

HIPS : un solveur hybride direct-itératif basé sur une approche complément de Schur

Groupe de travail

Jérémie GAIDAMOUR, Pascal HÉNON, Jean ROMAN



21 novembre 2006

Introduction

Résolution de systèmes linéaires creux $A.x = b$ à plusieurs dizaines de millions d'inconnues.

Méthode Directe (factorisation exacte)

- Exhibe une structure dense par blocs (BLAS 3)
- Solution très précise ($\approx 10^{-15}$)
- Consomme beaucoup de mémoire (ne passe pas à l'échelle en mémoire sur de grands problèmes 3D)

Méthode de Krylov + ILU ou ICC

- Robustesse dépend du remplissage admis dans les facteurs
- Basée sur une implémentation scalaire (ILUT ou ILU(k))
- Précision moins élevée que les méthodes directes

⇒ on voudrait un compromis : un solveur permettant de résoudre des cas difficiles et/ou avec une bonne précision

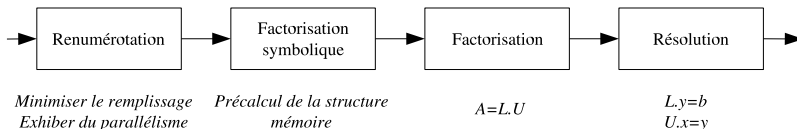
Plan

- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

Méthode directe (1/2)

Factorisation exacte de A suivie d'une résolution triangulaire.

Chaîne de résolution :



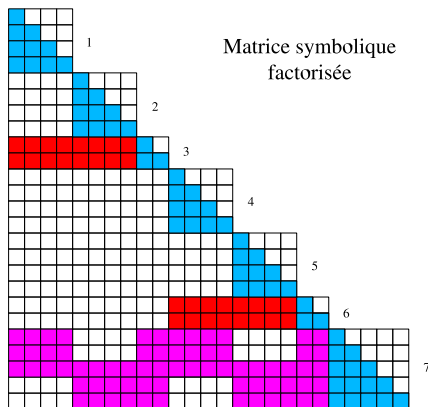
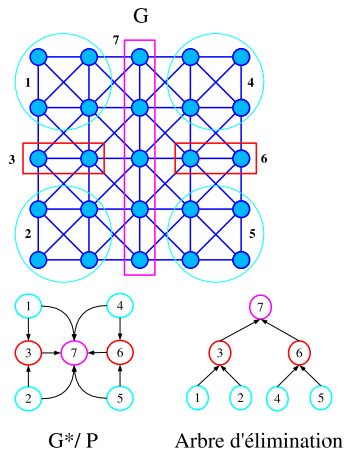
Outils d'analyse :

- Graphe G associé à A : arête $(i, j) \Leftrightarrow a_{ij} \neq 0$
- Graphe d'élimination G^* associé à L

Factorisation symbolique par blocs :

- Partition P des inconnues
- Graphe quotient $G^*/P \Rightarrow (G/P)^* = G^*/P$

Méthode directe (2/2)



Méthode itérative préconditionnée

Méthode de Krylov :

Calculer successivement des approximations de la solution jusqu'à une précision voulue (ex : dans Krylov, l'espace de recherche est augmenté à chaque itération).

La convergence dépend de $\|A\|_p \|A^{-1}\|_p$

Méthode de préconditionnement : On transforme le système à l'aide d'un opérateur linéaire M tel que le nouveau système ait un meilleur conditionnement.

Préconditionnement

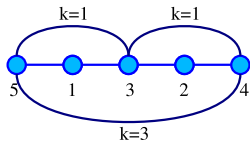
Plusieurs variantes sont possibles :

- précond. à gauche : $M^{-1}.A.x = M^{-1}.b$;
 - précond. à droite : $A.M^{-1}u = b$, $x = M^{-1}.u$;
 - précond. découpé (cas sym.) : $M^{-1}.A.M^{-t}.u = M^{-1}.b$,
 $x = M^{-1}.u$;
- En règle générale, on ne stocke pas $M^{-1}.A$ sous forme matricielle on utilise $(M^{-1}.A).x = M^{-1}.(A.x)$. Pour que ce soit rentable, il faut que $M^{-1}.y$ soit peu coûteux à calculer.
 - Usuellement, on cherche M proche de A (i.e. $M^{-1} \approx A^{-1}$).
 - On va s'intéresser au préconditionneur utilisant une technique de **factorisation incomplète** : $M^{-1} \approx U^{-1}.L^{-1}$.

