CORTONA 2004

Updating incomplete factorizations for PDEs

DANIELE BERTACCINI

Università di Roma "La Sapienza", Dipartimento di Matematica

web: http://www.mat.uniroma1.it/∼bertaccini

Large and Sparse Linear Systems

Many algorithms in scientific computing require the solution of **large and sparse linear systems**.

Large and Sparse Linear Systems

Many algorithms in scientific computing require the solution of **large and sparse linear systems**.

Sometimes **sequences of large and sparse linear systems** need to be solved. If they share something, it would be nice to use the information from the solution of one of them for the solution of the others. But how?

Large and Sparse Linear Systems

Many algorithms in scientific computing require the solution of **large and sparse linear systems**.

Sometimes **sequences of large and sparse linear systems** need to be solved. If they share something, it would be nice to use the information from the solution of one of them for the solution of the others. But how?

In this setting we would like to update/downdate incomplete factorizations of the underlying matrices.

The problem

The numerical solution of several problems in scientific computing requires the solution of sequences of parametrized large and sparse linear systems of the form

$$A_j x_j = b_j, \quad A_j = A + \alpha_j E_j, \quad j = 0, ..., s$$

where $\alpha_j \in \mathbb{C}$ are scalar quantities and E_0,..., E_s are (real or complex) symmetric matrices. Here we will consider the case where E_j, $j = 0, ..., s$ are diagonal matrices.

The problem

The numerical solution of several problems in scientific computing requires the solution of sequences of parametrized large and sparse linear systems of the form

$$A_j x_j = b_j, \quad A_j = A + \alpha_j E_j, \quad j = 0, ..., s$$

where $\alpha_j \in \mathbb{C}$ are scalar quantities and E_0,..., E_s are (real or complex) symmetric matrices. Here we will consider the case where E_j, $j = 0, ..., s$ are diagonal matrices.

Let us solve the underlying linear systems by an iterative method. Assume that an incomplete factorization P is initially computed for the *seed* matrix A.

The problem

The numerical solution of several problems in scientific computing requires the solution of sequences of parametrized large and sparse linear systems of the form

$$A_j x_j = b_j, \quad A_j = A + \alpha_j E_j, \quad j = 0, ..., s$$

where $\alpha_j \in \mathbb{C}$ are scalar quantities and E_0,..., E_s are (real or complex) symmetric matrices. Here we will consider the case where E_j, $j = 0, ..., s$ are diagonal matrices.

Let us solve the underlying linear systems by an iterative method. Assume that an incomplete factorization P is initially computed for the *seed* matrix A.

How to compute a new incomplete factorization $P_{\alpha,E}$ for $A+\alpha E$, $\alpha \in \mathbb{C}$ and E diagonal with complex entries (say) ?

The problem

The numerical solution of several problems in scientific computing requires the solution of sequences of parametrized large and sparse linear systems of the form

$$A_j x_j = b_j, \quad A_j = A + \alpha_j E_j, \quad j = 0, ..., s$$

where $\alpha_j \in \mathbb{C}$ are scalar quantities and E_0,..., E_s are (real or complex) symmetric matrices. Here we will consider the case where E_j, $j = 0, ..., s$ are diagonal matrices.

Let us solve the underlying linear systems by an iterative method. Assume that an incomplete factorization P is initially computed for the *seed* matrix A.

How to compute a new incomplete factorization $P_{\alpha,E}$ for $A+\alpha E$, $\alpha \in \mathbb{C}$ and E diagonal with complex entries (say) ?

We consider the seed matrix A SPD and incomplete factorizations (ILUT/ILDL^T) and sparse approximate inverses (AINV) for P (P^{-1}).

Preconditioning

For large and sparse problems, iterative methods are mandatory. Unfortunately, in many interesting frameworks, convergence of iterative methods can be very slow $\rightarrow$ **preconditioning** is the right way to go.

Preconditioning

For large and sparse problems, iterative methods are mandatory. Unfortunately, in many interesting frameworks, convergence of iterative methods can be very slow $\rightarrow$ **preconditioning** is the right way to go.

Recall: preconditioning with P means replacing $Ax = b$ by
$P^{-1}Ax = P^{-1}b$ (left prec.)
or by
$AP^{-1}y = b$, $y = Px$ (right prec.)

Preconditioning

For large and sparse problems, iterative methods are mandatory. Unfortunately, in many interesting frameworks, convergence of iterative methods can be very slow $\rightarrow$ **preconditioning** is the right way to go.

