HIPS: a parallel hybrid direct/iterative solver based on a Schur complement approach.

Sparse days at CERFACS

J. Gaidamour, P. Hénon

LaBRI and INRIA Bordeaux - Sud-Ouest (ScAlApplix team), France
ANR Project Solstice (ANR-06-CIS)

June 23-24, 2008

Outlines

Outlines

1. Introduction
2. Incomplete factorization based on a HID
3. Hybrid Solver
   - Experimental results
4. Parallelization
   - Experimental results
5. Conclusion

Introduction

HIPS : **H**ierarchical **I**terative **P**arallel **S**olver

Goals :

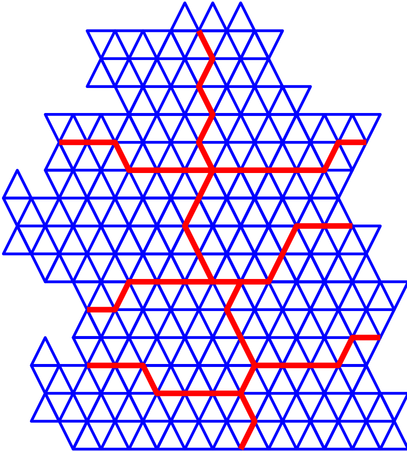
- Solve $A.x = b$
- Build algebraic preconditioners for a Krylov method : no information about the mathematical problem (black box).
- Parallelism of domain decomposition like methods (e.g. add. Schwarz methods) is appealing but
  ▶ convergence can decrease quickly with the number of domains.
- Build a global Schur complement preconditioner (ILU) from the local domain matrices only.

Introduction

We want the smallest Schur complement vs nb domains :

- use decomposition of the adj. graph of the matrix with an overlap of one-vertex wide.

Example :



Outlines

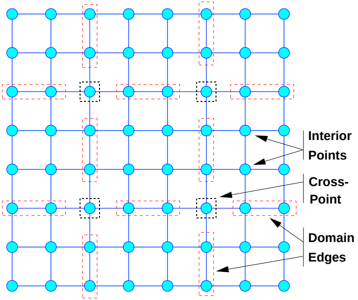
Incomplete factorization based on a HID [Hénon, Saad, 06]

We want an incomplete factorization of the matrix without creating edge (fill-in) outside the local domain matrices (keep the parallelism).
Problem : in a domain, we need an ordering of the interface compatible with neighboring domains.

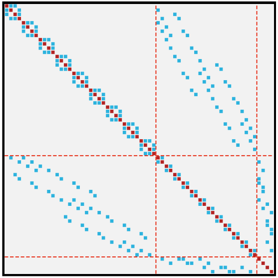
- Use a hierarchical interface decomposition :
  - partition the interface nodes according to the domains they belong to.
  - respect some properties to ensure good parallelism and numerical behavior.

Simple case : a 2D grid

In this case the HID



Grid 8 × 8.



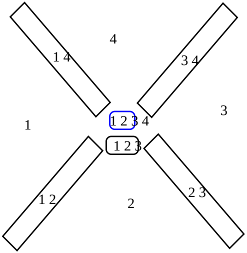
The reordered matrix.

We use the quotient graph induced by this partition to define block incomplete factorizations

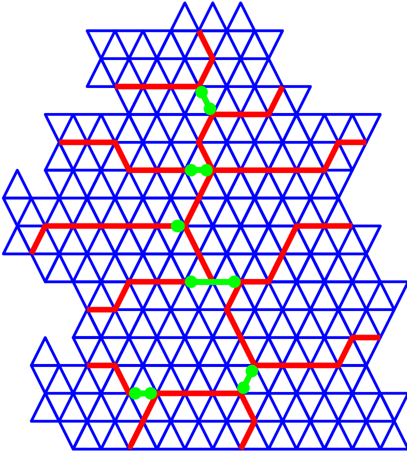
Extend to irregular graphs

A HID respects two rules :

1. Connectors of a same level are not connected
2. A connector of the level k is a separator for at least two connectors of the level $k-1$.



