Complexity and parallelism of the solution phase in sparse direct solvers

P. Amestoy[1] A. Buttari[2] J.-Y. L'Excellent[3] T. Mary[4] G. Moreau[3]

[1]INP-IRIT, [2]CNRS-IRIT, [3]INRIA-LIP, [4]University of Manchester

PMAA18, June 29, 2018

Systems of linear equations:

$Ax = b$, where A is sparse. In direct methods, 3 steps:

- analysis: nested dissection;
- factorization: $A \to LU$;
- solve: $Ly = b$ and $Ux = y$.

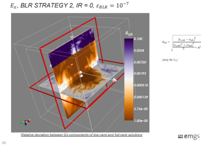
$\mathcal{C}$	$n = N \times N$	$n = N \times N \times N$
Factorization	$\Theta(N^3)$	$\Theta(N^6)$
Solve	$\Theta(N^2 \log N)$	$\Theta(N^4)$

Complexities on regular 2D/3D problems[1]. N is the grid size.

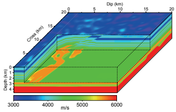
Factorization is usually the most expensive part, however solve can be critical...

[1] **geor:73**.

Critical solve: one RHS multiple times, multiple RHS...



Example of applications:
Helmholtz or Maxwell equations

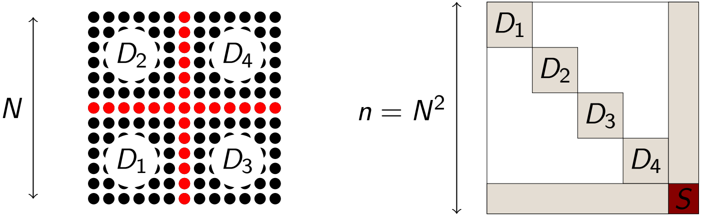


matrix		n	$nrhs$	T_{facto}	T_{solve}
SEISCOPE	5Hz	2.9M	2302	44	236
	10Hz	17.2M	2302	779	2585
EMGS	H3	2.8M	8000	82	569
	H17	17.4M	8000	1559	8118

Run on EOS computer, 90 nodes (full rank solve).

More attention should be given to the complexity of the solve phase!

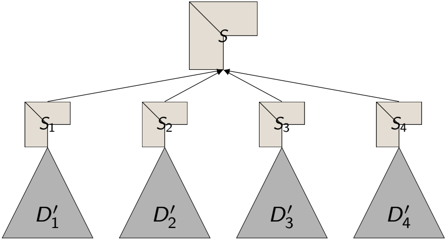
Nested dissection



Ordering

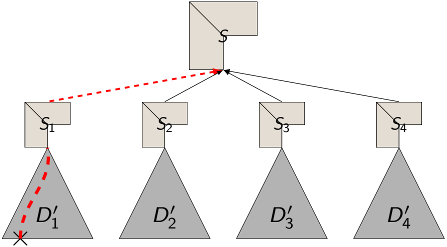
Nested dissection (ND): divide and conquer algorithm to reorder variables of the matrix A to reduce fill-in and build the separator tree.

Separator tree and solve algorithm



- Separator tree: representation of the dependencies between computations during the solve algorithm.
- Solve algorithm: $Ly = b$ (resp. $Ux = y$) follows a bottom up (resp. top down) traversal of the separator tree;

Separator tree and solve algorithm

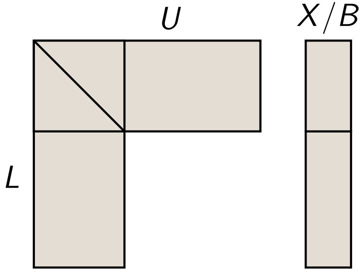


- Separator tree: representation of the dependencies between computations during the solve algorithm.
- Solve algorithm: $Ly = b$ (resp. $Ux = y$) follows a bottom up (resp. top down) traversal of the separator tree;

Critical path: longest path in the separator tree in terms of operations.

At each node

$\mathcal{C}_{dense}(m) = \Theta(m^\alpha)$: solve complexity for node of size m.



Full-rank (forward):

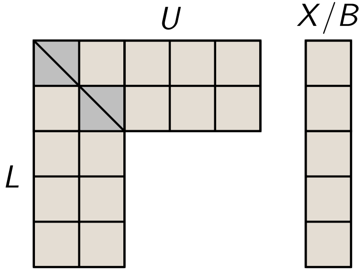
- $y_1 \leftarrow L_{11}^{-1} b_1$;
- $b_2 \leftarrow b_2 - L_{21} y_1$.

$\Rightarrow \mathcal{C}_{dense}(m) = \Theta(m^2)$

[2] **ablm:17**.

At each node

$\mathcal{C}_{dense}(m) = \Theta(m^{\alpha})$: solve complexity for node of size m.



Full-rank (forward):

- $y_1 \leftarrow L_{11}^{-1} b_1$;
- $b_2 \leftarrow b_2 - L_{21} y_1$.