Préconditionnement par factorisation incomplète

Exemple de factorisation incomplète :

- **ILUT** : dropping numérique
- **ILU(k)** : schéma de remplissage basé sur la longueur des chemins d'élimination



Préconditionnement par factorisation incomplète

Problèmes :

- Algorithmes “scalaires” pas très performant malgré un faible coût théorique.
- Factorisation symbolique impossible en ILUT, de complexité du même ordre que la factorisation numérique pour $ILU(k)$ (et aussi problématique à paralléliser).
- Parallélisation difficile (sauf $ILU(0)$).

Bilan

Ce qu'il faut pour être performant :

- Fournir des algorithmes par blocs qui permettent de mieux exploiter les calculateurs hautes performances.
- Maîtriser le remplissage
- Exhiber du parallélisme.

On a besoin :

- d'une numérotation adaptée.
 - d'un partitionnement ($\Rightarrow$ bloc dense = BLAS).
 - d'une phase d'analyse qui permet de bien gérer le parallélisme.
- $\Rightarrow$ Mélanger méthodes directes et itératives
- $\Rightarrow$ Chercher un compromis en fonction de la difficulté numérique

Plan

- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

Généralités sur le complément de Schur (1/2)

En décomposant le système linéaire $A.x = b$:

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} \cdot \begin{pmatrix} x_B \\ x_C \end{pmatrix} = \begin{pmatrix} y_B \\ y_C \end{pmatrix} \quad (1)$$

On a :

$$\begin{cases} B.x_B + F.x_C = y_B \\ E.x_B + C.x_C = y_C \end{cases} \Rightarrow \mathbf{x_B} = B^{-1}(y_B - F.x_C) \quad (2)$$

$$\Rightarrow (C - E.B^{-1}.F).x_C = y_C - E.B^{-1}.y_B$$

$A.x = B$ peut ainsi se résoudre en 3 étapes :

$$\begin{cases} B.z_B = y_B \\ S.x_C = y_C - E.z_B \\ B.x_B = y_B - F.x_C \end{cases} \quad (3)$$

$$\text{avec } S = C - E.B^{-1}.F = C - E.U^{-1}.L^{-1}.F$$

Généralités sur le complément de Schur (2/2)

Remarque : La factorisation LU de A peut s'écrire sous la forme :

$$A \approx \begin{pmatrix} L & \\ EU^{-1} & L_S \end{pmatrix} \times \begin{pmatrix} U & L^{-1}F \\ & U_S \end{pmatrix}$$

$$\text{avec } S = C - E.U^{-1}.L^{-1}.F$$

$$\begin{cases} x_C = U_S^{-1}L_S^{-1}(y_C - (\mathbf{E}\mathbf{U}^{-1}).L^{-1}y_B) \\ x_B = U^{-1}(L^{-1}y_B - (\mathbf{L}^{-1}\mathbf{F}).x_C) \end{cases}$$

$$\begin{cases} x_C = U_S^{-1}L_S^{-1}(y_C - E.U^{-1}.L^{-1}y_B) \\ x_B = U^{-1}.L^{-1}(y_B - L^{-1}.F.x_C) \end{cases}$$

Utilisation du complément de Schur (1/2)

Utilisation du complément de Schur :

- $B = L.U$: factorisation exacte
 $\Rightarrow$ résolution directe des sous-systèmes (1) et (3)
- $S \approx L_s.U_s$: factorisation incomplète
 $\Rightarrow$ (2) est résolu par une méthode de Krylov
 préconditionnée (GMRES)

$$\begin{cases} B.z_B = y_B & (1) \\ S.x_C = y_C - E.z_B & (2) \\ B.x_B = y_B - F.x_C & (3) \end{cases}$$

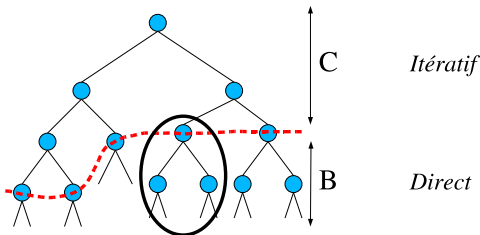
Remarques :

- Itérer sur S est alors numériquement équivalent à itérer sur A .
- Itération sur un système plus petit que le système global
 ($\Rightarrow$ paramètre de restart plus élevé dans GMRES)
- Pour avoir $\frac{\|r\|}{\|b\|} < tol$, il suffit que $tol_S < tol \cdot \frac{\|b\|}{\|b_S\|}$.

