On the Usage of Triangular Preconditioner Updates in Matrix-Free Environment

Jurjen Duintjer Tebbens

Institute of Computer Science

Academy of Sciences of the Czech Republic

joint work with

Miroslav Tůma

Institute of Computer Science

Academy of Sciences of the Czech Republic

and Technical University in Liberec

5th PMAA workshop, Neuchâtel, June 21, 2008.

1. Introduction

- The solution of *sequences* of linear systems arises in numerous applications

- Rather few work has been done in our community on efficient solution of *general* sequences of linear systems

- The central question of such work will be:

  How can we share part of the computational effort throughout the sequence ?

- Below we list some known strategies.

1. Introduction

- Very simple trick: Hot starts, i.e. use the solution of the previous system as initial guess.

- Sometimes exact updating of the factorizations for large problems is feasible: Rank-one updates of LU factorizations have been used since decades in the simplex method where the change of one system matrix to another is restricted to one column [Bartels, Golub, Saunders - 1970; Suhl, Suhl - 1993].

- General rank-one updates of an LU decomposition are discussed in [Stange, Griewank, Bollhoefer - 2005].

- If the linear solver is a Krylov subspace method, strategies to recycle information gained from previously generated Krylov subspaces have shown to be beneficial in many applications [Parks, de Sturler, Mackey, Johnson, Maiti - 2006], [Giraud, Gratton, Martin - 2007], [Frank, Vuik - 2001].

1. Introduction

- When shifted linear systems with identical right hand sides have to be solved, Krylov subspace methods allow advantageous implementations based on the fact that all systems generate the same subspace [Frommer, Glässner - 1998]

- In nonlinear systems solved with a Newton-type method one can skip evaluations of the (approximate) Jacobian during some iterations, leading to Shamanskii's combination of the chord and Newton method [Brent - 1973] $\Rightarrow$ linear solving techniques with multiple right hand sides can be exploited.

- Another option: Allow changing the system matrices but freeze the preconditioner [Knoll, Keyes - 2004].

1. Introduction

To enhance the power of a frozen preconditioner one may also use approximate updates:

- In [Meurant - 2001] we find approximate updates of incomplete Cholesky factorizations and
- in [Benzi, Bertaccini - 2003, 2004] banded updates were proposed for both symmetric positive definite approximate inverse and incomplete Cholesky preconditioners.
- In Quasi-Newton methods the difference between system matrices is of small rank and preconditioners may be efficiently adapted with approximate small-rank updates; this has been done in the symmetric positive definite case, see e.g. [Bergamaschi, Bru, Martínez, Putti - 2006, Nocedal, Morales - 2000].

2. The proposed preconditioner updates

- We focus on a black-box approximate preconditioner update for *general nonsymmetric* systems solved by arbitrary iterative methods.
- Updating frozen preconditioners for preconditioned iterative methods instead of their recomputation.
- Simple algebraic updates which might be applied also in matrix-free environment

Notation: Consider two systems

$$Ax = b, \quad A^+x^+ = b^+; \text{ preconditioned by } M,\ M^+,$$

$$\text{let} \qquad B \equiv A - A^+.$$

We would like the update M^+ to become as powerful as M.

2. The proposed preconditioner updates

If $||A - M||$ is the accuracy of the preconditioner M for A, we will try to find an updated M^+ for A^+ with comparable accuracy,

$$||A - M|| \approx ||A^+ - M^+||.$$

Let M be factorized as $M = LDU$, then the choice

$$M^+ = LDU - B$$

would give $||A - M|| = ||A^+ - M^+||$. We will approximate this ideal update $LDU - B$ in two steps, similarly to the techniques in [Benzi, Bertaccini - 2003, Bertaccini - 2004]. First we use

$$LDU - B = L(DU - L^{-1}B) \approx L(DU - B) \quad \text{or}$$

$$LDU - B = (LD - BU^{-1})U \approx (LD - B)U$$

depending on whether L is closer to identity or U.

2. The proposed preconditioner updates

Define the standard splitting

$$B = L_B + D_B + U_B.$$

Then the second approximation step is

$$L(DU - B) \approx L(DU - D_B - U_B) \equiv M^+$$

(upper triangular update) or

$$(LD - B)U \approx (LD - L_B - D_B)U \equiv M^+$$

(lower triangular update). Then M^+ is for free and its application asks for one forward and one backward solve.

- Ideal for upwind/downwind modifications
- Our experiments cover broader spectrum of problems

2. The proposed preconditioner updates

As an example consider a two-dimensional nonlinear convection-diffusion model problem: It has the form

$$-\Delta u + Ru\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) = 2000x(1-x)y(1-y), \tag{1}$$

on the unit square, discretized by 5-point finite differences on a uniform grid.

