A parallel direct/iterative solver based on a Schur complement approach

Gene around the world at CERFACS

Jérémie Gaidamour

LaBRI and INRIA Bordeaux - Sud-Ouest (ScAlApplix project)

February 29th, 2008

Outline

Plan

Motivation of this work

The most popular algebraic methods to solve large sparse linear system $A.x = b$ are :

Direct method (exact factorization)

- Build a dense block structure of the factor (BLAS 3)
- Solution have a great accuracy ($\approx 10^{-15}$)
- High memory consumption (unable to solve very large 3D problems)

Preconditioned iterative methods

- Robustness depends on how much memory is allowed in the preconditioner
- Based on scalar implementation (eg : ILU(k) or ILUT)
- Convergence difficult on very ill-conditioned system

$\Rightarrow$ we want a trade-off : a solver that can solve difficult problems and that requires less memory than direct solver

Our approach

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

- Generic algebraic approach : no information about the problem (black box)
- Use direct solver technologies (BLAS, elimination tree ...)

- Build a decomposition of the adjacency graph of the system into a set of **small subdomains** with **overlap**.

  We want to solve a boundary problem
  $\Rightarrow$ need a robust preconditioner in the Schur complement.

Plan

Schur complement (1/2) :

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

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

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

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

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

Schur complement (2/2) :

Schur Complement utilization :

- $B = L.U$: exact factorization
  $\Rightarrow$ direct resolution of subsystems (1) and (3)
  Each interior of subdomains can be computed independently
- $S \approx L_s.U_s$: incomplete factorization
  $\Rightarrow$ (2) is solved by a preconditioned Krylov subspace method
  Solve the Schur complement by a preconditioned GMRES.

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

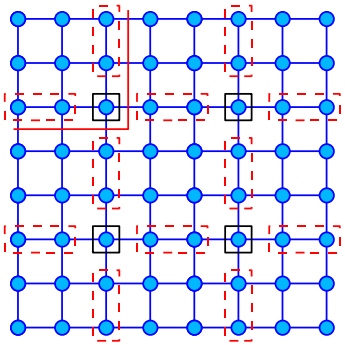
Iterative resolution :

- Iterate on S is numerically equivalent to iterate on the whole system A.
- We do not need to store S to compute Schur product using its implicit formulation : $(C - E.U^{-1}.L^{-1}.F).x$

Ordering and partitioning of the Schur complement

We need a special ordering for the Schur complement to compute a block incomplete factorization.

The unknowns in the interface (in the Schur complement) are ordering according to a Hierarchical Interface Decomposition (Hénon, Saad, SIAM SISC).



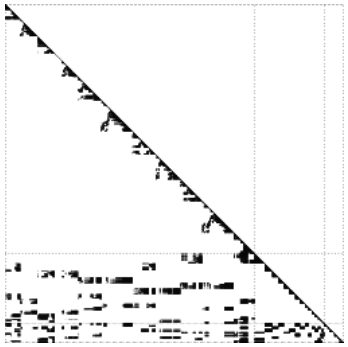
The unknowns are partitioned into connectors to insure that :

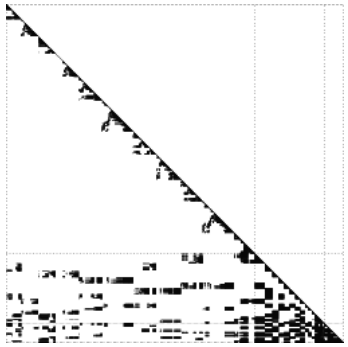
1. There is no edges between two connectors of a same level
2. Any connector is a separator for at least 2 connectors of the inferior level

   $\Rightarrow$ *give elimination order, parallelism*

Precondition the Schur complement

We use the quotient graph induced by this partition to define block incomplete factorizations and two different block fill-in patterns :

(1) 

(2) 

(1) Strictly consistent rules :
No fill-in is allowed between the connectors of a same level (same block pattern than A).

(2) Locally consistent rules :
Fill-in allowed between connectors adjacent to a same domain (same block pattern than S).

