Extensions of Certain Graph-based Algorithms for Preconditioning

David Fritzsche

Temple University

Bergische Universität Wuppertal

with

Andreas Frommer and Daniel B. Szyld

Joint GAMM-SIAM Conference on Applied Linear Algebra

July 24–27, 2006

Overview

We consider a nonsingular linear system

$$Ax = b$$

with $A \in \mathbb{R}^{n \times n}$, sparse and nonsymmetric.

In this talk:

- We solve $Ax = b$ iteratively (e.g. GMRES)
- block Jacobi / block Gauss-Seidel preconditioning
- **Main Topic:** An algorithm which finds (non-overlapping) sub-matrices, which are permuted to become diagonal blocks.

Outline

1. Graph-based permutations
2. Preprocessing
3. Numerical experiments

Basic Definitions

Definition: Let $A \in \mathbb{R}^{n \times n}$. Then $G(A) = (V, E)$ is the *directed* graph of A, i.e.

- vertices $V = \{1, \ldots, n\}$
- edges $E = \{(i, j) : a_{ij} \text{ is nonzero}\}$

Definition:

- Vertices i, j in $G(A)$ are adjacent if a_{ij} **or** a_{ji} is nonzero. (direction of edges not important)
- i adjacent to $W \subset V$ if i is adjacent to any $j \in W$

Finding "suitable" sub-matrices in A

$$\Longleftrightarrow$$

Finding "suitable" clusters of vertices in $G(A)$

Overview of PABLO

PABLO (O'Neil, Szyld 1990) is a simple algorithm to find dense blocks.

Basic Features

- Builds only one block at a time
- Considers only one vertex at a time for inclusion
- Checks all adjacent vertices for inclusion

Other Features

- Time complexity: $\mathcal{O}(n + nz(A))$. $\rightsquigarrow$ PABLO is fast
- Block sizes are not known a priori.
- The inclusion test can be varied
  $\rightsquigarrow$ different versions (PABLO, TPABLO, XPABLO)

Inclusion Test I: Structural Criteria

General Situation: Let B be the vertices currently in the block, i the eligible vertex (i.e., i adjacent to some $j \in B$)
We add i to B if the inclusion test $\tau(i)$ is true
Later: Definition of τ
Now: Basic criteria (predicates) used to define τ

Fullness Criterion (FC)
The "fullness" of $B \cup \{i\}$ is at least α times the fullness of P

$$\text{fullness} := \frac{\text{number of edges}}{\text{number of all possible edges}}$$

(typical values: $\alpha \geq 1$)

Connectivity Criterion (CC)
A fraction of β or more of all edges of i goes into B
(typical values: $\beta \geq 0.5$)

Inclusion Test II: Threshold Criteria

Observation: So far only the structure of A was considered

New: Fix *threshold parameters* δ and γ.

- Matrix entries $< \delta$ are ignored
  (considered to be zero by PABLO)
- Matrix entries $> \gamma$ are considered to be large.

Additionally to $G = G(A)$ PABLO also looks at

$$G^\gamma := G \setminus \{\text{edges with represents matrix entries } \leq \gamma\}$$

Threshold Fullness Criterion (TFC)
The fullness of $B \cup \{i\}$ in G^γ is at least θ.

Threshold Connectivity Criterion (TCC)
A fraction of at least ζ of the edges of i into B represent large entries.

Inclusion Test III: Combining Criteria

The four basic criteria (FC, CC, TFC, TCC) are logically combined into the inclusion test τ.

PABLO (O'Neil, Szyld 1990)

- $\tau = \text{FC} \vee \text{CC}$

TPABLO (Choi, Szyld 1996)

- TPABLO1: $\tau = (\text{FC} \vee \text{CC}) \wedge \text{TCC}$
- TPABLO2: $\tau = (\text{FC} \vee \text{CC}) \wedge \text{TFC}$ with $\theta = 1$

XPABLO (Fritzsche 2004)

- $\tau = \text{FC} \vee \text{TCC}$

Other Parameters

- *minbs*: Minimum size of a block
- *maxbs*: Maximum size of a block

PABLO based Preconditioners

Idea: Use the blocks found by PABLO for preconditioning.

- M_J :=Block Jacobi preconditioner
- M_G :=Block Gauss-Seidel preconditioner

Advantages

- M_J / M_G is "substantial" part of A if PABLO "worked"
- We can use dense BLAS/LAPACK subroutines

Jacobi or Gauss-Seidel

- In a non-parallel setting Gauss-Seidel has the same cost as Jacobi, but gives better convergence
- Block Jacobi preconditioning can be done in parallel
  $\rightsquigarrow$ may be better in some applications

Preprocessing: Scaling and Transversals

We need to preprocess the matrix:

- PABLO does not change the diagonal values
- Without scaling it is very difficult to find good threshold parameters.

We use

(MPS, MC64) (Duff, Koster 2001)
MC64 finds a nonsymmetric permutation P and a scaling (R, C), s.t.

$$(PRAC)_{ii} = 1 \qquad \text{for } i = 1, \ldots, n$$
$$(PRAC)_{ij} \leq 1 \qquad \text{for } i \neq j$$

Choosing the Parameters

The best choice for the parameters depends on the system to solve and can be quite different from the recommendations given here.

