla_{\mathbf{x}})\mathbf{u}^{n+1} - \mu \Delta_{\mathbf{x}}\mathbf{u}^{n+1} + \nabla_{\mathbf{x}}\rho^{n+1} & = \mathbf{f}^{n+1}, \\ \operatorname{div}_{\mathbf{x}}\mathbf{u}^{n+1} & = 0. \end{cases}$$

Ici $\bar{\mathbf{u}}^{n+1} = 2\mathbf{u}^n - \mathbf{u}^{n-1}$ et $\rho^* = \rho^n$ ou ρ^{n+1}

Discrétisation en espace : Pb de point selle

$$\begin{pmatrix} \mathbf{A}_{n+1} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix},$$

Avec

$$\mathbf{A}_{n+1} = \frac{3}{2\Delta t}M + \frac{1}{Re}L + D$$

Le préconditionneur choisi est de la forme

$$\begin{pmatrix} \hat{\mathbf{A}} & \mathbf{B}^T \\ \mathbf{0} & \hat{\mathbf{S}} \end{pmatrix} \text{ ou bien } \begin{pmatrix} \hat{\mathbf{A}} & \mathbf{0} \\ \mathbf{B} & \hat{\mathbf{S}} \end{pmatrix}$$

où

- $\hat{\mathbf{A}}$ est un préconditionneur de \mathbf{A}_{n+1}
- $\hat{\mathbf{S}}$ est une approximation du complément de Schur $\mathbf{B}\mathbf{A}_{n+1}^{-1}\mathbf{B}^T$.

On utilise FGMRES avec préconditionnement à droite

$$\begin{pmatrix} \mathbf{A}_{n+1} & \mathbf{B}^T \\ \mathbf{B} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{A}} & \mathbf{B}^T \\ \mathbf{0} & \hat{\mathbf{S}} \end{pmatrix}^{-1} \begin{pmatrix} W_U \\ W_P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}$$

$$\text{Avec } \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} \hat{\mathbf{A}} & \mathbf{B}^T \\ \mathbf{0} & \hat{\mathbf{S}} \end{pmatrix}^{-1} \begin{pmatrix} W_U \\ W_P \end{pmatrix}$$

Approximation du complément de Schur $\mathbf{S} = \mathbf{B}\mathbf{A}_{n+1}^{-1}\mathbf{B}^T$

- Masse de Pression :

$$\hat{\mathbf{S}}^{-1} = \mathbf{M}_p^{-1}$$

- Cahouet-Chabard :

$$\hat{\mathbf{S}}^{-1} = \frac{2\Delta t}{3}\mathbf{M}_p^{-1} + \mathit{Re}L_p^{-1}$$

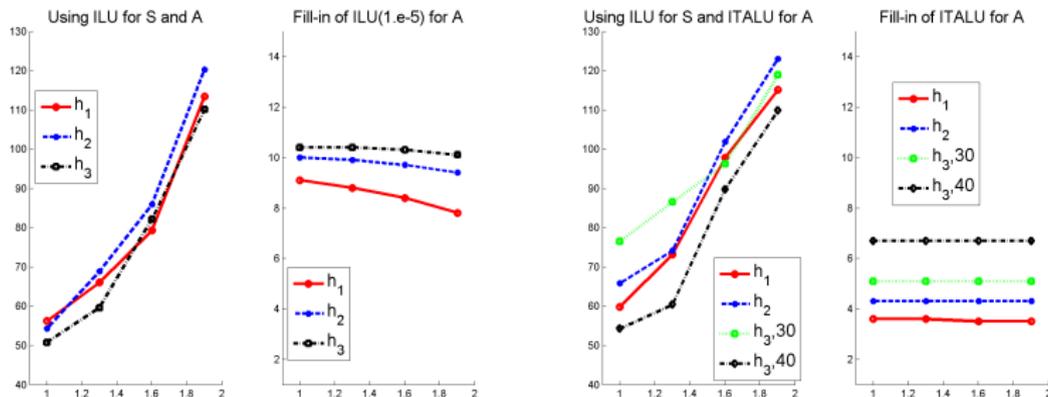


Figure : Mean number of iterations required by FGMRES(30) for the saddle point problem using pressure-mass preconditioner for the Schur complement, for different values of h and $Re = 1$.