▶ ILUT (numerical dropping according to a threshold) inside choosen block pattern

Plan

Construction of the domain partition

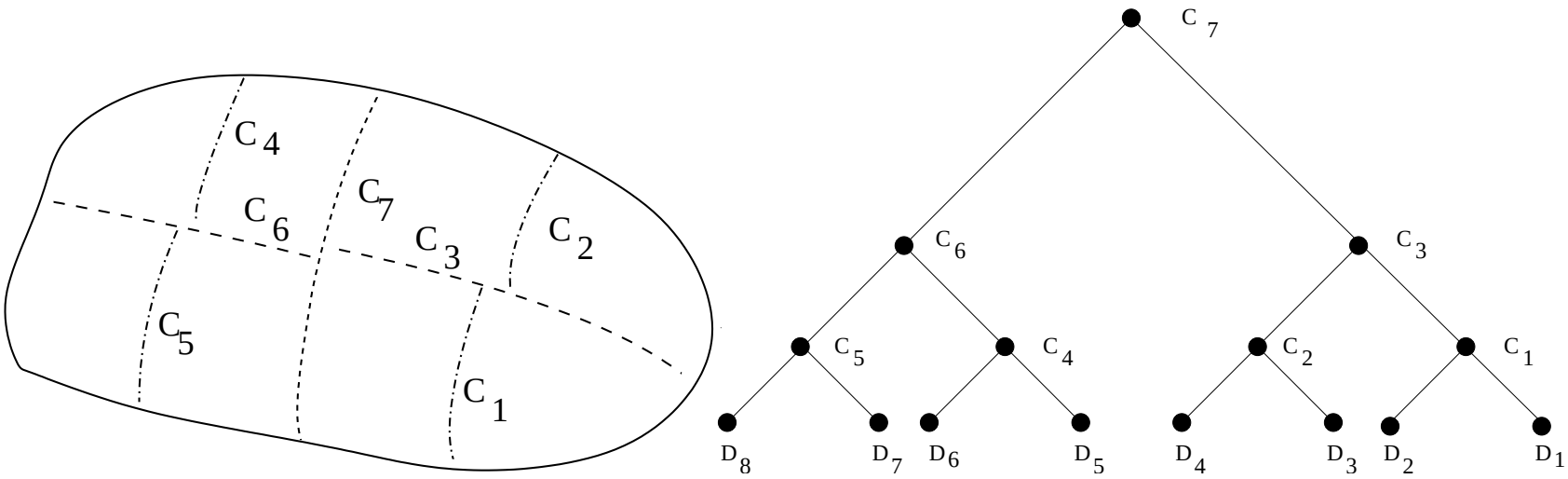
We build a decomposition of the adjacency graph of the system into a set of **small subdomains** ($\simeq$ 100 - 1000 nodes).

Justification of small subdomains choice :

- Need low memory (not too much direct),
- Convergence independent of the number of processors,
- Number of subdomains become a parameter to control memory / convergence according to the problem difficulty,
- Give high potential parallelism (multiple domains per processors).

