Solving Linear Systems of the Form $(A + \gamma UU^T)x = b$

Michele Benzi, Scuola Normale Superiore, Pisa

Due Giorni di Algebra Lineare Numerica
Napoli, February 14-15, 2022

Outline

Joint work with Chiara Faccio (SNS).

Outline

The problem

We are interested in finding efficient solvers for large systems of the form

$$(A + \gamma UU^T)x = b\,, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times k}$, $\gamma > 0$ and $b \in \mathbb{R}^n$. Here $1 \ll k \ll n$.

We target problems with the following characteristics:

- A is possibly singular, but $A + \gamma UU^T$ is nonsingular for $\gamma > 0$.
- A has one or more desirable property (sparsity, structure, etc.) which is lost if we form $A + \gamma UU^T$ explicitly.
- Mat-vecs with $A + \gamma UU^T$ can be computed efficiently.
- k may not be "small", but $k \times k$ systems can be solved accurately.
- Problem (1) must be solved repeatedly within a given application. Often, either A or U remains constant.

The problem (cont.)

Problems of the form (1) which such characteristics arise frequently in scientific computing.

Examples include:

- Augmented Lagrangian methods for saddle point problems;
- Solution of KKT systems in constrained optimization;
- Solution of sparse-dense least squares problems;
- Certain types of integro-differential equations;
- Solution of PDEs from slightly compressible elasticity;
- Numerical solution of PDEs with nonlocal BC's;
- ...

Example 1: Augmented Lagrangian methods

Consider the saddle point problem

$$\mathcal{A}\,\mathbf{x} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} = \mathbf{f}.$$

Such systems arise frequently from the finite element discretization of systems of PDEs, such as for example the Stokes equations, the Oseen problem (obtained from the steady Navier-Stokes equations via Picard linearization), or the coupled Stokes-Darcy system.

A powerful approach to solve such systems is the one based on the augmented Lagrangian.

This method is widely used for solving constrained optimization problems, too.

M. B. and M. Olshanskii, *An augmented Lagrangian-based approach to the Oseen problem*, SIAM J. Sci. Comput., 28 (2006), pp. 2095–2113.

Example 1: Augmented Lagrangian methods (cont.)

The idea is to replace the original saddle point problem with an equivalent one of the form:

$$\mathcal{A}_\gamma \, \mathbf{x} = \begin{bmatrix} A + \gamma B^T W^{-1} B & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} \hat{f} \\ g \end{bmatrix} = \hat{\mathbf{f}} \,,$$

where $\hat{f} = f + \gamma B^T W^{-1} g$. Here W is usually diagonal and positive definite. In the finite element setting, W is often the diagonal of the (pressure) mass matrix.

This new, augmented system is then solved by a Krylov subspace method like (F)GMRES with preconditioner

$$\mathcal{P}_\gamma = \begin{bmatrix} A + \gamma B^T W^{-1} B & B^T \\ 0 & -\gamma^{-1} W \end{bmatrix} .$$

In practice, the preconditioner is applied inexactly.

Example 1: Augmented Lagrangian methods (cont.)

The convergence of the preconditioned iteration is usually very fast and independent of parameters like the mesh size and viscosity, especially in the "large γ" limit.

However, at each iteration of the Krylov subspace method a linear system with coefficient matrix $A + \gamma B^T W^{-1} B$ must be solved (inexactly).

This linear system is of the form (1) with $U = B^T W^{-1/2}$. Here A is sparse, often block diagonal, and positive definite (or $A + A^T$ is).

Forming $A + \gamma B^T W^{-1} B$ explicitly would lead to loss of sparsity and structure. This system can be quite ill-conditioned (esp. for large γ) and its solution is the main challenge associated with the augmented Lagrangian approach.

It is therefore necessary to develop efficient iterative methods for it.

Ideally, we would like such solvers to be robust with respect to $\gamma > 0$.

Example 2: KKT systems in constrained optimization

The solution of (smooth) constrained minimization problems by interior point (IP) methods leads to sequences of linear systems of the form

$$\mathcal{A}\,\mathbf{x} = \begin{bmatrix} H & -C^T & 0 \\ C & 0 & -I \\ 0 & Z & \Lambda \end{bmatrix} \begin{bmatrix} \delta x \\ \delta\lambda \\ \delta z \end{bmatrix} = \begin{bmatrix} -r_1 \\ -r_2 \\ -r_3 \end{bmatrix} = \mathbf{f}.$$

Here $H = H^T$ is the Hessian of the objective function at the current point x_k, C is the Jacobian of the constraints at the same point, and Z and Λ are diagonal, positive definite matrices associated with the current values of the Lagrange multipliers λ_k and slack variables z_k, respectively.

The variable δz can easily be obtained using the last equation:

$$\delta z = -\Lambda^{-1}(r_3 + Z\delta\lambda)$$

and substituted into the second (block) equation.

Example 2: KKT systems in constrained optimization (cont.)

This yields the reduced system

$$\begin{bmatrix} H & -C^T \\ C & \Lambda^{-1}Z \end{bmatrix} \begin{bmatrix} \delta x \\ \delta\lambda \end{bmatrix} = \begin{bmatrix} -r_1 \\ -r_2 - \Lambda^{-1}r_3 \end{bmatrix}.$$

Eliminating $\delta\lambda$ leads to the fully reduced (Schur complement) system

$$(H + C^T Z^{-1}\Lambda C)\delta x = -r_1 - C^T Z^{-1}(r_3 + \Lambda r_2) =: b.$$

After solving for δx, the other unknowns $\delta\lambda$ and δz are readily obtained.