Utilisation du complément de Schur (2/2)

Arbre d'élimination :

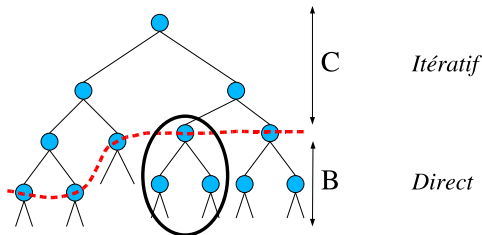
- Factorisation exacte partie inférieure de l'A.E.
- Le complément de Schur sur la partie haute de l'A.E. est résolu par une méthode itérative (Krylov).



Utilisation du complément de Schur (2/2)

Arbre d'élimination :

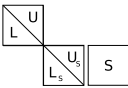
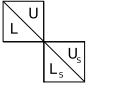
- Factorisation exacte partie inférieure de l'A.E.
- Le complément de Schur sur la partie haute de l'A.E. est résolu par une méthode itérative (Krylov).



Compromis : Choix de la proportion de directe/itératif en fonction de la difficulté du problème et de la précision désirée.

Itérations sur le complément de Schur (1/2)

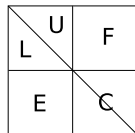
Itérations sur S : nécessite le calcul du produit de Schur $S.x$

	Produit de Schur	Stockage
M_1	$S.x$	
M_2	$C - E.U^{-1}.L^{-1}.F.x$	

Itérations sur le complément de Schur (2/2)

Coût d'une itération :

- 1 descente / remontée : $nnz(L_S) + nnz(U_S)$
- 1 produit matrice / vecteur :



M_1 : $nnz(S)$ opérations

M_2 : $nnz(C) + nnz(E) + nnz(F) + nnz(L) + nnz(U)$ opérations

Choisir M_1 ou M_2 ?

M_1 : partie importante du système résolue de manière exacte

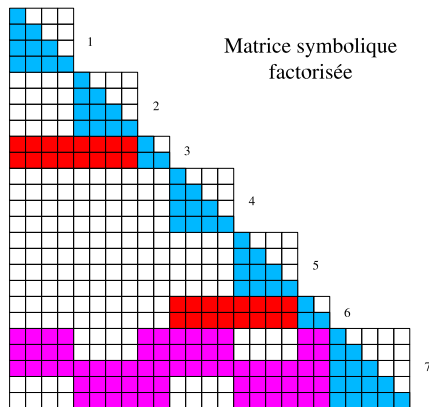
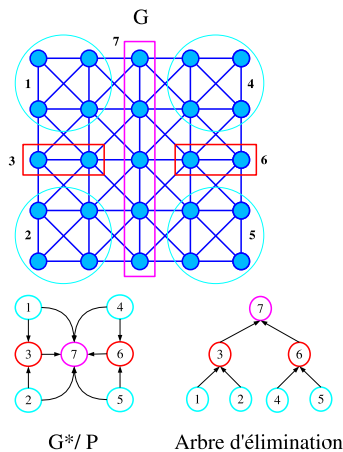
M_2 : petits domaines, fort remplissage de S (problèmes 3D)

$\Rightarrow$ *décidable à l'avance*

Plan

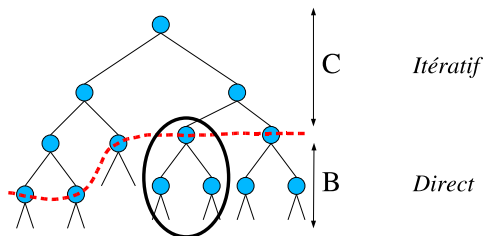
- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

Rappel sur les méthodes directes



Renumérotation et Partitionnement

- Partition à partir d'une numérotation issue du direct,
- **Repartitionnement des super-noeuds de C** : ordonnancement des inconnues de l'*interface* selon une **Décomposition Hierarchique de l'Interface** (HID) (Hénon, Saad, SIAM SISC).



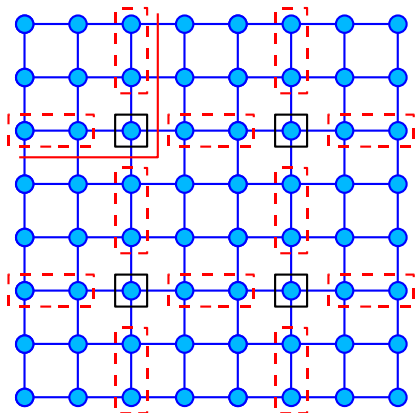
Ordonnancement des inconnues dans le cplt. de Schur

Exemple :

L'HID partitionne les inconnues en *connecteurs* :

- 1 Il n'y a aucune arête reliant 2 connecteurs d'un même niveau.
- 2 Un connecteur est toujours séparateur pour au moins 2 connecteurs du niveau inférieur.