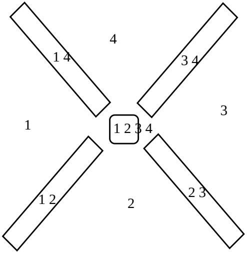
Example :



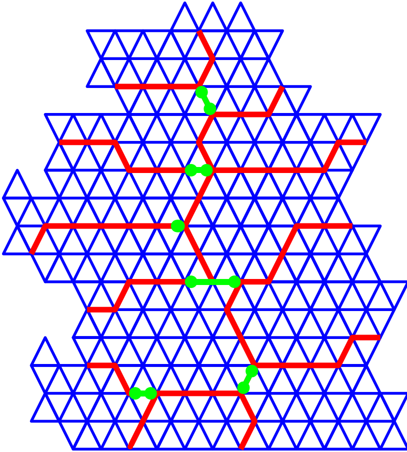
Extend to irregular graphs

A HID respects two rules :

1. Connectors of a same level are not connected
2. A connector of the level k is a separator for at least two connectors of the level $k-1$.

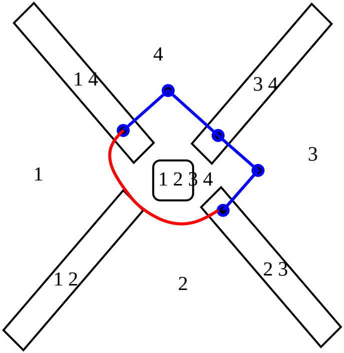


Example :

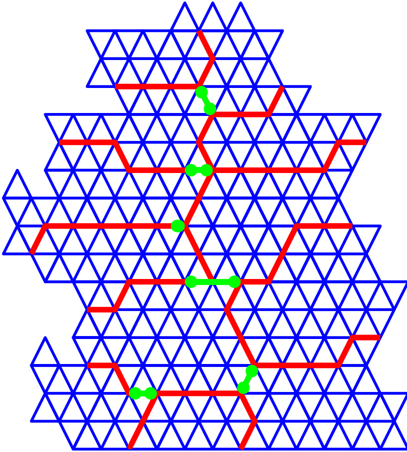


Extend to irregular graphs

- Unmatched elimination path are at least of length 4.



Example :



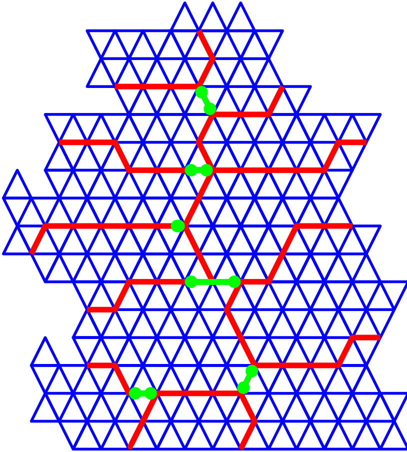
Extend to irregular graphs

A HID respects two rules :

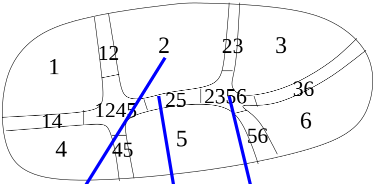
1. Connectors of a same level are not connected
2. A connector of the level k is a separator for at least two connectors of the level $k-1$.

- The HID heuristics (NP problems) :
  - minimize the number of nodes in the higher connector levels.
  - minimize the number of connector levels.

Example :



Fill-in block pattern (viewed from a local matrix)



Global domain partitioned into 6 subdomains

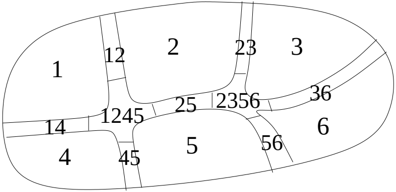
Local blocked−matrix for subdomain 2

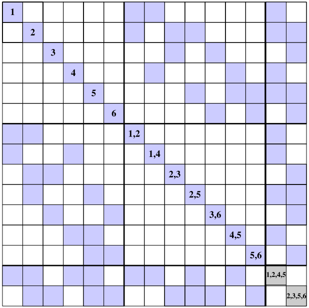
	1	2	3	4	5	6
1	2	2	2	2	2	2
2	2	1,2	2	2	1,2	2
3	2	2	2,3	2	2	2,3
4	2	2	2	2,5	2,5	2,5
5	2	1,2	2	2,5	1,2,4,5	2,5
6	2	2	2,3	2,5	2,5	2,3,5,6