- The initial approximation is the discretization of $u_0(x, y) = 0$.
- We use here $R = 100$ and different grid sizes.
- We solve the resulting linear systems with BiCGSTAB with right preconditioning.
- Iterations were stopped when the Euclidean norm of the residual was decreased by seven orders.
- Matlab implementation.

2. The proposed preconditioner updates

BiCGSTAB iteration counts; reference factorization ILU(0)			
Matrix	Recomp	Freeze	Triangular update
$A^{(0)}$	40	40	40
$A^{(1)}$	25	37	37
$A^{(2)}$	24	41	27
$A^{(3)}$	20	48	26
$A^{(4)}$	17	56	30
$A^{(5)}$	16	85	32
$A^{(6)}$	15	97	35
$A^{(7)}$	14	106	43
$A^{(8)}$	13	97	44
$A^{(9)}$	13	108	45
$A^{(10)}$	13	94	50
$A^{(11)}$	15	104	45
$A^{(12)}$	13	156	49
overall time	13 s	13 s	7.5 s

3. Updates in matrix-free environment

In many applications the linear solver is chosen such that it is not necessary to store system matrices; a matrix-vector product subroutine suffices.

Standard example: Newton iteration of the form

$$J(x_k)(x_{k+1} - x_k) = -F(x_k), \quad k = 1, 2, \ldots$$

where $J(x_k)$ is the (approximate) Jacobian of F evaluated at x_k. Then a matvec with $J(x_k)$ is replaced by the finite difference approximation,

$$J(x_k) \cdot v \approx \frac{F(x_k + h\|x_k\|v) - F(x_k)}{h\|x_k\|},$$

for some small h.

Can the preconditioner updates be used in matrix-free environment ?

3. Updates in matrix-free environment

First note that to compute an incomplete factorization like ILU in matrix-free environment at all, the system matrix has to be *estimated by matvecs*.

The entries of a simple tridiagonal matrix

$$\begin{pmatrix} * & * & & & & \\ * & * & * & & & \\ & * & * & * & & \\ & & \ddots & \ddots & \ddots & \\ & & & * & * & * \\ & & & & * & * \end{pmatrix}$$

can be easily obtained with matvecs with $(1,0,0,1,0,0,\dots)^T$, $(0,1,0,0,1,0,\dots)^T$ and $(0,0,1,0,0,1,\dots)^T$.

3. Updates in matrix-free environment

In general, one uses a graph coloring algorithm that tries to minimize the number of matvecs for a good estimate [Cullum, Tůma - 2006].

Hence recomputing the preconditioner requires for every linear system:

- A number of additional matvecs to estimate the current matrix
- When the nonzero pattern changes during the sequence: Rerunning the graph coloring algorithm.

Preconditioner recomputation is even more expensive in matrix-free environment !

How about the preconditioner updates ?

3. Updates in matrix-free environment

Recall the upper triangular update is of the form

$$M^+ = L(DU - D_B - U_B)$$

based on the splitting

$$L_B + D_B + U_B = B = A - A^+.$$

Thus the update somehow needs A and A^+.

However:

- A has been estimated before to obtain the reference ILU-factorization
- Of A^+ we need estimate only the upper triangular part
- Estimating $triu(A^+)$ with graph colouring techniques may be much less expensive than estimating A^+

3. Updates in matrix-free environment

The problem of estimating only the upper triangular part leads to a *partial graph coloring problem* [Pothen et al. - 2007].

Example:

$$\begin{pmatrix} * & & & & * \\ & * & & & * \\ & & \ddots & & \vdots \\ & & & \ddots & * \\ * & * & * & * & * \end{pmatrix}$$

Estimation of the whole matrix asks for n matvecs with all unit vectors e_i, but estimating the upper triangular part requires only 2 matvecs, $(1, \ldots, 1, 0)^T$ and e_n.

3. Updates in matrix-free environment

Theoretical explanation:

The graph coloring algorithm for A works on the *intersection graph*

$$G(A^T A).$$

For $triu(A)$ we can prove:

The graph coloring algorithm for $triu(A)$ works on

$$G(triu(A)^T triu(A)) \cup G_K,$$

where

$$G_K = \cup_{i=1}^n G_{L_i, U_i},$$

with $L_i = \{v_j | a_{ij} \neq 0 \wedge j \leq i\}$ and $U_i = \{v_j | a_{ij} \neq 0 \wedge j > i\}$.