⇒ Réordonnancement des inconnues.



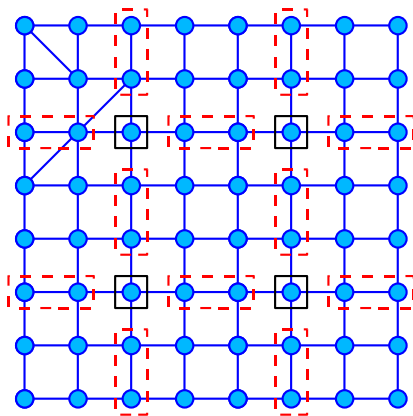
Ordonnancement des inconnues dans le cplt. de Schur

L'HID partitionne les inconnues en *connecteurs* :

- 1 Il n'y a aucune arête reliant 2 connecteurs d'un même niveau.
- 2 Un connecteur est toujours séparateur pour au moins 2 connecteurs du niveau inférieur.

⇒ Réordonnancement des inconnues.

Exemple :



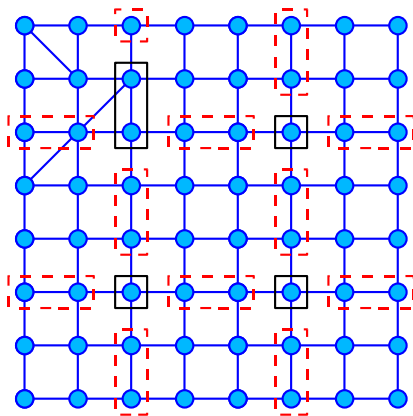
Ordonnancement des inconnues dans le cplt. de Schur

L'HID partitionne les inconnues en *connecteurs* :

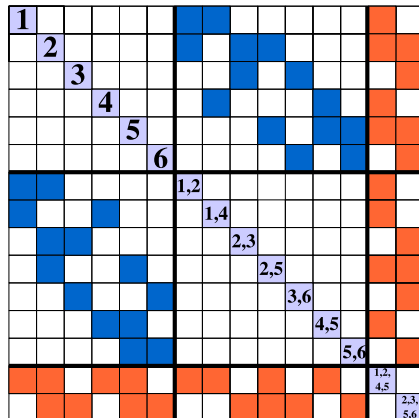
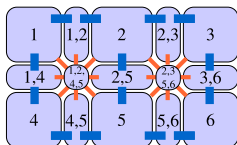
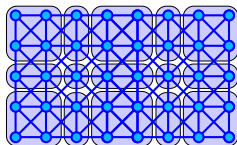
- 1 Il n'y a aucune arête reliant 2 connecteurs d'un même niveau.
- 2 Un connecteur est toujours séparateur pour au moins 2 connecteurs du niveau inférieur.

⇒ Réordonnancement des inconnues.

Exemple :

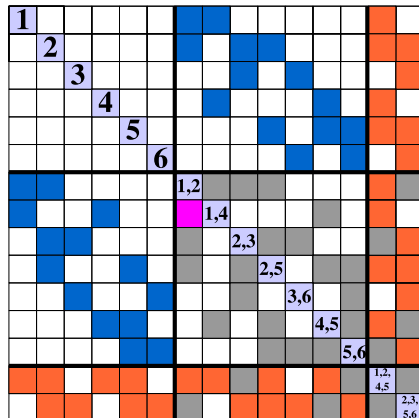
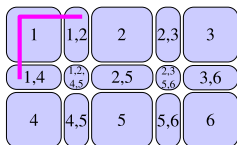
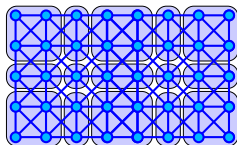


Stratégie de remplissage (1/2)



L_S , U_S : Remplissage **strictement consistant** : aucun remplissage autorisé entre les connecteurs d'un même niveau (même structure bloc que A).

