BFD: a preconditioning technique using block filtering decomposition

Riadh Fezzani

in collaboration with

L. Grigori and F. Nataf

Work performed in the context of
ANR Petalh project

http://petal.saclay.inria.fr/

Sparse Days
September 7, 2011

Plan

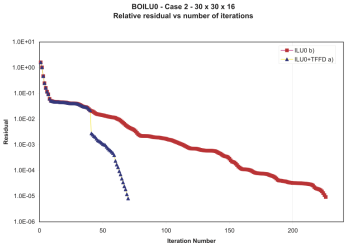
Motivation and challenges

Preconditionning problem

$$M^{-1}Ax = M^{-1}b$$

- iLU has scalability problems often due to few low eigenvalues.
- M satisfies a filtering property for a given vector t if

$$Mt = At \quad \text{or} \quad t^T M = t^T A.$$



- Filtering the low frequencies accelerate the convergence [Achdou and Nataf, 2007].

Challenge

Extending the existing filtering decomposition for block tridiagonal matrices to general matrices.

Block tridiagonal matrices

- Let A be a block tridiagonal matrix,
- the diagonal blocks are square, but not necessarily of a same size :

$$A = \begin{pmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & \ddots & \\ & \ddots & \ddots & A_{N-1,N} \\ & & A_{N,N-1} & A_{NN} \end{pmatrix}$$

Exact factorization of block tridiagonal matrices

Exact block *LDU* factorization of A

$$A = \begin{pmatrix} D_{11} & & & \\ A_{21} & D_{22} & & \\ & \ddots & \ddots & \\ & & A_{N,N-1} & D_{NN} \end{pmatrix} \begin{pmatrix} D_{11}^{-1} & & & \\ & D_{22}^{-1} & & \\ & & \ddots & \\ & & & D_{NN}^{-1} \end{pmatrix} \begin{pmatrix} D_{11} & A_{12} & & \\ & D_{22} & \ddots & \\ & & \ddots & A_{N-1,N} \\ & & & D_{NN} \end{pmatrix}$$

where D_{ii} are invertible matrices defined as

$$D_{ii} = \begin{cases} A_{11} & i = 1 \\ A_{ii} - A_{i,i-1} D_{i-1,i-1}^{-1} A_{i-1,i} & i > 1 \end{cases}$$

Filtering decomposition for block tridiagonal matrices

Filtering decomposition of the preconditionner M

$$M = \begin{pmatrix} \bar{D}_{11} & & & \\ A_{21} & \bar{D}_{22} & & \\ & \ddots & \ddots & \\ & & A_{N,N-1} & \bar{D}_{NN} \end{pmatrix} \begin{pmatrix} \bar{D}_{11}^{-1} & & & \\ & \bar{D}_{22}^{-1} & & \\ & & \ddots & \\ & & & \bar{D}_{NN}^{-1} \end{pmatrix} \begin{pmatrix} \bar{D}_{11} & A_{12} & & \\ & \bar{D}_{22} & \ddots & \\ & & \ddots & A_{N-1,N} \\ & & & \bar{D}_{NN} \end{pmatrix}$$

where $\bar{D}_{ii}$ are invertible matrices defined as [Achdou and Nataf, 2007]

$$\bar{D}_{ii} = \begin{cases} A_{11} & i = 1 \\ A_{ii} - A_{i,i-1}(2\beta_{i-1} - \beta_{i-1} D_{i-1,i-1} \beta_{i-1}) A_{i-1,i} & i > 1 \end{cases}$$

Different approximation matrices β_{i-1} can be used

- Wagner and Wittum [1997] mainly for symmetric matrices,
- Nested factorization [Appleyard and Cheshire, 1983] uses $t = (1, 1, \cdots, 1)^T$.

Filtering decomposition for block tridiagonal matrices

Pros

- Very efficient for classes of problems, e.g. convection-diffusion equations with heterogeneous and/or anisotropic diffusion tensors,
- alleviate the effect of low frequency modes by a judicious choice of filter vector,
- similarities with deflation techniques.

Cons

- Restricted to sequential computation and mainly structured grids,
- dedicated to block tridiagonal matrices.

Arbitrary matrices

- Let A be partitioned into a block matrix of size $N \times N$,
- the diagonal blocks are square, but not necessarily of a same size.

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{pmatrix}$$

- The generalization of filtering preconditioner to arbitrary matrices is the way to go towards parallel computation.