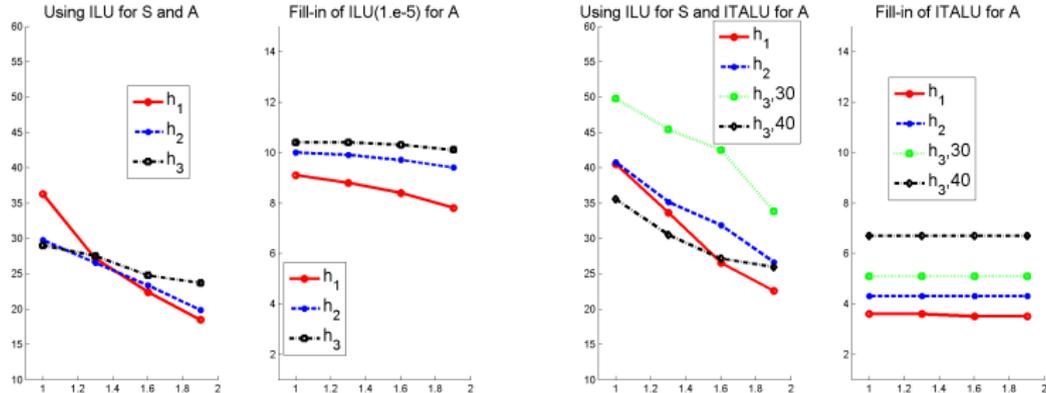


Figure : Mean number of iterations required by FGMRES(30) for the saddle point problem using the Cahouet-Chabard preconditioner for the Schur complement, for different values of h and $Re = 1$.

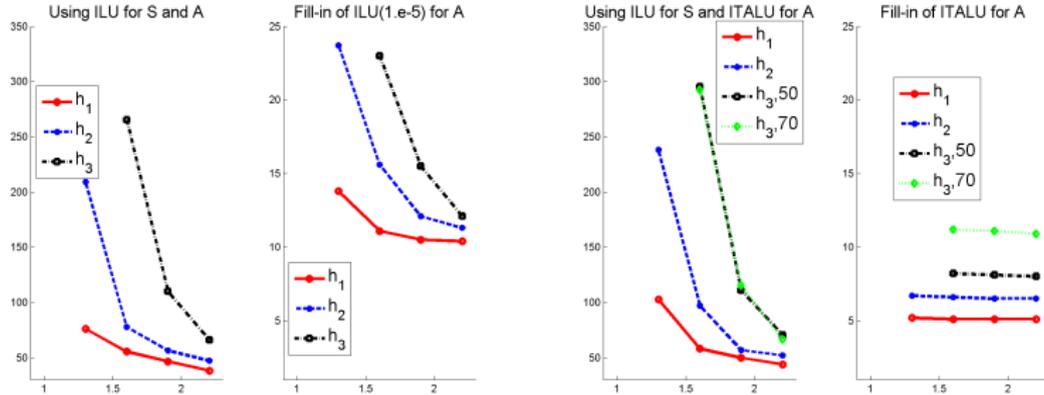


Figure : Mean number of iterations required by FGMRES(30) for the saddle point problem using the Cahouet-Chabard preconditioner for the Schur complement, for different values of h and $Re = 1000$.

Conclusion et perspectives

- Approches pratiques pour l'enrichissement de préconditionneurs
 - Littérature très vaste pour LU, peu de travaux pour les enrichissements
 - Target matrix (MERLU), Itérations alternées (ITALU, flots)
 - Méthodes simples, résultats de convergence
 - Semble robuste et intéressant (fill in réduit, convergence comparable)
- Reste la mise en œuvre pour problèmes de grande échelle. L'efficacité (pour ITALU) repose sur les techniques de résolution creuse de systèmes creux avec second membre creux (!) mais cela existe déjà (travail en cours).
- Cas non-linéaire (quasi-Newton), S. Bellavia *et al*

Références

Approx. ILU incremental updates C. Calgari, J.P. Chehab, Y.Saad, Incremental incomplete ILU factorizations with applications, Numerical Linear Algebra with Applications, 17 (2010), pp. 811-837.

Approx. Inverse by LS M. Benzi, M. Tuma, A sparse approximate inverse preconditioner for nonsymmetric linear systems, SIAM J. Sci. Comput., 19 (1998), pp. 968-994.

Nonlinear, NSE, $Re < 400$ S. Bellavia, D. Bertaccini, B. Morini, Nonsymmetric preconditioner updates in Newton- Krylov methods for nonlinear systems, SIAM J. Sci. Comput., 33 (2011), pp. 2595-2619.

Survey Talk : E. De Sturler

http://people.ma.ths.ox.ac.uk/ekert1/PRECON13/talks/Talk_deSturler.pdf