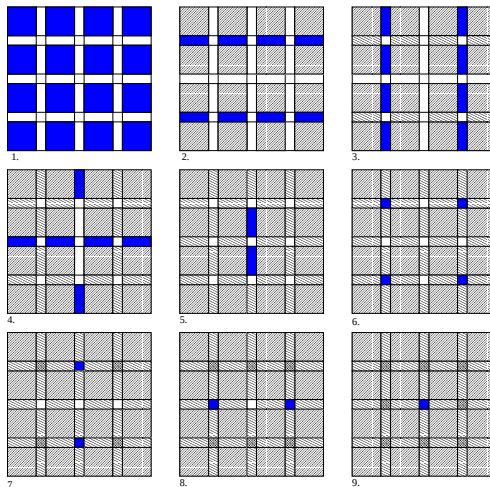
Stratégie de remplissage (2/2)



L_S , U_S : Remplissage **localement consistant** : remplissage autorisé qu'entre des connecteurs adjacents à un même sous-domaine (même structure bloc que S).

Élimination des inconnues en parallèle

Ordre d'élimination des inconnues (remplissage localement consistant) :



Plan

- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

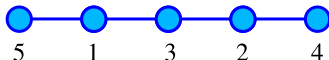
Factorisation symbolique par bloc

⇒ Prévoir le schéma de remplissage

Caractérisation du remplissage dans les méthodes directes :

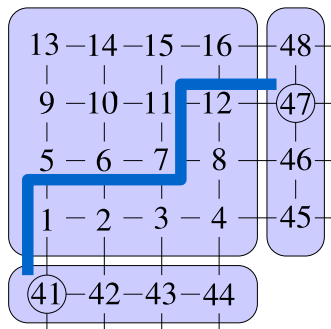
Théorème

Un terme nul a_{ij} de A deviendra non nul dans la factorisation $A = L.U$ ssi il existe un chemin allant du sommet i au sommet j en ne passant que par des sommets de plus petit numéro que i et j dans le graphe $G(V, E)$ d'adjacence de A .



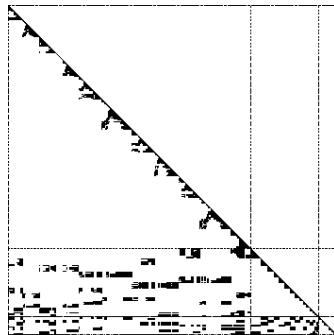
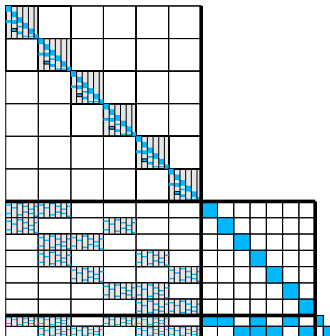
Factorisation symbolique par bloc

- Intérieur des domaines : factorisation symbolique par bloc du direct,
- Interface entre les domaines (complément de Schur) : la structure par **bloc dense** est donnée par le partitionnement des connecteurs et l'une des règles **localement** ou **strictement** consistant.



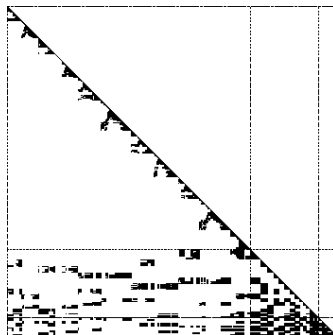
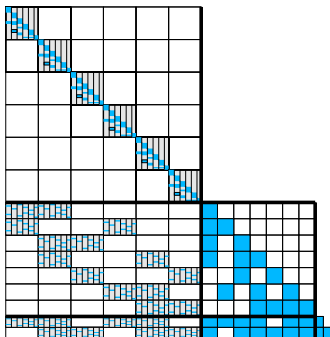
Factorisation symbolique par bloc

- Intérieur des domaines : factorisation symbolique par bloc du direct,
- Interface entre les domaines (complément de Schur) : la structure par **bloc dense** est donnée par le partitionnement des connecteurs et l'une des règles **localement** ou **strictement** consistant.