Empty sparse matrix in intial matrix (fill−in occurs during factorization)

Fill−in in these blocks is allowed in the locally consistent strategy

Fill-in block pattern (viewed from the global matrix)

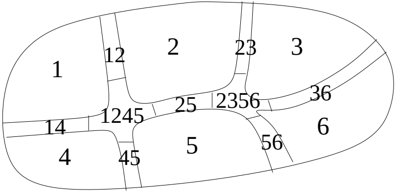


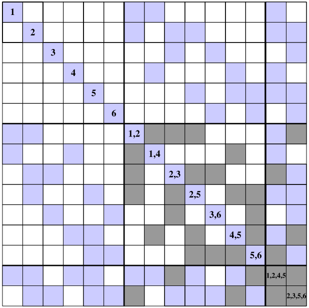


L_S, U_S : strictly consistent rule

No fill-in is allowed outside the initial block pattern of A (keep the block diag. struct.)

Fill-in block pattern (viewed from the global matrix)

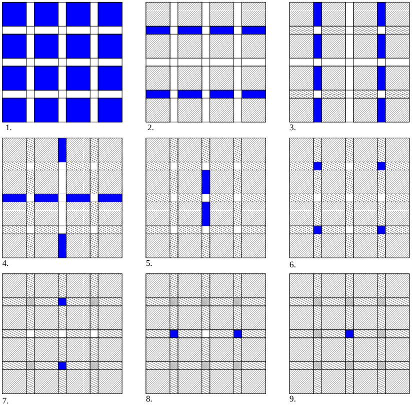




L_S, U_S : locally consistent rule

Fill-in is allowed in any place of the local domain matrices.

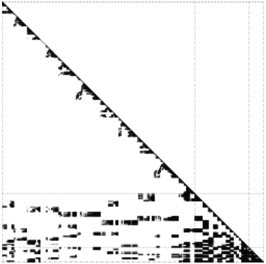
Elimination order (matters in locally consistent case) :



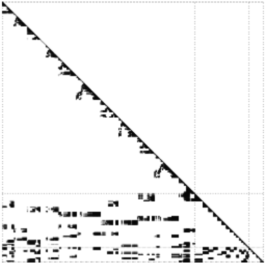
The connectors are ordered locally according to a global ordering. We use a MIS algorithm to eliminate at a time a maximum number of connectors.

Block fill-in pattern in a real case (bcsstk14)

- the strictly pattern is really "cheaper" than the locally pattern.
- we can use a ILUT (numerical threshold) to control the fill inside the block pattern.



Locally consistent rules



Strictly consistent rules

Outlines

Hybrid direct/iterative : Schur complement approach

The linear system $A.x = b$ can be written as :

$$\begin{pmatrix} A_B & A_{BC} \\ A_{CB} & A_C \end{pmatrix} . \begin{pmatrix} x_B \\ x_C \end{pmatrix} = \begin{pmatrix} y_B \\ y_C \end{pmatrix} \quad (1)$$

The system $A.x = B$ can be solved in three steps :

$$\begin{cases} A_B.z_B = y_B \\ S.x_C = y_C - A_{CB}.z_B \\ A_B.x_B = y_B - A_{BC}.x_C \end{cases} \quad (2)$$

with $S = A_C - A_{CB}.\mathbf{A_B^{-1}}.A_{BC} = A_C - A_{CB}.\mathbf{U_B^{-1}}.\mathbf{L_B^{-1}}.A_{BC}$

Hybrid direct/iterative : Schur complement approach

Schur Complement utilization :

- $A_B = L_B.U_B$: exact factorization
  $\Rightarrow$ direct resolution of subsystems (1) and (3)
  Each interior of subdomains can be computed independently
- $S \approx \tilde{L_S}.\tilde{U_S}$: incomplete factorization
  $\Rightarrow$ (2) is solved by a preconditioned Krylov subspace method
  Solve the Schur complement by a preconditioned Krylov method (GMRES).