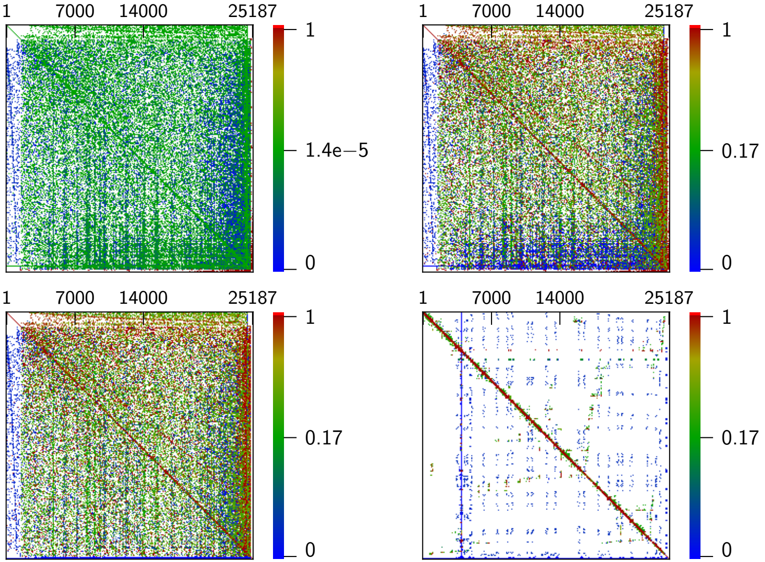
XPABLO default parameters

- $\tau = (\text{FC}) \vee (\text{TCC})$
- $\alpha = 1.1$ and $\zeta = 1/2n$
- $minbs = 200$ and $maxbs = 1000$
  ($minbs = 10$ and $maxbs = 50$ for dense LU)
- $\delta = 0.05$ and $\boldsymbol{\gamma = \sum_{i,j} |a_{ij}|/nz}$

Numerical Results I: Test Matrices

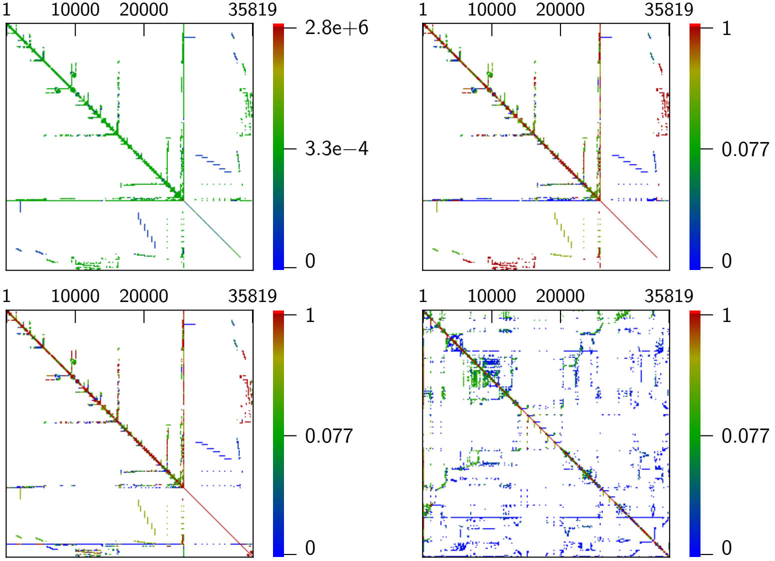
matrix	n	nz	application
CIRCUIT_4	80209	307604	circuit simulation
HCIRCUIT	105676	513072	circuit design
IGBT3	10938	130500	semiconductor device simulation
MEMPLUS	17758	99147	circuit design
MULT_DCOP_01	25187	193276	circuit simulation
MULT_DCOP_02	25187	193276	circuit simulation
MULT_DCOP_03	25187	193216	circuit simulation
NMOS3	18588	237130	semiconductor device simulation
SCIRCUIT	170998	958936	circuit design
M_XC	35819	188564	circuit simulation
M_XCRCX	119193	593608	circuit simulation

Spy Plots: MULT_DCOP_01



Top left: original. Top right: scaled (and not yet permuted) by MC64.
Bottom left: after MC64. Bottom right: after MC64 and XPABLO

Spy plots II: M_XC



Top left: original. Top right: scaled (and not yet permuted) by MC64.
Bottom left: after MC64. Bottom right: after MC64 and XPABLO

Measurements using GMRES(50)

	Times in seconds				Total Iterations			
	XPABLO		ILUT		XPABLO		ILUT	
Matrix	D	S	I1	I2	D	S	I1	I2
CIRCUIT_4	2.1	**0.89**	11	4.4	36	**14**	381	128
HCIRCUIT	3.4	1.3	0.92	**0.59**	40	**8**	20	**8**
IGBT3	3.9	1.2	0.34	**0.24**	577	171	73	**23**
MEMPLUS	0.19	0.19	0.15	**0.1**	12	**11**	31	13
MULT_DCOP_01	0.53	**0.47**	0.75	0.8	21	15	7	**5**
MULT_DCOP_02	0.49	**0.24**	0.68	0.69	26	8	7	**5**
MULT_DCOP_03	0.25	**0.2**	0.66	0.68	7	**5**	7	**5**
NMOS3	1.9	1.3	0.37	**0.35**	150	96	34	**17**
SCIRCUIT	–	43	40	**25**	–	425	467	**277**
M_XC	0.5	**0.48**	0.61	0.51	**16**	17	41	21
M_XCRCX	11	**2.9**	20	11	110	**33**	310	150

Preconditioners:

D: MC64 + XPABLO + block Gauss-Seidel + **dense LU**

S: MC64 + XPABLO + block Gauss-Seidel + **sparse LU**

I1: MC64 + RCM + ILUT$(10^{-2}, 5)$

I2: MC64 + RCM + ILUT$(10^{-3}, 10)$

Conclusions

- not a black-box method
- competitve with ILUT when good parameters can be predicted
- less overhead than ILUT
- easy parallelization with block Jacobi preconditioning

More Information

- `http://math.temple.edu/~daffi/pablo`
- Report coming soon