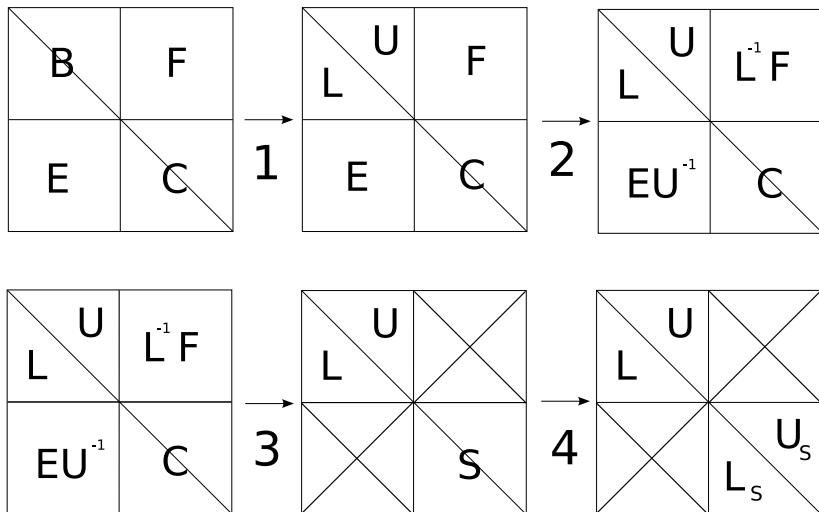


Factorisation symbolique par bloc

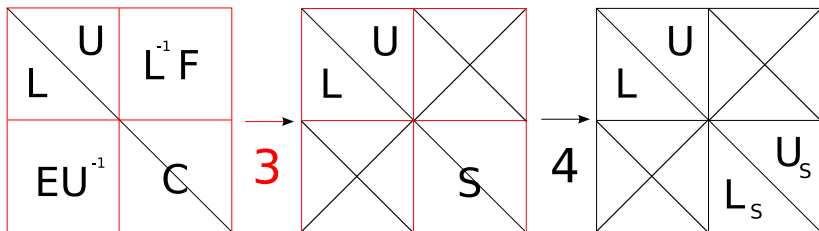
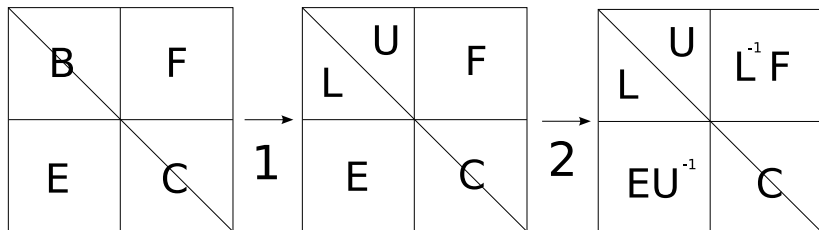
- Intérieur des domaines : factorisation symbolique par bloc du direct,
- Interface entre les domaines (complément de Schur) : la structure par **bloc dense** est donnée par le partitionnement des connecteurs et l'une des règles **localement** ou **strictement** consistant.



Factorisation numérique :

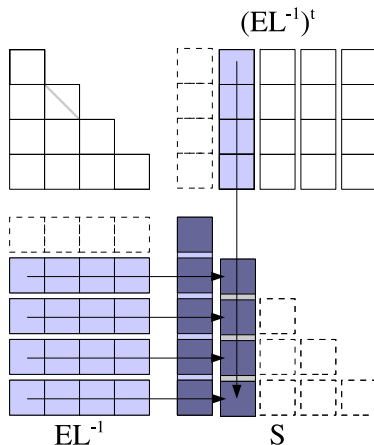
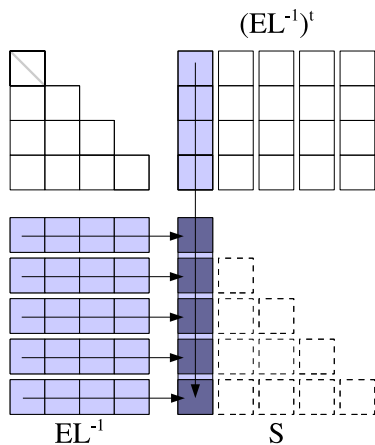


Factorisation numérique :



$$S = C - EB^{-1}F$$

Calcul du complément de Schur (matrice S) :



Plan

- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

Cas tests

Matrices :

- AUDI (collection PARASOL) : matrice symétrique
 $n = 943\,695$, $nnz(A) = 39\,297\,771$
- INLINE (collection PARASOL) : matrice symétrique
 $n = 503\,712$, $nnz(A) = 18\,660\,027$
- MHD1 : matrice non symétrique
 $n = 485\,597$, $nnz(A) = 24\,233\,141$

Précision relative du résidu à 10^{-7} .