Arbitrary matrices

- Let A be partitioned into a block matrix of size $N \times N$,
- the diagonal blocks are square, but not necessarily of a same size.

$$A = \begin{pmatrix} A_{11} & \cdots & A_{1N} \\ \vdots & \ddots & \vdots \\ A_{N1} & \cdots & A_{NN} \end{pmatrix}$$

- The generalization of filtering preconditioner to arbitrary matrices is the way to go towards parallel computation.

Exact factorization of arbitrary matrices

Exact Block *LDU* factorization of *A*

$$A = (L+D)D^{-1}(D+U)$$

$$= \begin{pmatrix} D_{11} & & & \\ L_{21} & D_{22} & & \\ \vdots & \ddots & \ddots & \\ L_{N1} & \cdots & L_{N,N-1} & D_{NN} \end{pmatrix} \begin{pmatrix} D_{11}^{-1} & & & \\ & D_{22}^{-1} & & \\ & & \ddots & \\ & & & D_{NN}^{-1} \end{pmatrix} \begin{pmatrix} D_{11} & U_{12} & \cdots & U_{1N} \\ & D_{22} & \ddots & \vdots \\ & & \ddots & U_{N-1,N} \\ & & & D_{NN} \end{pmatrix}$$

Recurcive computation

Let $C = L + D + U$. Each block of L,D,U is computed as

$$C_{i,j} = \begin{cases} A_{ij} & i = 1 \text{ or } j = 1 \\ A_{ij} - \sum_{k=1, L_{ik}\neq 0, U_{kj}\neq 0}^{min(i,j)-1} L_{ik} D_{kk}^{-1} U_{kj} & i > 1 \text{ or } j > 1 \end{cases}$$

Block Filtering Decomposition (BFD)

- The BFD preconditionner M is written as

$$M = (\bar{L} + \bar{D})\bar{D}^{-1}(\bar{D} + \bar{U})$$

- Let $\bar{C} = \bar{U} + \bar{D} + \bar{U}$ with

$$\bar{C}_{ij} = \begin{cases} A_{ij} & i = 1 \text{ or } j = 1 \\ A_{ij} - \displaystyle\sum_{k=1, L_{ik} \neq 0, U_{kj} \neq 0}^{min(i,j)-1} L_{ik} F_{kj} U_{kj} & i > 1 \text{ or } j > 1 \end{cases}$$

where F_{kj} is a sparse approximation of $\bar{D}_{kk}^{-1}$.

Construction of approximation matrices

Given a filtering vector t, filtering is ensured if

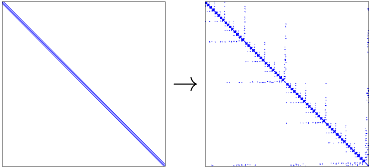
$$F_{kj}\bar{U}_{kj}t_j = \bar{D}_{kk}^{-1}\bar{U}_{kj}t_j$$

Example

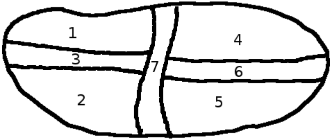
$$F_{kj} = Diag\left(\left(\bar{D}_{kk}^{-1}\bar{U}_{kj}t_j \cdot / \bar{U}_{kj}t_j\right)\right)$$

Suitability for parallel computing

- A sparse matrix can be permuted using nested dissection (metis, scotch) :



- Systems from domain decomposition methods :



$$\rightarrow \begin{pmatrix} A_{11} & & A_{13} & & & & A_{17} \\ & A_{22} & A_{23} & & & & A_{27} \\ A_{31} & A_{32} & A_{33} & & & & A_{37} \\ & & & A_{44} & & A_{46} & A_{47} \\ & & & & A_{55} & A_{56} & A_{57} \\ & & & A_{64} & A_{65} & A_{66} & A_{67} \\ A_{71} & A_{72} & A_{73} & A_{74} & A_{75} & A_{76} & A_{77} \end{pmatrix}$$

Test matrices

Boundary value problem (provided by Achdou, Nataf)

$$\begin{aligned} div(a(x)u) - div(\kappa(x)\nabla u) &= f && \text{in } \Omega \\ u &= 0 && \text{on } \partial\Omega_D \\ \frac{\partial u}{\partial n} &= 0 && \text{on } \partial\Omega_N \end{aligned}$$

where $\Omega = [0,1]^2, \partial\Omega_D = [0,1] \times \{0,1\}, \partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$ discretized on a cartesian grid.