Construction of the domain partition

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

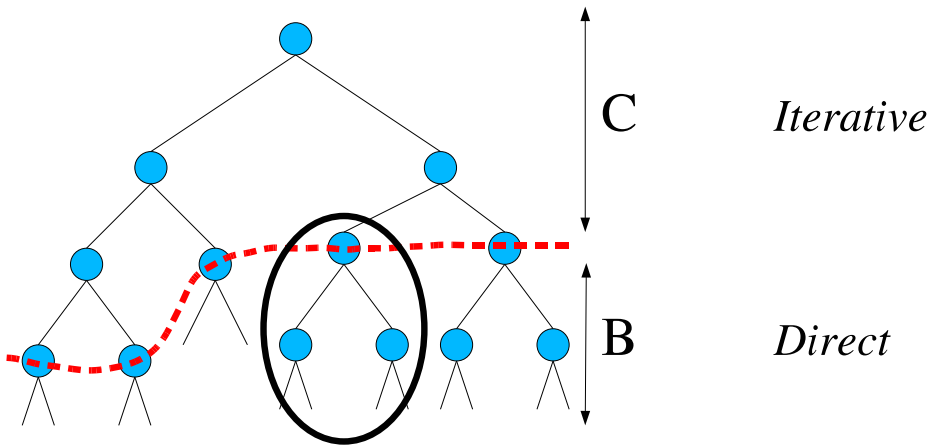


$\Rightarrow$ Minimize overlap between subdomains, quality of the interface

Construction of the domain partition

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

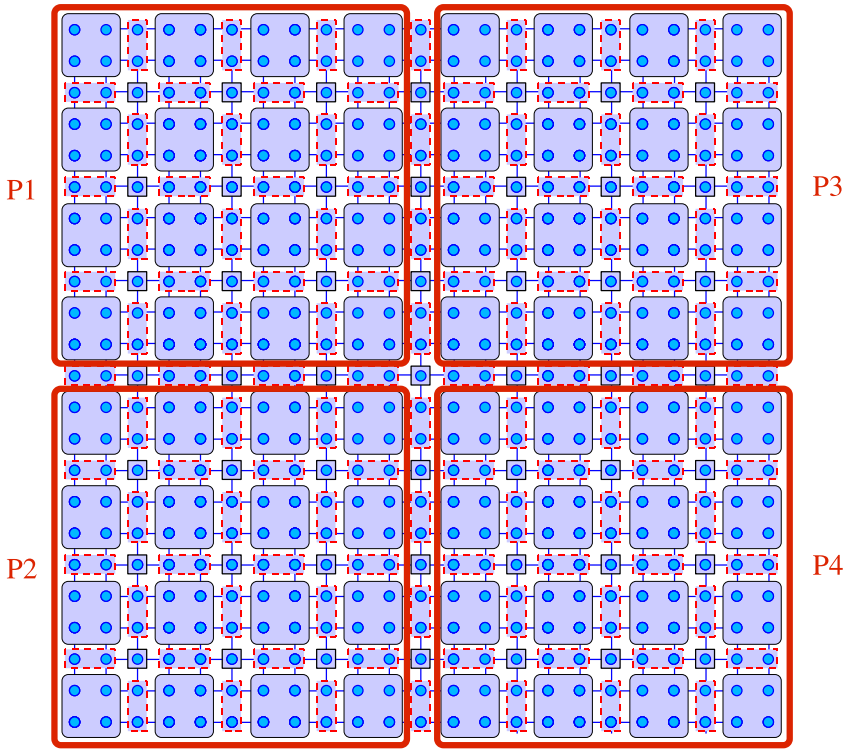
- Subtrees rooted in this level are the interior of subdomains
- The upper part of the elimination tree corresponds to the interfaces



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

Unknown elimination in parallel

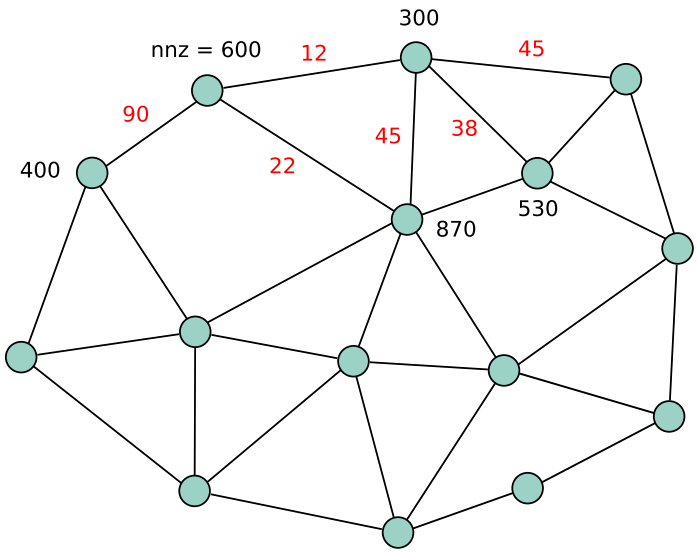
Many small subdomains per processors :

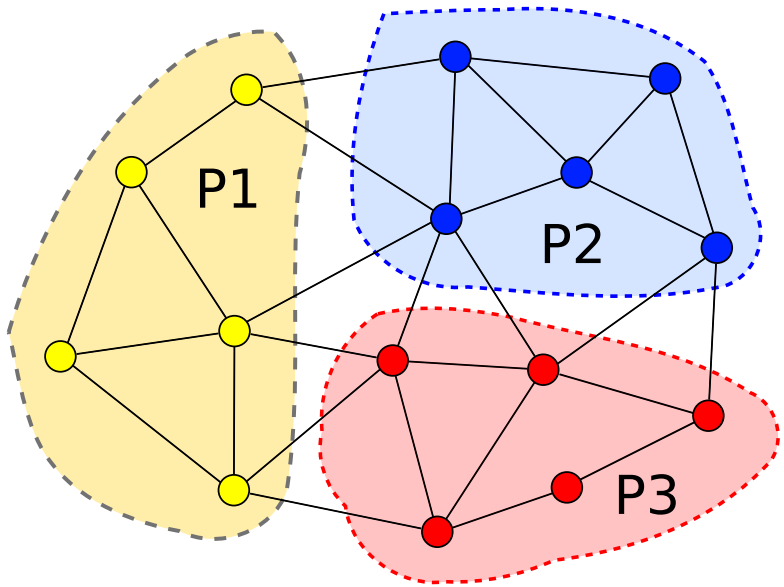


Perspective : We can recover communications between processors by elimination of local subdomains

Equilibration

- Subdomains distribution over available processors :
  - Equilibration using a graph partitionner (SCOTCH)
  - Equilibration of $S.x$ computation (solving step) by using the symbolic factorization to compute the number of NNZ of the interiors of subdomains.