Recall: preconditioning with P means replacing $Ax = b$ by
$P^{-1}Ax = P^{-1}b$ (left prec.)
or by
$AP^{-1}y = b$, $y = Px$ (right prec.)
or by
$L^{-1}AL^{-T}u = L^{-1}b$, $x = L^{-T}u$ (split prec., here $P = LL^T$)

Preconditioning means solving a (matematically) equivalent linear system.

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;
- Levenberg-Marquardt methods for ill-posed quasi-Newton steps;
- Model trust region and globalized Newton algorithms;

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;
- Levenberg-Marquardt methods for ill-posed quasi-Newton steps;
- Model trust region and globalized Newton algorithms;
- solution of ill-posed least squares problems by Tikhonov–like regularization strategies;

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;
- Levenberg-Marquardt methods for ill-posed quasi-Newton steps;
- Model trust region and globalized Newton algorithms;
- solution of ill-posed least squares problems by Tikhonov–like regularization strategies;
- Iterative methods for Total least square problems (TLS);

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;
- Levenberg-Marquardt methods for ill-posed quasi-Newton steps;
- Model trust region and globalized Newton algorithms;
- solution of ill-posed least squares problems by Tikhonov–like regularization strategies;
- Iterative methods for Total least square problems (TLS);
- Shift-and-invert and Jacobi–Davidson algorithms for large-scale eigenvalue calculations;

Motivations

Parametric linear systems arise in:

- Solution of time-dependent PDEs / ODEs / Helmholtz eq. / etc.;
- Levenberg-Marquardt methods for ill-posed quasi-Newton steps;
- Model trust region and globalized Newton algorithms;
- solution of ill-posed least squares problems by Tikhonov–like regularization strategies;
- Iterative methods for Total least square problems (TLS);
- Shift-and-invert and Jacobi–Davidson algorithms for large-scale eigenvalue calculations;
- Control theory;
- ...and many others.

Is preconditioning always necessary ?

Assume now that $E = I$ and A has been normalized in such a a way that the largest entry in A is equal to 1.

Indeed, denoting by $\lambda_{\min}$ and $\lambda_{\max}$ the extremal eigenvalues of A, we have that

$$\kappa_2(A + \alpha\, I) = \frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} \leq \frac{\lambda_{\max}}{\alpha} + 1,$$

and in practice preconditioning is no longer necessary (or beneficial) as soon as $\lambda_{\max}/\alpha$ is small enough.

Is preconditioning always necessary ?

Assume now that $E = I$ and A has been normalized in such a a way that the largest entry in A is equal to 1.

Indeed, denoting by $\lambda_{\min}$ and $\lambda_{\max}$ the extremal eigenvalues of A, we have that

$$\kappa_2(A + \alpha\, I) = \frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} \leq \frac{\lambda_{\max}}{\alpha} + 1,$$

and in practice preconditioning is no longer necessary (or beneficial) as soon as $\lambda_{\max}/\alpha$ is small enough.

At the other extreme, continuity suggests that there is a value of α under which one might as well reuse the preconditioner computed for the original A.

Is preconditioning always necessary ?

Assume now that $E = I$ and A has been normalized in such a a way that the largest entry in A is equal to 1.

Indeed, denoting by $\lambda_{\min}$ and $\lambda_{\max}$ the extremal eigenvalues of A, we have that

$$\kappa_2(A + \alpha\, I) = \frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} \leq \frac{\lambda_{\max}}{\alpha} + 1,$$

and in practice preconditioning is no longer necessary (or beneficial) as soon as $\lambda_{\max}/\alpha$ is small enough.

At the other extreme, continuity suggests that there is a value of α under which one might as well reuse the preconditioner computed for the original A.

However, in our experiments we found cases where reusing a preconditioner for A SPD, $E = I$ and real α gives poor results already for α as small as $\mathcal{O}(10^{-5})$ (entries of A normalized to 1 and $0 < \lambda(A) \leq 1$). [Benzi,B., BIT'03]

Is preconditioning always necessary ?

Assume now that $E = I$ and A has been normalized in such a a way that the largest entry in A is equal to 1.