3. Updates in matrix-free environment

An alternative strategy circumvents any estimation of A^+:

Let the matvec be replaced with a function evaluation

$$A^+ \cdot v \quad \rightarrow \quad F^+(v), \quad F^+ : \mathbb{R}^n \rightarrow \mathbb{R}^n,$$

e.g. in Newton's method

$$J(x^+) \cdot v \quad \approx \quad \frac{F(x^+ + h\|x^+\|v) - F(x^+)}{h\|x^+\|} \equiv F^+(v).$$

We assume it is possible to compute the components $F_i^+ : \mathbb{R}^n \rightarrow \mathbb{R}$ of F^+ individually,

$$e_i^T \cdot A^+ \cdot v \quad \rightarrow \quad F_i^+(v), \quad F_i^+ : \mathbb{R}^n \rightarrow \mathbb{R}.$$

Then:

- The forward solves with L in $M^+ = L(DU - D_B - U_B)$ are trivial.

3. Updates in matrix-free environment

- For the backward solves, use a mixed explicit-implicit strategy: Split

$$DU - D_B - U_B = DU - triu(A) + triu(A^+)$$

in the explicitly given part

$$X \equiv DU - triu(A)$$

and the implicit part $triu(A^+)$.

We then have to solve the triangular systems

$$\left(X + triu(A^+)\right) z = y,$$

yielding the standard backward substitution cycle

$$z_i = \frac{y_i - \sum_{j>i} x_{ij} z_j - \sum_{j>i} a^+_{ij} z_j}{x_{ii} + a^+_{ii}}, \qquad i = n, n-1, \ldots, 1.$$

4. Updates in matrix-free environment: I

In

$$z_i = \frac{y_i - \sum_{j>i} x_{ij} z_j - \sum_{j>i} a^+_{ij} z_j}{x_{ii} + a^+_{ii}}, \qquad i = n, n-1, \ldots, 1.$$

the sum $\sum_{j>i} a^+_{ij} z_j$ can be computed by the function evaluation

$$\sum_{j>i} a^+_{ij} z_j = F^+_i \left((0, \ldots, 0, z_{i+1}, \ldots, z_n)^T \right). \tag{2}$$

The diagonal $\{a^+_{11}, \ldots, a^+_{nn}\}$ can be found by computing

$$a^+_{ii} = F^+_i(e_i), \qquad 1 \leq i \leq n.$$

If it is necessary to avoid the storage of A, one can also replace the part $triu(A)$ in

$$DU - triu(A) + triu(A^+)$$

by analogue implicit solves.

4. Updates in matrix-free environment: I

Cost comparison with explicitly given matrices

	computational costs	storage costs
recomputation	factorization	A^+, current L, U
updating	0	A^+ + $triu(A)$, reference L, U

Cost comparison in matrix-free environment with update estimating

	computational costs	storage costs
recomputation	$est(A^+)$ + factorization	current L, U
updating	$est(triu(A^+))$	$triu(A^+)$, $triu(A)$, reference L, U

Cost comparison in matrix-free environment without update estimation

	computational costs	storage costs
recomputation	$est(A^+)$ + factorization	current L, U
updating	0	reference L, U

4. Updates in matrix-free environment: I

The technique without update estimation seems superior but note: The mixed explicit-implicit backward solve

$$z_i = \frac{y_i - \sum_{j>i} x_{ij} z_j - \sum_{j>i} a^+_{ij} z_j}{x_{ii} + a^+_{ii}}, \qquad i = n, n-1, \dots, 1.$$

admits no merging of entries on the same position in X and $triu(A^+)$ by explicitly computing $X + triu(A^+)$, making the solves more expensive.

In general the choice between estimating $triu(A^+)$ or not is a trade-off between

- estimation costs (graph coloring algorithm and matvecs)
- amount of overlap
- function evaluation cost
- available storage space.

If minimal storage cost is the goal, then it should be avoided to estimate $triu(A^+)$.

5. Conclusions, future work

- Our black-box preconditioner update for general nonsymmetric sequences can be applied in matrix-free environment
- The costs difference between recomputing and updating is even larger in matrix-free environment than when matrices are explicitly given
- Experiments are for the moment missing but we have a good impression on overall costs for individual techniques
- Future work includes implementation of the techniques, permutations to enhance triangular dominance

For more details see:

- Birken Ph, Duintjer Tebbens J, Meister A, Tůma M: *Preconditioner Updates Applied to CFD Model Problems*, published online in Applied Numerical Mathematics in October 2007.
- Duintjer Tebbens J, Tůma M: *Improving Triangular Preconditioner Updates for Nonsymmetric Linear Systems*, LNCS vol. 4818, pp. 737–744, 2007.
- Duintjer Tebbens J, Tůma M: *Efficient Preconditioning of Sequences of Nonsymmetric Linear Systems*, SIAM J. Sci. Comput., vol. 29, no. 5, pp. 1918–1941, 2007.

Thank you for your attention.

Supported by the project “Information Society” of the Academy of Sciences of the Czech Republic under No. 1ET400300415