3D computations for Black Oil Model (provided by R. Masson IFP)

Simulation of reservoir models : compositional triphase Darcy flow simulator (oil, water and gas).

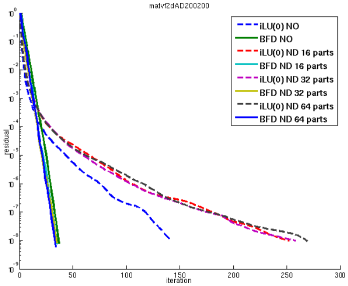
Experience condition

- All tests use GMRES (tolerance = 10^{-8}, maxit = 1000, restart =200),
- computed error = $\frac{\|sol - sol_{exact}\|}{\|sol_{exact}\|}$
- different mothods compared :
  - iLU(0),
  - iLU(τ),
  - BFD.
- Matrices are used with
  - natural order (NO) - sequential,
  - nested dissection (ND) with 16, 32 and 64 parts - parallel.

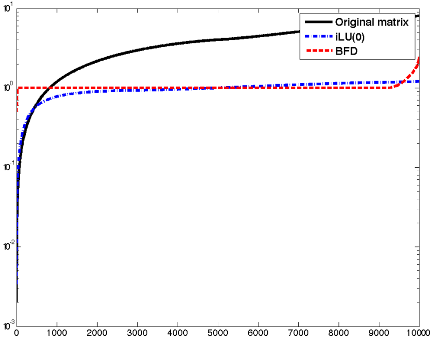
2D advection-diffusion with rotating velocity

- The tensor κ is the identity.
- Velocity $a = \left(2\pi(x_2 - 0.5), 2\pi(x_1 - 0.5)\right)^T$.

	iLU(0)		BFD	
	Iter	Error	Iter	Error
N.O.	141	E-6	**36**	E-9
N.D. 16	251	E-7	**34**	E-8
N.D. 32	257	E-5	**34**	E-8
N.D. 64	268	E-5	**33**	E-8



Effect on eigenvalue distribution

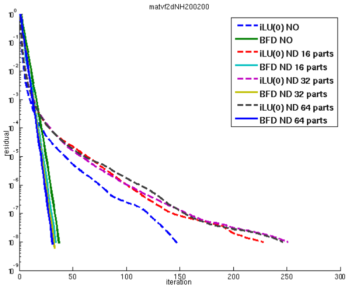


Same advection-diffusion problem, discretized on agrid with 100 $\times$ 100 nodes.

Non-homogeneous problem

- The tensor κ is isotropic and discontinuous. It jumps from the constant value 10^3 in the ring $\frac{1}{2\sqrt{2}} \leq |x - c| \leq \frac{1}{2}$ to 1 outside.
- Uniform grid with 200×200 nodes.

	iLU(0)		BFD	
	Iter	Error	Iter	Error
N.O.	146	E-6	**36**	E-9
N.D. 16	227	E-5	**33**	E-7
N.D. 32	250	E-5	**32**	E-8
N.D. 64	245	E-5	**30**	E-8



Black Oil

- 3D computation : $60 \times 60 \times 32$ cells.
- Permeability highly heterogeneous (jumps on the order of 2^8).

	N.D. 16			N.D. 32		
	Iter	Error	Fill	Iter	Error	Fill
iLU(0)	850	E-4	1	554	E-4	1
iLU(10^{-2})	147	E-6	2.4	144	E-6	2.4
iLU(10^{-3})	57	E-7	5.1	60	E-7	5.1
iLU(10^{-4})	24	E-8	10	24	E-8	10
BFD	90	E-7	3.9	75	E-6	6.5

Conclusions and future work

Block Filtering Decomposition

- accepts any input matrix,
- Schur complement approximation can be adapted to the user needs :
  - preserve symetry,
  - less filling...

Future work : go toward black-box method

- Build approximation strategies to deal with hard cases :
  - some of the domains are close to singularity,
  - develop reordering techniques that help alleviate this problem.
- Introduce parameters.

Références

Y. Achdou and F. Nataf. Low frequency tangential filtering decomposition. *Numerical Linear Algebra with Applications*, 14(2) :129–147, 2007.

JR Appleyard and IM Cheshire. Nested factorization. Technical report, Exploration Consultants Ltd., 1983.

C. Wagner and G. Wittum. Adaptive filtering. *Numerische Mathematik*, 78(2) : 305–328, 1997.