Indeed, denoting by $\lambda_{\min}$ and $\lambda_{\max}$ the extremal eigenvalues of A, we have that

$$\kappa_2(A + \alpha\, I) = \frac{\lambda_{\max} + \alpha}{\lambda_{\min} + \alpha} \leq \frac{\lambda_{\max}}{\alpha} + 1,$$

and in practice preconditioning is no longer necessary (or beneficial) as soon as $\lambda_{\max}/\alpha$ is small enough.

At the other extreme, continuity suggests that there is a value of α under which one might as well reuse the preconditioner computed for the original A.

However, in our experiments we found cases where reusing a preconditioner for A SPD, $E = I$ and real α gives poor results already for α as small as $\mathcal{O}(10^{-5})$ (entries of A normalized to 1 and $0 < \lambda(A) \leq 1$). [Benzi,B., BIT'03]

$\Rightarrow$there is a fairly broad range of values of α where modification strategies are of potential benefit

Example: Time-dependent partial differential equations

$$\frac{\partial u}{\partial t} = \nabla \cdot (a\nabla\, u) + f$$

simple linear diffusion problem on a plane region, Dirichlet boundary conditions, initial condition $u(x, 0) = u_0(x)$. Discretization in space with stepsize h and an implicit (e.g., backward Euler) time discretization with time step τ results in a sequence of linear systems

$$(I + \frac{\tau}{h^2} A) u^{m+1} = u^m + \tau f^{m+1}, \quad m = 0, 1, 2, \ldots, M$$

where A is SPD.

Example: Time-dependent partial differential equations

$$\frac{\partial u}{\partial t} = \nabla \cdot (a \nabla u) + f$$

simple linear diffusion problem on a plane region, Dirichlet boundary conditions, initial condition $u(x,0) = u_0(x)$. Discretization in space with stepsize h and an implicit (e.g., backward Euler) time discretization with time step τ_m results in a sequence of linear systems

$$(I + \frac{\tau_m}{h^2} A) u^{m+1} = u^m + \tau f^{m+1}, \quad m = 0, 1, 2, \ldots, M$$

where A is SPD.

Note that the time step τ_m will not be constant, but it will change adaptively and it would be nice to avoid to compute new incomplete factorizations of $(I + \frac{\tau_m}{h^2} A)$ for each m.

The update of incomplete factorizations is not easy

Note that

$$\frac{d}{d\alpha}(A+\alpha E)^{-1}_{|\alpha=0} = -A^{-2}E,$$

showing that the inverse of $A+\alpha E$ can be very sensitive around $\alpha = 0$ when A^{-2} has large entries, as it is to be expected if A is ill-conditioned

Our incomplete updated factorizations

Let us write our incomplete factorization for A (ILDL^T) as

$$P = \tilde{L}\tilde{D}\tilde{L}^T \simeq A.$$

We update P based on ILDL^T for $A_{\alpha,E}$ as follows:

$$P_{\alpha,E} = \tilde{L}\,(\tilde{D} + \alpha B)\,\tilde{L}^T$$

where the correction matrix B is suitably chosen and depends on E and on L^{-1}.

Our incomplete updated factorizations

Let us write our incomplete factorization for A (ILDL^T) as

$$P = \tilde{L}\tilde{D}\tilde{L}^T \simeq A.$$

We update P based on ILDL^T for $A_{\alpha,E}$ as follows:

$$P_{\alpha,E} = \tilde{L}\,(\tilde{D} + \alpha B)\,\tilde{L}^T$$

where the correction matrix B is suitably chosen and depends on E and on L^{-1}.

Let us consider an approximate factorization for A^{-1} (AINV[1]):

$$P^{-1} = \tilde{Z}\,\tilde{D}^{-1}\,\tilde{Z}^T \left(= (\tilde{L}\tilde{D}\tilde{L}^T)^{-1}\right) \approx A^{-1}.$$

We update P^{-1} based on AINV for $A_{\alpha,E}^{-1}$ as follows:

$$Q_{\alpha,E} = P_{\alpha,E}^{-1} = \tilde{Z}\,(\tilde{D} + \alpha B)^{-1}\,\tilde{Z}^T;$$

---

[1] [Benzi, Meyer, Tuma, SISC96]

How to get correction matrix B?

Assume that

$$P^{-1} = ZD^{-1}Z^T = A^{-1} \quad \text{(the exact inverse)}$$

hence using for P our updated factorization gives

$$P_{\alpha,E}^{-1} = Z\,(D + \alpha B)^{-1}\,Z^T.$$

How to get correction matrix B?