- Election of the processor responsible for the computation of a piece of interface (connectors).

Plan

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

Tests cases :

Haltere, Amande (CEA/CESTA) :

- Symmetric complex matrix
- 3D electromagnetism problems (Helmholtz operator)

Test case : Haltere (sequential study)

Haltere (CEA/CESTA) :

- $n = 1,288,825$; $nnz(A) = 10,476,775$, fill ratio : x 38.65

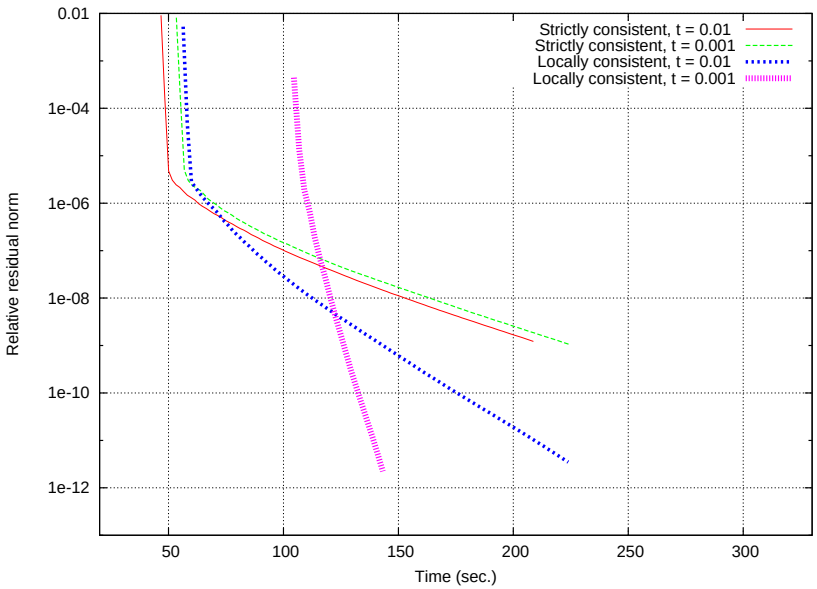
- HIPS : ILUT (locally consistent, $\tau = 0.01$, 10^{-7})

# domains	Precond. (sec.)	Solve (sec.)	Total (sec.)	Iter.	Fill ratio
1894	77.99	37.37	115.36	14	4.65
1021	54.55	**24.90**	**79.45**	**12**	**5.70**
555	56.49	25.62	82.10	12	7.25
289	73.01	27.09	100.10	11	9.35

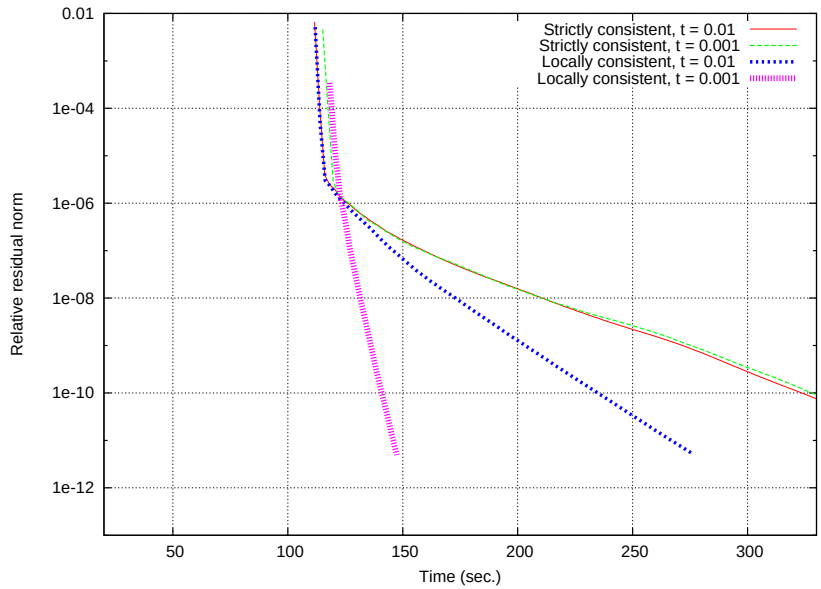
Test case : Haltere (sequential study)

- 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	2.74

Test case : Amande

Amande (CEA/CESTA) :