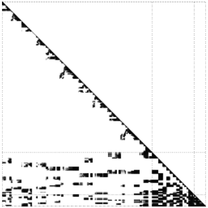
$$\begin{cases} A_B.z_B = r_B & (1) \\ S.x_C = r_C - A_{CB}.z_B & (2) \\ A_B.x_B = r_B - A_{BC}.x_C & (3) \end{cases}$$

Iterative resolution : Iterate on S is numerically equivalent to iterate on the whole system A.

Precondition the Schur complement

- Non-zero pattern of the global factors obtained on a small matrix : (Fill-in allowed only in local Schur complement)

$$\begin{pmatrix} L_B & \\ A_{CB}{U_B}^{-1} & S \end{pmatrix}$$



- How to avoid memory cost of $A_{CB}{U_B}^{-1}$ and S in 3D problems [Gaidamour, Hénon, 08] :
  - We do not need to store S, instead of $S.x$ we use $(A_C - A_{CB}.U_B^{-1}.L_B^{-1}.A_{BC}).x$
  - ILUT : $A_{CB}{U_B}^{-1}$ (resp. ${L_B}^{-1}A_{BC}$) is numerically sparsified along its computations (blockwise ILUC(t)= left looking) : $\tilde{L_S}.\tilde{U_S} \approx \tilde{S} = (A_{CB}.\widetilde{U_B^{-1}}).(\widetilde{L_B^{-1}}.A_{BC}) \approx S$

Test cases

Test cases from grid-TLSE collection (http://gridtlse.enseeiht.fr:8080/websolve/)

MHD1 :

- Unsymmetric real matrix
- 3D magneto-hydrodynamic flow problem

AUDI :

- Symmetric real matrix
- 3D structural mechanic problem

Haltere, Amande :

- Symmetric complex matrix
- 3D electromagnetism problems (Maxwell)

Test cases

Fill-in and OPC are given for a direct method.

Matrix	unknowns	non-zeros	fill-in	OPC
MHD1	485, 597	24, 233, 141	52.4	$9.0 \,.\, 10^{12}$
AUDI	943, 695	39, 297, 771	41.2	$5.4 \,.\, 10^{12}$
Haltere	1, 288, 825	10, 476, 775	38.7	$7.5 \,.\, 10^{12}$
Amande	6, 994, 683	58, 477, 383	53.9	$1.5 \,.\, 10^{13}$

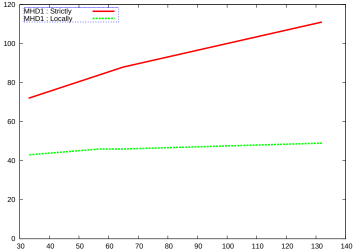
Test cases

Experimental conditions :

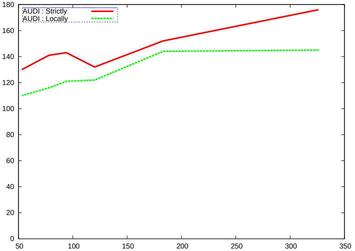
- 10 nodes of 2.6 Ghz quadri dual-core Opteron (Myrinet)
- Partitionner : Scotch
- $||b - A.x||/||b|| < 10^{-7}$, no restart in GMRES
- Thresholds : MHD1, Audi, Amande = 0.001, Haltere = 0.01
- The Amande test case does not converge with the strictly pattern and better results were obtained with using no threshold in coupling (L_s, L_s^t are computed from the exact Schur complement).