Assume that

$$P^{-1} = ZD^{-1}Z^T = A^{-1} \quad \text{(the exact inverse)}$$

hence using for P our updated factorization gives

$$P_{\alpha,E}^{-1} = Z\,(D + \alpha B)^{-1}\,Z^T.$$

Consider now the difference

$$P_{\alpha,E} - A_{\alpha,E} = Z^{-T}\,(D + \alpha B)\,Z^{-1} - (A + \alpha E) = \alpha(LBL^T - E).$$

Taking

$$B = L^{-1}\,E\,L^{-T} = Z^T\,E\,Z$$

would result in the exact inverse

$$P_{\alpha,E}^{-1} = A_{\alpha,E}^{-1}$$

How to get correction matrix B?

Assume that

$$P^{-1} = ZD^{-1}Z^T = A^{-1} \quad \text{(the exact inverse)}$$

hence using for P our updated factorization gives

$$P_{\alpha,E}^{-1} = Z\,(D+\alpha B)^{-1}\,Z^T.$$

Consider now the difference

$$P_{\alpha,E} - A_{\alpha,E} = Z^{-T}\,(D+\alpha B)\,Z^{-1} - (A+\alpha E) = \alpha(LBL^T - E).$$

Taking

$$B = L^{-1}\,E\,L^{-T} = Z^T\,E\,Z$$

would result in the exact inverse

$$P_{\alpha,E}^{-1} = A_{\alpha,E}^{-1}$$

$\longrightarrow$Not a practical choice:

i) we don't know the exact Z in practice, but only $\tilde{Z}$;
ii) $B = L^{-1}\,E\,L^{-T} = Z^T\,E\,Z$ can be DENSE!

How to get correction matrix B?

Solve (possibly inexactly)

$$\min_{B \in \mathcal{S}} \|E - LBL^T\|_F$$

where $\mathcal{S}$ is a set of matrices B such that $D + \alpha B$ is "easy to invert".

The minimization problem would have to be dealt with only once, since there is no dependency on α.

Our proposal: order k updates

Define the order k updates (using ILDL^T) as

$$P_{\alpha,E} := \tilde{L}(\tilde{D} + \alpha B_k)\tilde{L}^T \tag{1}$$

where B_k is the band matrix given by

$$B_k = \tilde{Z}_k^T \, E \, \tilde{Z}_k, \tag{2}$$

$\tilde{Z}_k$ is obtained by extracting the $k-1$ upper diagonals from $\tilde{Z}$ if $k > 1$ or $B_1 = \mathrm{diag}(\tilde{Z}^T E \tilde{Z})$ if $k = 1$ (*note: we need $\tilde{Z}$ $(= \tilde{L}^{-T})$ if $k > 0$*)

Our proposal: order k updates

Define the order k updates (using ILDL^T) as

$$P_{\alpha,E} := \tilde{L}(\tilde{D} + \alpha B_k)\tilde{L}^T \tag{3}$$

where B_k is the band matrix given by

$$B_k = \tilde{Z}_k^T \, E \, \tilde{Z}_k, \tag{4}$$

$\tilde{Z}_k$ is obtained by extracting the $k-1$ upper diagonals from $\tilde{Z}$ if $k > 1$ or $B_1 = \mathrm{diag}(\tilde{Z}^T E \tilde{Z})$ if $k = 1$ (*note: we need* $\tilde{Z}$ *(*$= \tilde{L}^{-T}$*) if* $k > 0$)

Similarly, for order k inverse updates

$$Q_{\alpha,E} = P_{\alpha,E}^{-1} := \tilde{Z}(\tilde{D} + \alpha B_k)^{-1}\tilde{Z}^T \tag{5}$$

Hierarchy of order k updates

- $B = B_0 = I$ corresponds to *order 0 update*;
- $B = B_1$ corresponds to the *order 1 update*;
- the symmetric tridiagonal band matrix $B = B_2$ corresponds to the *order 2 update*;
- ...;
- To complete the hierarchy of approximations, we define the *order* -1 *update* by letting $B = B_{-1} = 0$ (which corresponds to just using $P^{-1} = A^{-1}$ as an approximation of $A_{\alpha,E}^{-1}$).

Are our updates $P_{\alpha,E}$ well defined?