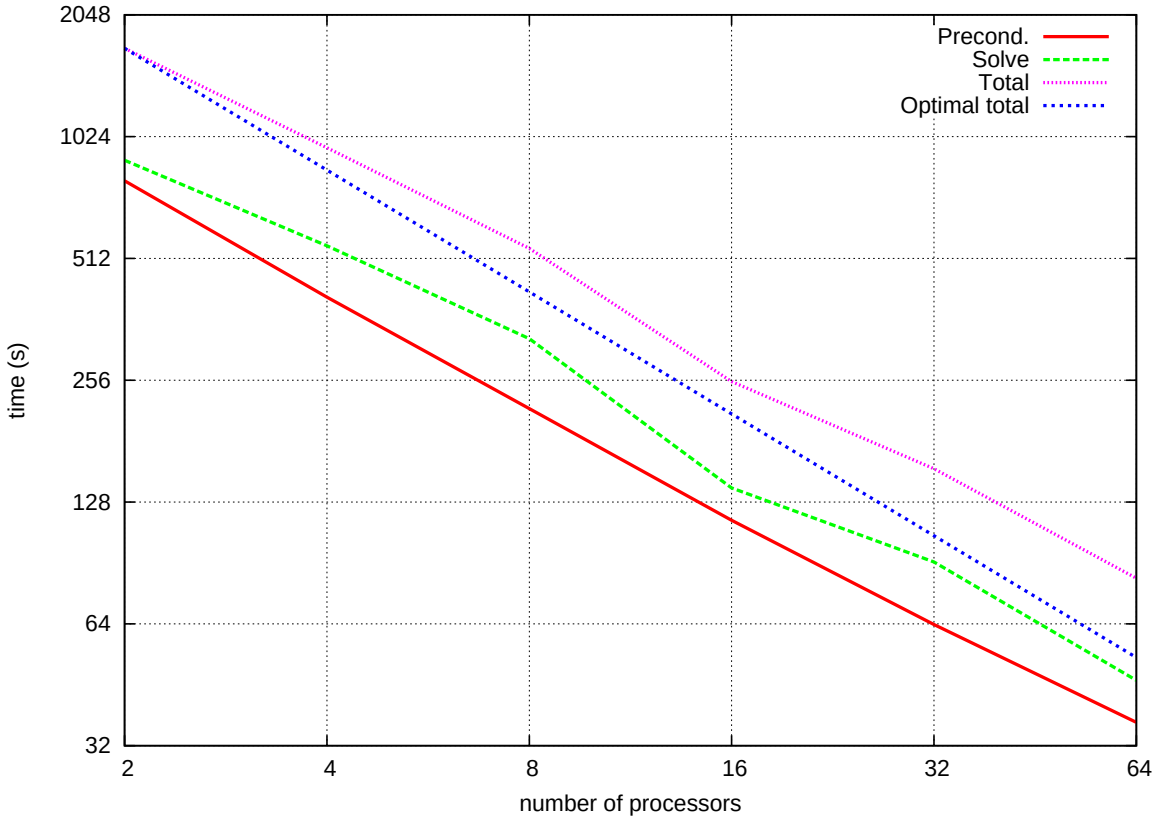
- $n = 6,994,683$; $nnz(A) = 58,477,383$, fill ratio : x 53.87

- HIPS : ILUT (locally consistent, $\tau = 0.001$, 10^{-7})
  - 2053 domains of $\simeq$ 3770 nodes
  - 77 iterations
  - fill ratio in precond / solve : 13.53 (peak)
  - $dim(S) = 9.59\,\%$ of $dim(A)$

# proc	Precond. (sec.)	Solve (sec.)	Total (sec.)	$nnz(P_{max}).10^6$
2	796.71	895.20	1691.91	399.38
4	410.68	550.35	961.03	200.87
8	217.76	324.20	541.96	100.76
16	115.37	138.75	254.12	50.77
32	63.78	91.01	154.79	25.91
64	36.53	46.43	82.96	13.15

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

Plan

Conclusion

Conclusion :

- Generic algebraic approach, mix direct and iterative methods thought a Schur complement approach,
- The part of direct factorization is controlled by the size of domains,
- Many different strategies are implemented (dense block ILU).

Perspective (preprocessing) :

- PT-Scotch integration,
- Parallel interface renumbering,
- Providing indications about good domain size parameters.

HIPS public release :

- March 2008 (Cecill-C license)
- Features : real (symmetric, unsymmetric), complex (symmetric)

`http://hips.gforge.inria.fr`