Iterations/number of domains

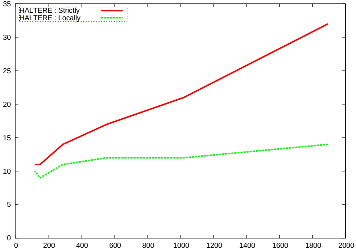
MHD1 (485, 597) : nb domains from 33 to 132



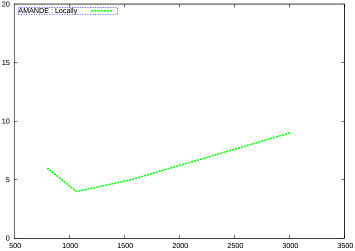
AUDI (943, 695) : nb domains from 53 to 326



HALTERE (1, 288, 825) : nb domains from 119 to 1894

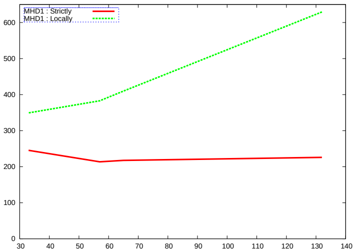


AMANDE (6, 994, 683) : nb domains from 801 to 3004

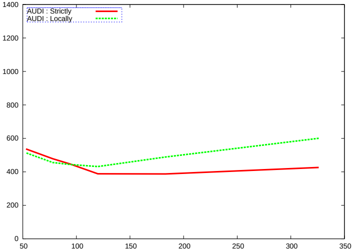


Seq. Times (prec+solve)/number of domains

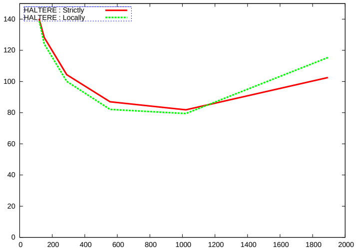
MHD1 (485, 597) : nb domains from 33 to 132



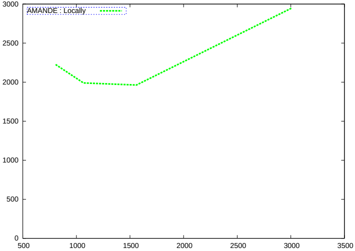
AUDI (943, 695) : nb domains from 53 to 326



HALTERE (1, 288, 825) : nb domains from 119 to 1894

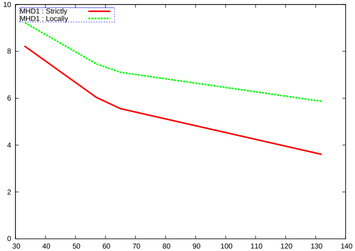


AMANDE (6, 994, 683) : nb domains from 801 to 3004

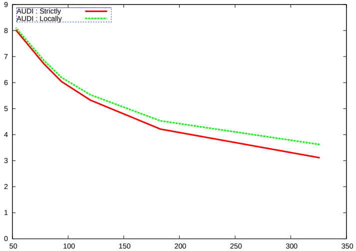


Fill ratio/number of domains (Amande : Peak = +3.5)

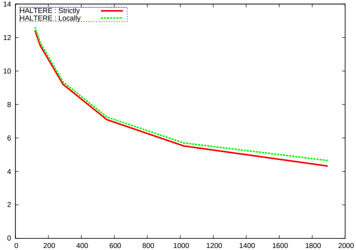
MHD1 (485, 597) : nb domains from 33 to 132



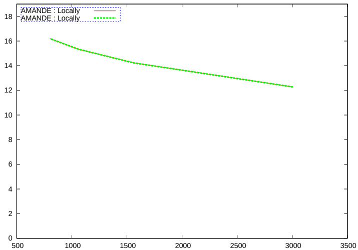
AUDI (943, 695) : nb domains from 53 to 326



HALTERE (1, 288, 825) : nb domains from 119 to 1894



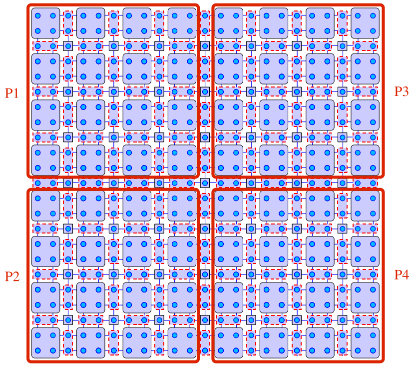
AMANDE (6, 994, 683) : nb domains from 801 to 3004