$\Rightarrow \mathcal{C}_{dense}(m) = \Theta(m^2)$

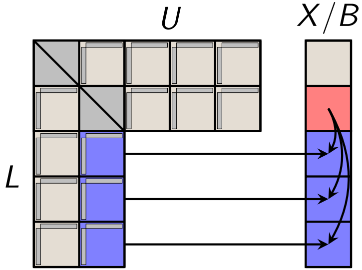
Block Low-rank (BLR): low-rank property on off-diagonal block:

$C \approx UV^T$, with U, V of size $m \times r$

[2] **ablm:17**.

At each node

$\mathcal{C}_{dense}(m) = \Theta(m^\alpha)$: solve complexity for node of size m.



Full-rank (forward):

- $y_1 \leftarrow L_{11}^{-1} b_1$;
- $b_2 \leftarrow b_2 - L_{21} y_1$.

$\Rightarrow \mathcal{C}_{dense}(m) = \Theta(m^2)$

Block Low-rank (BLR): low-rank property on off-diagonal block:

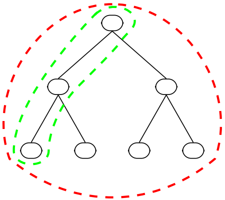
$$C \approx UV^T, \text{ with } U, V \text{ of size } m \times r \Rightarrow \mathcal{C}_{dense}(m) = \Theta(m^{1.5})$$

Complexity: $\Theta(N^2 \log N) \rightarrow \Theta(N^2)$ in 2D, $\Theta(N^4) \rightarrow \Theta(N^3)$ in 3D[2]

[2] **ablm:17**.

We consider the potential gain $\mathcal{G}(N)$ such that

$$\mathcal{G}(N) = \frac{\mathcal{C}(N)}{\mathcal{C}^c(N)}$$



where

- $\mathcal{C}(N)$ is the complexity of the solve phase;
- $\mathcal{C}^c(N)$ is the complexity of the critical path.

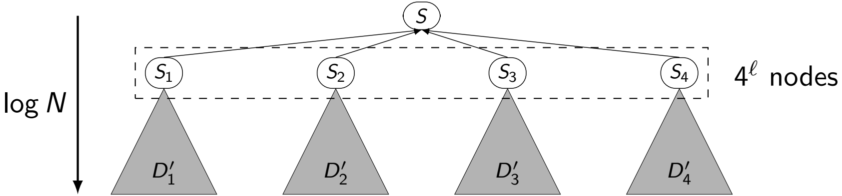
Two possible applications:

- Sparse RHS: since one RHS $= \mathcal{O}(1)$ branch of the separator tree, $\mathcal{G}(N)$ is the potential gain when exploiting sparsity.
- Tree parallelism: $\mathcal{G}(N)$ is a metric to measure parallelism.

Outline

Complexity study

Complexity on the separator tree



Solve phase and critical path:

Let m_ℓ be the size of frontal matrix at layer ℓ and $\mathcal{C}_{dense} = \Theta(m_\ell^\alpha)$ be the dense complexity of the solve:

$$\mathcal{C}(N) = \sum_\ell \#\text{nodes}_\ell \times \mathcal{C}_{dense}(m_\ell)$$

$$\mathcal{C}^c(N) = \sum_\ell \cancel{\#\text{nodes}_\ell} \times \mathcal{C}_{dense}(m_\ell)$$

Complexity on the separator tree

Nested dissection formulas 2D: $\#\text{nodes}_\ell = 4^\ell$, $m_\ell = N/2^\ell$.

$$\mathcal{C}(N) = \sum_{\ell=0}^{\log N} \Theta(4^\ell \times (N/2^\ell)^\alpha) \qquad = \Theta(N^\alpha \sum_{\ell=0}^{\log N} 2^{2-\alpha})$$

$$\mathcal{C}^c(N) = \sum_{\ell=0}^{\log N} \Theta(\cancel{4^\ell} \times (N/2^\ell)^\alpha) \qquad = \Theta(N^\alpha \sum_{\ell=0}^{\log N} 2^{-\alpha})$$

Depending on the values of α:

	$\mathcal{C}(N)$	$\mathcal{C}^c(N)$
FR $(\alpha = 2)$	$\Theta(N^2 \log N)$	$\Theta(N^2)$
BLR $(\alpha = 1.5)$	$\Theta(N^2)$	$\Theta(N^{1.5})$

Complexity analysis results for 2D regular problems.

Asymptotic complexity analysis

Same applies for 3D problems.

	$\mathcal{G}_{2D}(N)$	$\mathcal{G}_{3D}(N)$
FR $(\alpha = 2)$	$\Theta(\log N)$	$\Theta(1)$
BLR $(\alpha = 1.5)$	$\Theta(N^{1/2})$	$\Theta(\log N)$

Complexity analysis results for 2D and 3D regular problems.