Validation expérimentale : matrice AUDI

Avec un solveur direct :

$$nnz(L) = 32.nnz(A)$$

factorisation en 1280 secondes (PaStiX séquentiel)

Méthode de Krylov (GMRES, paramètre de restart = ∞)

► Préconditionneur $ILU(k)$ scalaire :

k	$ILU(k)$ (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
1	74.97	656.94	731.91	155	2.85
3	466.94	574.50	1041.44	92	6.45
5	1010.40	574.25	1584.65	77	8.72
7	1561.65	557.85	2119.5	67	10.35

Validation expérimentale : matrice AUDI

Décomposition de la matrice :

Taille domaines	Nombre de niveaux	Nombre de connecteurs	$\dim(S)/n$ (%)
500	1199	26694	34.42
1000	666	13864	28.27
2000	360	6368	22.78
5000	158	2024	16.26
10000	83	915	12.30

Validation expérimentale : matrice AUDI

► GMRES, préconditionneur M_1 et M_2 :

Remplissage de L_S et U_S **strictement** consistant.

Taille domaines	Précond. (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
Préconditionneur : M_1 (S stocké)					
1000	145.63	503.94	649.58	160	13.68
2000	161.06	356.75	517.81	142	13.97
5000	234.44	244.58	479.02	110	15.54
10000	339.06	228.32	567.39	108	17.84
Préconditionneur : M_2 (S non stocké)					
1000	91.24	380.34	471.58	160	3.53
2000	103.90	331.57	435.47	142	4.25
5000	157.95	298.06	456.01	110	6.00
10000	237.55	316.25	553.81	108	7.93

Validation expérimentale : matrice AUDI

► GMRES, préconditionneur M_1 et M_2 :

Remplissage de L_S et U_S **localement** consistant.

Taille domaines	Précond. (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
Préconditionneur : M_1 (S stocké)					
1000	668.41	197.28	865.69	33	22.21
2000	485.27	167.56	652.83	35	22.24
5000	530.61	125.19	655.80	30	23.70
10000	735.50	123.32	858.82	30	26.33
Préconditionneur : M_2 (S non stocké)					
1000	609.52	173.81	783.33	33	12.05
2000	466.83	160.93	627.76	35	12.51
5000	522.94	145.58	668.51	30	14.16
10000	727.99	145.49	873.48	30	16.42

Validation expérimentale : matrice INLINE

Avec un solveur direct :

$$nnz(L) = 10.nnz(A)$$

factorisation en 60.9 secondes (PaStiX séquentiel)

Méthode de Krylov (GMRES, paramètre de restart = ∞)

► Préconditionneur ILU(k) scalaire :

ne converge pas pour restart = 60

Lev. of fill	ILU(k) (sec.)	Solve (sec.)	Total (sec)	Iter	Fill ratio
5	235.4	454.7	690.1	198	5.15
7	310.6	423.3	733.9	183	5.67

Validation expérimentale : matrice INLINE

► GMRES, préconditionneur M_1 et M_2 :

Remplissage de L_S et U_S **strictement** constant.

Taille domaines	Précond. (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
Préconditionneur : M_1 (S stocké)					
1000	22.13	87.71	109.84	232	5.00
2000	24.84	57.77	82.61	212	5.44
5000	30.69	32.62	63.31	161	6.22
10000	36.21	23.34	59.55	135	6.81
Préconditionneur : M_2 (S non stocké)					
1000	18.48	182.44	200.92	232	2.74
2000	21.23	190.64	211.87	211	3.53
5000	25.54	177.34	202.88	180	4.65
10000	31.77	139.21	170.98	135	5.57

Validation expérimentale : matrice INLINE

► GMRES, préconditionneur M_1 et M_2 :

Remplissage de L_S et U_S **localement** consistant.

Taille domaines	Précond. (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
Préconditionneur : M_1 (S stocké)					
1000	32.77	156.67	189.44	250	6.81
2000	33.81	61.50	95.31	132	7.02
5000	39.92	33.11	73.03	87	7.51
10000	44.80	19.04	63.84	59	7.82
Préconditionneur : M_2 (S non stocké)					
1000	31.85	255.62	287.47	250	4.55
2000	32.13	139.84	171.97	132	5.11
5000	37.84	97.73	135.57	87	5.94
10000	44.22	68.21	112.43	59	6.58

Validation expérimentale

Remplissage de L_S et U_S **strictement** consistant.