Some sufficient conditions:

- A SPD, E diagonal and $Re(E)$ with positive entries. Then, $Re(B_k)$ is symmetric positive definite since $Re(E)$ SPD and $\tilde{Z}_k$ is a unit upper triangular matrix and therefore nonsingular. $\Rightarrow$the updates are guaranteed to be well defined.
- A and E SPD. Then, B_k is symmetric positive definite since $\tilde{Z}_k$ is a unit upper triangular matrix and therefore nonsingular. $\Rightarrow$the updates are guaranteed to be well defined.
- ...

Why this approach ?

This approach is motivated by the observation that, under suitable assumptions, the entries along the rows of A^{-1} decay away from the main diagonal [Demko, Moss, Smith: MathComp'84] $\Rightarrow$ banded approximations of Z tend to contain most of the large entries in Z (in L^{-1}).

Why this approach ?

This approach is motivated by the observation that, under suitable assumptions, the entries along the rows of A^{-1} decay away from the main diagonal [Demko, Moss, Smith: MathComp'84] $\Rightarrow$ banded approximations of Z tend to contain most of the large entries in Z (in L^{-1}).

Example: if A is banded SPD $\Rightarrow$

$$|(A^{-1})_{i,j}| \leq c\rho^{|i-j|}, \quad i, j = 1, ..., n, 0 < \rho < 1$$

and c is a constant

However, the decay can be imperceptible... / very strong if A is strongly diagonally dominant.

Why this approach ?

This approach is motivated by the observation that, under suitable assumptions, the entries along the rows of A^{-1} decay away from the main diagonal [Demko, Moss, Smith: MathComp'84] $\Rightarrow$ banded approximations of Z tend to contain most of the large entries in Z (in L^{-1}).

Example: if A is banded SPD $\Rightarrow$

$$|(A^{-1})_{i,j}| \leq c\rho^{|i-j|}, \quad i,j = 1,...,n, 0 < \rho < 1$$

and c is a constant

However, the decay can be imperceptible... / very strong if A is strongly diagonally dominant.

More on this and the analysis of the quality of updates/convergence of iterations can be found in

- [B.,ETNA vol.18, 2004].

Other approaches

- Preconditioners for SPD shifted linear systems by using an updated incomplete Cholesky factorization [Meurant, SISC, 2001] (not guaranteed if A is not an M-matrix and/or the shift is not positive real);
- non-preconditioned iterations reusing the approximation subspace because $\mathcal{K}_m(A, v) = \mathcal{K}_m(A+\alpha I, v)$ or "almost $=$"(not suitable when the r.h.s. are not collinear/initial approximation changes with α or $E \neq I$; not suitable when preconditioning is essential; not suitable if we don't want CG/GMRES-style Krylov solver)

There are no alternatives to our approach in the general case (e.g., $E \neq I$).

Numerical experiments

- Matlab implementation of the proposed techniques;
- Updates of order 0, 1 and 2 here;
- updates are compared with the "full" preconditioners (i.e., the incomplete factorizations are recomputed from scratch for each different α and E) and with the "order -1" "update", which is just the incomplete factorization computed for $\alpha = 0$.
- Example: Helmholtz equation with complex wave numbers.
- Another example: GLMs for time-dependent PDEs (a multiple-stages solver);

Helmholtz equation

An example of a problem whose discretization produces complex symmetric linear systems is the Helmholtz equation with complex wave numbers

$$-\nabla \cdot (c\nabla u) + \sigma_1(j)u + \mathbf{i}\sigma_2(j)u = f_j, \quad j = 0, ..., s, \tag{6}$$

where $\sigma_1(j)$, $\sigma_2(j)$ are real coefficient functions and c is the diffusion coefficient.

We choose domain $\mathcal{D} = [0,1] \times [0,1]$ with σ_1 constant, σ_2 a real coefficient function and $c(x,y) = e^{-x-y}$.

Helmholtz equation/2

As in [Freund, SISC92], we consider two cases.

→[Problem 1] Complex Helmholtz equation, u satisfies Dirichlet boundary conditions in $\mathcal{D}$. Discretizing the problem on a $n \times n$ grid, $N = n^2$, and mesh size $h = 1/(n+1)$ gives $s+1$ linear systems ($j = 0, ..., s$):

$$A_j x_j = b_j,\ A_j = H + h^2\sigma_1(j)I + \mathbf{i}h^2 D_j,\ D_j = \mathrm{diag}(d_1, ..., d_N),$$

where H is the discretization of $-\nabla \cdot (c\nabla u)$ by means of centered differences. The $d_r = d_r(j)$, $r = 1, ..., N$, $j = 0, ..., s$, are the values of $\sigma_2(j)$ at the grid points.