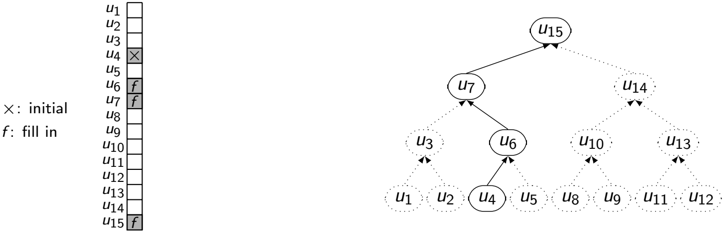
$\Rightarrow$ **Asymptotic value of $\mathcal{G}(N)$ increases more rapidely in BLR!**

Application

Exploiting RHS sparsity

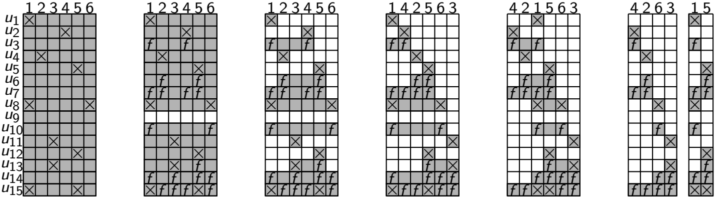
Theorem

Computation follows paths in the separator tree from active nodes to root. Each RHS requires to traverse one branch.



When sufficiently sparse, computation of RHS vector amounts to traverse $\Theta(1)$ branches.

Does this remain true with multiple RHS ?



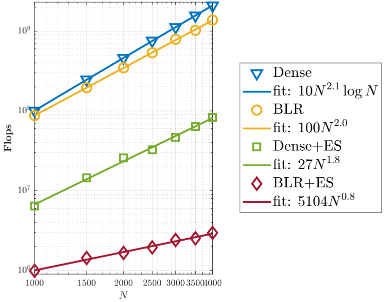
(a) `DEN` (b) `TP` (c) `INI` (d) `PO` (e) `FT` (f) `FT+BLK`

Toward an optimal number of operations[3]

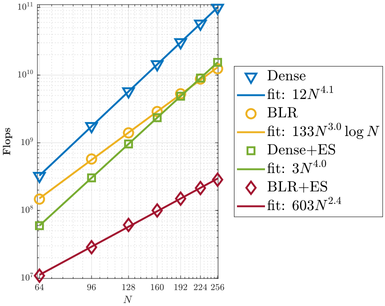
- **Vertical sparsity**: avoiding computation within columns;
- **Horizontal sparsity**: avoiding computation within rows;
- **Column ordering**: reducing interval sizes (Postorder or Flat Tree);
- **Blocking**: building minimal number of groups (`BLK`).

[3] **amlm:17**.

Configuration: one RHS with one nonzero.



(a) 2D Poisson problem.



(b) 3D Poisson problem.

Theory is confirmed by experimental results.

Asymptotic results were also confirmed for multiple RHS with multiple nonzeros.

Impact on real-life problems

	H3		H17		5Hz		10Hz	
OPS	FR	BLR	FR	BLR	FR	BLR	FR	BLR
DEN	72.01	36.8	813.41	286.15	15.51	12.85	184.68	117.77
ES	12.95	6.46	138.35	46.56	3.28	1.3	38.77	10.98
$\mathcal{G}(N)$	5.56	5.69	5.87	6.14	4.72	9.88	4.76	10.72

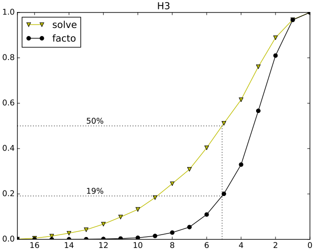
Number of operations ($\times 10^{12}$) of the forward elimination in BLR and FR.

	H3		H17		5Hz		10Hz	
T_f	FR	BLR	FR	BLR	FR	BLR	FR	BLR
DEN	377	273	3532	2008	50	43	456	251
ES	166	119	1339	630	23	16	186	85
$\mathcal{G}_t(N)$	2.27	2.29	2.63	3.18	2.17	2.69	2.45	2.95

Times (s) of the forward elimination in BLR and FR. 90 nodes.

$\Rightarrow$ **Potential gains from BLR and ES are both significative.**
However, $\mathcal{G}_t(N)$ does not follow $\mathcal{G}(N)$.

Tree parallelism



Distribution of operations in the separator tree.

$\mathcal{G}(N)$ is equivalent to theoretical speed up:

	Factorization $\mathcal{G}_{2D}(N)$	Factorization $\mathcal{G}_{3D}(N)$	Solve $\mathcal{G}_{2D}(N)$	Solve $\mathcal{G}_{3D}(N)$
FR	$\Theta(1)$	$\Theta(1)$	$\Theta(\log N)$	$\Theta(1)$
BLR	$\Theta(\log N)$	$\Theta(1)$	$\Theta(N^{1/2})$	$\Theta(\log N)$

Comparison with the factorization phase.

Consequences:

- more tree parallelism than factorization;
- should be taken into account in the design of parallel algorithms.

Conclusion

Solve phase

- Some applications are bounded by the solve time;
- More attention should be given to the solve phase.

Sparsity

- Exploiting sparsity becomes more efficient as the problem size grows.

Parallelism

- Exhibits more tree parallelism;
- Design solve-oriented algorithms.