Outlines

Unknown elimination in parallel

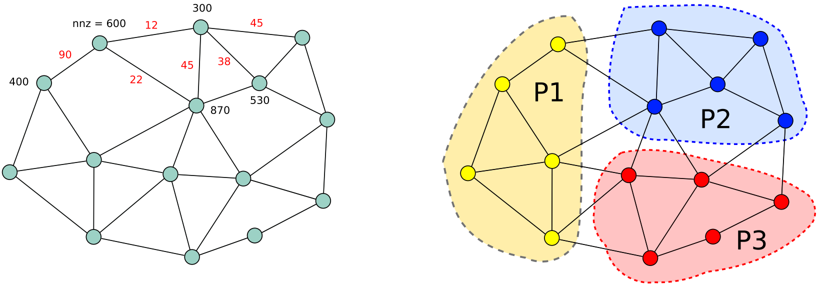
We build a decomposition of the adjacency graph of the system into a set of **small subdomains** ($\simeq$ 1000 to 5000 nodes).



We can recover communications between processors by elimination of local subdomains

Workload and memory balance

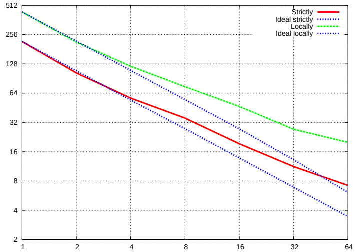
- Subdomains distribution on the processors :
  - Obtained by partitioning the weighted "domain graph" (SCOTCH or Metis)
  - Balance of $S.x$ computation (solving step) by using the symbolic factorization to compute the number of NNZ of the interiors of subdomains.



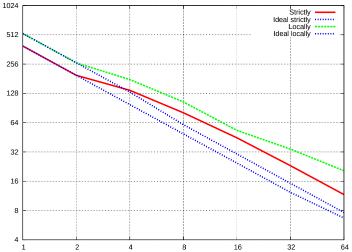
- Finer level of balance : election of the processor responsible for the computation of a piece of interface (connectors).

Parallel time (prec+solve) (logarithmic scale)

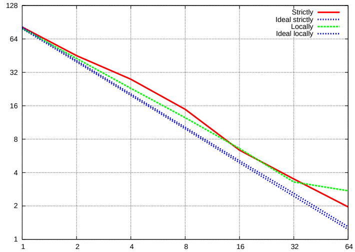
MHD1 (485, 597) : 64 domains



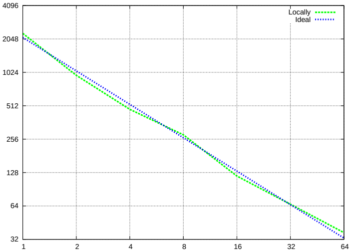
AUDI (943, 695) : 231 domains



HALTERE (1, 288, 825) : 1062 domains



AMANDE (6, 994, 683) : 2062 domains



Outlines

Conclusion

Conclusion :

- Generic algebraic approach, tight coupling between state of the art direct method (supernodal) and ILU(t) implementation,
- Trade-off performance/memory : easy tuning (choose thresholds, domain size)

Prospects :

- Provide automatically a domain size parameter (based on memory).
- Partitioning is an open question...numerical weight, connectivity
- Coarse grid based on the HID (for elliptic problems)

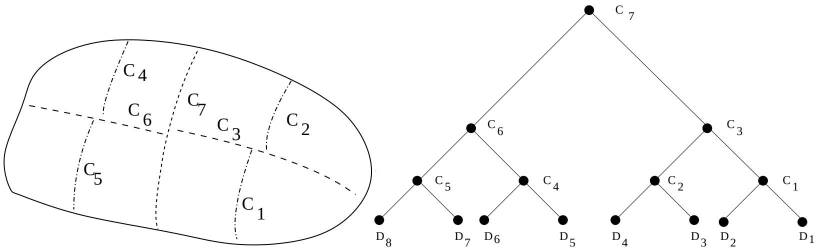
Download HIPS

http://hips.gforge.inria.fr

- Features : symmetric, unsymmetric (Cholesky/LU, ICC(t)/ILU(t)), real or complex systems.
- Method : Hybrid, multistage ICC(t), ILU(t).
- Compatible with the graph partitioners SCOTCH and METIS.
- Cecill-C license (LGPL-like licence)