All test problems are based on a 31×31 mesh, the right hand sides are vectors with random components in $[-1, 1] + \mathbf{i}[-1, 1]$ and the initial guess is a random vector.

	Not prec		$ILDL_0^H$		$ILDL_1^H$		$ILDL_2^H$		full $ILDL^H$	
σ_1	it	Mf	it	Mf	it	Mf	it	Mf	it	Mf
50	38	13.9	22	7.1	22	7.1	18	**7.0**	19	9.0
100	36	12.7	20	**6.2**	20	**6.2**	17	6.5	17	7.7
200	32	10.2	18	**5.3**	18	**5.3**	15	**5.3**	15	6.6
400	26	7.2	16	**4.5**	16	**4.5**	13	**4.5**	12	5.1
800	20	4.6	15	4.1	15	4.1	12	4.1	9	**3.7**

Order k, $k = 0, 1, 2$ modified and incomplete LDL^H preconditioners. Complex Helmholtz equation, Problem 1. Results for the complex Helmholtz equation and Dirichlet boundary conditions as in Problem 1. The diagonal entries of D are chosen randomly in $[0, 1000]$.

	$AINV_0$		$AINV_1$		AINV2		full $AINV$	
σ_1	it	Mf	it	Mf	it	Mf	it	Mf
50	26	8.8	26	8.8	20	**7.8**	15	1793
100	25	8.2	25	8.3	19	**7.3**	14	1793
200	22	6.7	22	6.8	17	**6.3**	13	1793
400	19	**5.4**	19	5.5	15	**5.4**	11	1792
800	17	4.6	17	4.7	13	**4.5**	8	1791

Order k, $k = 0, 1, 2$ inverse modified and AINV (i.e., recomputed at each step) preconditioners. Results for the complex Helmholtz equation and Dirichlet boundary conditions as in Problem 1. The diagonal entries of D are chosen randomly in $[0, 1000]$.

Helmholtz equation/3

[Problem 2]$\rightarrow$ Real Helmholtz equation with complex boundary condition

$$\frac{\partial u}{\partial n} = \mathbf{i}\sigma_2(j)u,\ \{(1,y) \mid 0 < y < 1\}$$

discretized with forward differences and Dirichlet boundary conditions on the remaining three sides gives again

$$A_j x_j = b_j,\ A_j = H + h^2\sigma_1(j)I + \mathbf{i}h^2 D_j,\ D_j = \mathrm{diag}(d_1, ..., d_N),$$

The diagonal entries of D_j are given by $d_r = d_r(j) = 1000/h$ if r/m is an integer, 0 otherwise.

All test problems are based on a 31×31 mesh, the right hand sides are vectors with random components in $[-1,1] + \mathbf{i}[-1,1]$ and the initial guess is a random vector.

	Not prec		$ILDL_0^H$		$ILDL_1^H$		$ILDL_2^H$		full $ILDL^H$	
σ_1	it	Mf	it	Mf	it	Mf	it	Mf	it	Mf
.5	146	175	34	**14.0**	34	**14.0**	34	17.2	29	17.0
1	145	173	33	**13.0**	33	**13.0**	33	16.5	28	15.7
2	143	168	33	**13.0**	33	**13.0**	33	16.5	28	15.7
4	137	155	31	**12.1**	31	**12.1**	31	15.0	27	14.8
8	127	134	28	**10.3**	28	**10.3**	29	13.6	24	12.5

Order k, $k = 0, 1, 2$ modified and incomplete LDL^H (i.e., recomputed at each step) preconditioners. Results for the real Helmholtz equation and complex boundary conditions as in Problem 2.

	$AINV_0$		$AINV_1$		$AINV_2$		full $AINV$	
σ_1	it	Mf	it	Mf	it	Mf	it	Mf
.5	47	**23.4**	47	23.5	47	27.8	46	1812
1	46	**22.6**	46	**22.6**	46	26.9	45	1811
2	46	22.6	45	**21.8**	45	26.0	45	1811
4	44	**20.0**	44	20.1	44	25.1	42	1809
8	41	18.5	40	**17.9**	40	21.6	40	1807

Order k, $k = 0, 1, 2$ inverse modified and AINV (i.e., recomputed at each step) preconditioners. Results for the real Helmholtz equation and complex boundary conditions as in Problem 2.