Calcul du coût d'une itération :

AUDI :

Taille domaines	M_1 opc	M_2 #iter eq.
500	5.25+08	3.42
1000	4.62+08	2.92
2000	4.38+08	2.40
5000	4.28+08	1.73
10000	4.45+08	1.39

INLINE :

Taille domaines	M_1 opc	M_2 #iter eq.
500	5.86+07	1.21
1000	5.02+07	0.88
2000	4.16+07	0.60
5000	3.43+07	0.39
10000	2.72+07	0.26

Validation expérimentale : matrice MHD1

Avec un solveur direct :

$$nnz(L) = 43.nnz(A) \text{ (METIS)}$$

factorisation en ~ 2000 secondes (PaStiX séquentiel)

ILUT : ($\tau = 0.02$)

$$nnz(L) = 1,6.nnz(A)$$

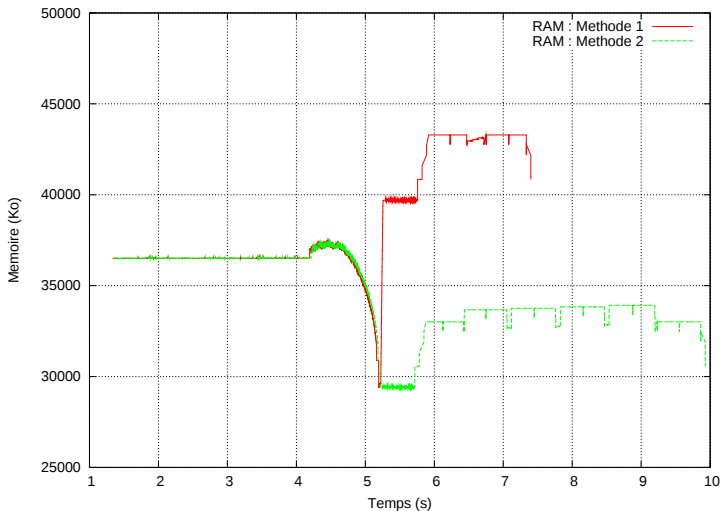
factorisation en ~ 160 secondes

M_2 : remplissage de L_S et U_S **strictement** consistant.

Taille domaines	Précond. (s)	Résolution (s)	Total (s)	Itér.	Remplissage (rapport)
300	47.60	189.60	237.20	174	2.01232
1000	53.25	103.11	156.35	130	2.65008
2000	67.56	86.44	154.00	107	3.5331
5000	99.75	72.69	172.44	86	4.99664
10000	186.03	65.23	251.26	69	6.96164

Validation expérimentale

► Pic mémoire : *(remplissage localement consistant)*

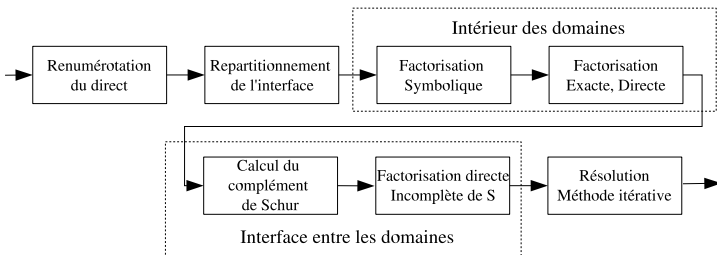


Plan

- 1 Méthodes de résolution
 - Méthode directe
 - Méthode itérative
- 2 Solveur hybride
 - Complément de Schur
 - Ordonnancement du complément de Schur
 - Factorisation symbolique
 - Factorisation numérique
- 3 Expériences avec la version séquentielle de HIPS
- 4 Conclusion et perspectives

Conclusion

Récapitulatif :



Conclusion

Conclusion :

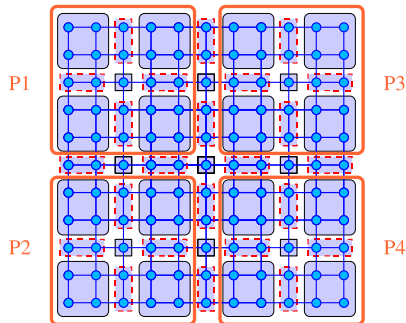
- Préconditionneurs plus robustes que $ILU(k)$
- Méthode efficace grâce aux calculs par blocs denses
- Choix possible d'un compromis mémoire / temps

Perspectives :

- Parallélisation du solveur (HID et factorisation)
- Critère automatique pour le choix de la méthode (taille des sous-domaines. . .)
- Amélioration du partitionnement et de la numérotation

Perspectives : parallélisation

Plusieurs sous-domaines par processeur (partitionner le graphe quotient) :



Recouvrement calculs / communications