Construction of the domain partition

The domain partition is constructed from the reordering based on Nested-Dissection like algorithms (eg : METIS, SCOTCH)

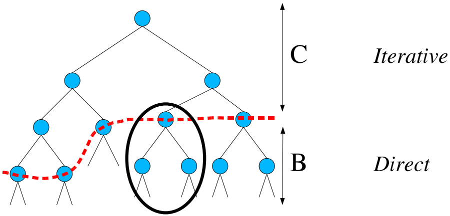


⇒ Minimize overlap between subdomains, quality of the interface

Construction of the domain partition

We choose a level of the elimination tree of direct method :

- Subtrees rooted in this level are the interior of subdomains
- The upper part of the elimination tree corresponds to the interfaces

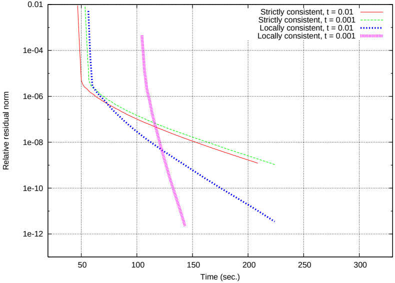


Possibility to choose the ratio of direct/iterative according to the problem difficulty or the accuracy needed.

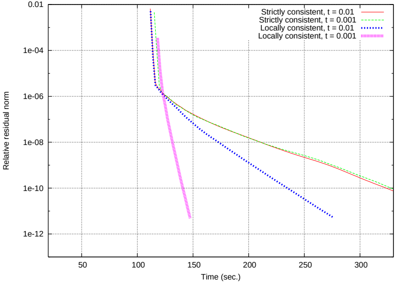
Test case : Haltere (sequential study)

- Convergence/time for several parameters with two different domain size parameters :

Domain size set to 1000 (1021 *domains) :*



Domain size set to 10000 (119 *domains) :*



(preconditioning time = curve offset)

Test case : Haltere (parallel study)

- HIPS : ILUT ($\tau = 0.01, 10^{-7}$)
  - 1021 domains of $\simeq$ 1481 nodes
  - fill ratio in precond : 5.70 (peak)
  - $dim(S) = 14.26\,\%$ of $dim(A)$

Strictly consistent :

21 iterations

fill ratio in solve : 5.52

# proc	Precond. (sec.)	Solve (sec.)	Total (sec.)
1	45.09	36.74	81.84
2	24.48	20.76	45.24
4	12.08	15.65	27.74
8	6.15	8.71	14.86
16	3.06	3.31	6.37
32	1.58	1.92	3.50
64	0.89	1.07	1.96

Locally consistent :

13 iterations

fill ratio in solve : 5.69

# proc	Precond. (sec.)	Solve (sec.)	Total (sec.)
1	54.55	24.90	79.45
2	29.17	13.50	42.68
4	14.28	8.69	22.96
8	7.31	5.19	12.50
16	3.82	2.76	6.58
32	1.97	1.31	3.29
64	1.89	0.86	2.75

Test case : Amande

TAB.: Direct factorization using MUMPS

Â

Haltere : Â

# proc	Facto. (sec.)	Solve (sec.)	Total (sec.)	nnz_{max} $\times 10^6$
16	29	0.44	29.44	52.8
32	16	0.26	16.26	22.6

Â

Amande : Â

# proc	Facto.	Solve	Total	nnz_{max}
16	512	4.5	516.5	407.7
32	299	2.4	301.4	179.5

Test case : Amande

- HIPS : ILUT (locally consistent, $\tau = 0.001$, 10^{-7})

Time decomposition for one iteration of GMRES :

# proc	Total 1 Iter. (sec.)	Triangular Solve (sec.)	$S.x$ (sec.)	Other (sec.)
2	11.29	3.94	6.91	0.44
64	0.58	0.19	0.31	0.08