Diffusion equation, variable coefficient, with GLMs

Let us consider the model problem

$$\begin{cases} \dfrac{\partial u}{\partial t} = \nabla \cdot (c\nabla u), & (x,y) \in \mathbf{R} = [0,\pi] \times [0,\pi], \\ u((x,y),t) = 0, & (x,y) \in \partial\mathbf{R}, \quad t \in [0,2\pi] \\ u((x,y),0) = x\,y, & (x,y) \in \mathbf{R}, \end{cases}$$

$c(x,y) = \exp(-x-y)$. Discretizing in space and using LMF in boundary value form to approximate u requires the solution of (block-quasi-Toeplitz) linear systems which need to be preconditioned. But now the preconditioner requires the solution of auxiliary shifted $m^2 \times m^2$ linear systems

$$(J + \alpha_j I)x_j = b_j,$$

where now the α_js are complex.

The linear system

$$MY = b, \quad Y = \left(y_0^T, y_1^T, ..., y_s^T\right)^T,$$

$$M = A \otimes I_m - hB \otimes J,$$

$$b = e_1 \otimes \eta_1 + e_{s+1} \otimes \eta_2 + h(B \otimes I_m)g, \;\; g = (g(t_0) \cdots g(t_s))^T$$

where $e_i \in \mathbb{R}^{s+1}$, $i = 1, ..., s+1$, is the i–th column of the identity matrix and A, $B \in \mathbb{R}^{(s+1)\times(s+1)}$ are small rank perturbations of nonsymmetric Toeplitz matrices.

The matrices A and B

$$A = \begin{pmatrix} 1 & \cdots & 0 & & & & \\ \alpha_0^{(1)} & \cdots & \alpha_k^{(1)} & & & & \\ \vdots & \vdots & \vdots & & & & \\ \alpha_0^{(\nu-1)} & \cdots & \alpha_k^{(\nu-1)} & & & & \\ \alpha_0 & \cdots & \alpha_k & & & & \\ & \alpha_0 & \cdots & & \alpha_k & & \\ & & \ddots & & \ddots & \ddots & \\ & & & & \alpha_0 & \cdots & \alpha_k \\ & & & \alpha_0^{(s-k+\nu+1)} & & \cdots & \alpha_k^{(s-k+\nu+1)} \\ & & & & \vdots & \vdots & \vdots \\ & & & & \alpha_0^{(s-1)} & \cdots & \alpha_k^{(s-1)} \\ & & & & 0 & \cdots & 1 \end{pmatrix}, \qquad (7)$$

$\alpha_j^{(r)}$, $j = 0, ..., k$, coefficients of additional formulas; B is defined similarly with the entries of the first and last rows set to zero.

The Kronecker Product Preconditioner

In [B.,SISC00] we propose the use of Krylov subspace methods with block-circulant preconditioners

$$P = \breve{A} \otimes I_m - h\breve{B} \otimes \tilde{J} \tag{8}$$

where $\breve{A}$, $\breve{B}$ are "circulant-like" approximations of matrices A, B, respectively, while $\tilde{J}$ is the approximation of J given by an incomplete factorization.

To apply P, we block-diagonalize and rewrite P as

$$P = (F^* \otimes I_m)G(F \otimes I_m),$$

F Fourier's matrix, $F_{j,r} = e^{2\pi \, \mathbf{i} \, j \, r/(s+1)}/\sqrt{s+1}$, $G = diag(G_0, ..., G_s)$.

Shifted matrices with complex shift

We need to solve $s+1$ auxiliary (shifted) linear systems when using these block-preconditioners[2] because of the block diagonal G in

$$P = (F^* \otimes I_m)G(F \otimes I_m),$$

F Fourier's matrix, $F_{j,r} = e^{2\pi \mathbf{i}\, j\, r/(s+1)}/\sqrt{s+1}$, $G = diag(G_0, ..., G_s)$

The $m \times m$ matrix G_j:

$$G_j = \phi_j I_m - h\psi_j \tilde{J} = h\psi_j \left((-\tilde{J}) + \frac{\phi_j}{h\psi_j} I_m \right), \ j = 0, ..., s, \qquad (9)$$

To apply P we need to (i) use FFTs; (ii) solve a block diagonal linear system whose blocks are parametric linear systems (in particular, complex shifted matrices with shift $\phi_j/(h\psi_j)$)

[2] see [B., SISC'00], [Chan,Jin,Ng, JNA'01], [B. SINUM 02],...

			GMRES		+full $AINV$		$+AINV_0$		$+AINV(J)$	
m	s	out	avg	Mf	avg	Mf	avg	Mf	avg	Mf
8	8	9	21.6	44	10.5	58	14.8	**18**	30.8	53
	16	8	17.3	58	10.3	103	15.7	**35**	35.9	122
	24	8	15.2	74	10.2	155	16.0	**55**	38.6	206
	32	8	13.9	84	10.2	204	16.2	**72**	40.2	291
16	8	9	51.8	789	15.7	2845	19.3	**105**	36.9	288
	16	9	40.2	1054	14.4	5663	19.2	**210**	45.0	815
	24	9	35.3	1139	14.1	7633	19.6	**292**	50.8	1334
	32	9	31.6	1276	13.7	10153	19.6	**385**	54.5	1997
24	8	9	84.3	4328	21.7	31282	24.1	**330**	40.8	754
	16	9	65.9	5783	19.4	62453	23.3	**635**	47.8	2001
	24	9	56.3	6698	19.1	93845	22.9	**942**	53.1	3629
	32	8	51.2	6753	18.7	124899	23.1	**1114**	57.5	4896
32	8	10	117	15874	*	*	28.8	**868**	43.9	1685
	16	9	92.7	19445	*	*	28.2	**1512**	51.2	3933
	24	9	79.3	22500	*	*	27.2	**2187**	55.7	6909
	32	9	70.6	24709	*	*	26.6	**2785**	59.0	10257

Convergence Theory

Just few words...

Theorem 1 [3] *Let us consider the sequence of algebraic linear systems. Let A be Hermitian positive definite, $\alpha \in \mathbb{C}$, $\delta > 0$ constant such that the singular values of the matrix $E - LB_kL^H$ are*

$$\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_t \geq \delta \geq \sigma_{t+1} \geq ... \geq \sigma_n \geq 0,$$

and $t \ll n$. Then, if

$$\max_{\alpha \in \{\alpha_0,\dots,\alpha_s\}} |\alpha| \cdot ||D^{-1} Z_k^H E Z_k||_2 \leq 1/2, \tag{10}$$

there exist matrices F, Δ and a constant c_α such that

$$\left(P_{\alpha,E}^{(k)}\right)^{-1} (A + \alpha\, E) = I + F + \Delta, \tag{11}$$

$$||F||_2 \leq \frac{2 \max_{\alpha \in \{\alpha_0,\dots,\alpha_s\}} |\alpha| c_\alpha \delta}{\lambda_{\min}(A)} \left(\frac{||Z||_2}{\min_i ||z_i||_2}\right)^2, \; Z = [z_1 \; \cdots \; z_n], \; z_i \in \mathbb{C}^n,$$

$\mathrm{rank}(\Delta) \leq t \ll n$*, the rank of Δ does not depend on α, c_α is a constant such that $\lim_{|\alpha| \to 0} c_\alpha = 1$, c_α of the order of unity.*

---

[3] [B.,ETNA04].

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;
- the updates are effective for important problems;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;
- the updates are effective for important problems;
- generating incomplete factorizations directly from the complex symmetric matrices $A_j = A + \alpha_j E_j$ requires complex arithmetic but not for our updates;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;
- the updates are effective for important problems;
- generating incomplete factorizations directly from the complex symmetric matrices $A_j = A + \alpha_j E_j$ requires complex arithmetic but not for our updates;
- the proposed algorithms can be used with other (incomplete) factorizations in different settings;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;
- the updates are effective for important problems;
- generating incomplete factorizations directly from the complex symmetric matrices $A_j = A + \alpha_j E_j$ requires complex arithmetic but not for our updates;
- the proposed algorithms can be used with other (incomplete) factorizations in different settings;
- no parameter estimates is required;

Conclusions.

- Updating incomplete factorizations for iterative methods is possible;
- the cost of the updates is negligible with respect to computing a new incomplete factorization;
- the updates are effective for important problems;
- generating incomplete factorizations directly from the complex symmetric matrices $A_j = A + \alpha_j E_j$ requires complex arithmetic but not for our updates;
- the proposed algorithms can be used with other (incomplete) factorizations in different settings;
- no parameter estimates is required;
- no alternative algorithms when $E_j \neq I$.