This system is of the form (1) with $A = H$, $U = C^T(Z^{-1}\Lambda)^{1/2}$ and $\gamma = 1$.

The Hessian is usually positive semidefinite, sparse and possibly structured. Again, forming $H + C^T Z^{-1}\Lambda C$ explicitly is generally undesirable. Instead, we propose to solve the fully reduced system with PCG using a suitable (algebraic) preconditioner.

Example 3: sparse-dense LS problems

Consider a large LS problem of the form

$$\|Bx - c\|_2 = \min,$$

where $B \in \mathbb{R}^{m\times n}$ and $c \in \mathbb{R}^m$. Assume that B has the following structure:

$$B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \quad B_1 \in \mathbb{R}^{(m-k)\times n}, \quad B_2 \in \mathbb{R}^{k\times n},$$

where B_1 is sparse and B_2 is dense. Then the LS problem is equivalent to the $n \times n$ system of normal equations:

$$(B_1^T B_1 + B_2^T B_2)x = B^T c\,,$$

which is of the form (1) with $A = B_1^T B_1$, $U = B_2^T$, $\gamma = 1$ and $b = B^T c$.

Once again, we would like to solve this system by an iterative method. The main challenge is again constructing an effective preconditioner.

Example 4: integro-differential equations

Consider an integro-differential equation of the form

$$\mathcal{L}[u] + \gamma \int_\Omega K(x,y)u(y)dy = f(x), \quad x \in \Omega \subset \mathbb{R}^d\,,$$

where $\mathcal{L}$ is a differential operator, $K(x,y) = K(y,x)$ is a symmetric kernel, and f is a given function. Boundary conditions are also prescribed.

Under suitable assumptions, the kernel of the integral operator can be well approximated by a degenerate kernel:

$$K(x,y) \approx \sum_{i=1}^{k} \phi_i(x)\phi_i(y)\,,$$

where $\{\phi_i(x)\}$ is a set of linear independent functions on Ω. The number of terms in this (approximate) expansion depends on how quickly the eigenvalues of the (compact) integral operator decay to 0, which in turn depends on the regularity of the kernel K.

Example 4: integro-differential equations (cont.)

Upon discretization, the problem takes the form

$$(A + \gamma U U^T)x = b$$

where $A \in \mathbb{R}^{n \times n}$ is a sparse matrix while $U \in \mathbb{R}^{n \times k}$ is generally a dense matrix. Each column of U corresponds to one of the ϕ_i.

Again, such problem calls for an iterative solution method. Mat-vecs can be performed efficiently without forming $A + \gamma U U^T$ explicitly (which would destroy the sparsity in A), and the main challenge is to find an effective preconditioner.

We are particularly interested in the case where the number of terms k in the expansion of the kernel, which is the rank of U, is not very small. In other words, the assumed regularity of the kernel is not necessarily high.

Note that here k is independent of n.

Outline

The preconditioner

Consider again the linear system (1): $(A + \gamma UU^T)x = b$.

As we have seen, in applications the matrix A (or $A + A^T$) is usually at least positive semidefinite, and we will make this assumption.

Also, although this is not strictly necessary, we will assume that

$$\text{Ker}\,(A) \cap \text{Ker}\,(U^T) = \{0\}\,,$$

so that $A + \gamma UU^T$ is nonsingular (and positive definite) for all $\gamma > 0$.

When A is nonsingular, we could use the Sherman-Morrison-Woodbury (SMW) formula to solve (1), but this is only applicable to problems of moderate size. Recall that SMW states that

$$(A + \gamma UU^T)^{-1} = A^{-1} - \gamma A^{-1}U(I_k + \gamma U^T A^{-1}U)^{-1}U^T A^{-1}.$$

Another possibility would be to build preconditioners based on the SMW formula, where the action of A^{-1} is replaced by some approximation, but our attempts were unsuccessful. Also, in some problems A is singular.

The preconditioner (cont.)

When k is small (say, $k = 10$ or less) then any good preconditioner for A (or $A + \alpha I_n$, $\alpha > 0$, if A is singular) tends to give good results. In fact, using CG preconditioned with A^{-1} yields convergence in at most $k + 1$ steps. However, if k is in the hundreds (or larger), this approach is not appealing.

Hence, we need to take into account both A and γUU^T when building the preconditioner. We do this by forming a suitable product preconditioner, as follows.

Let $\alpha > 0$ and consider the two splittings

$$A + \gamma UU^T = (A + \alpha I_n) - (\alpha I_n - \gamma UU^T)$$

and

$$A + \gamma UU^T = (\alpha I_n + \gamma UU^T) - (\alpha I_n - A).$$

Note that both $A + \alpha I_n$ and $\alpha I_n + \gamma UU^T$ are invertible.

The preconditioner (cont.)

Let $x^{(0)} \in \mathbb{R}^n$ and consider the alternating iteration

$$(A + \alpha I_n)x^{(k+1/2)} = (\alpha I_n - \gamma U U^T)x^{(k)} + b\,,$$

$$(\alpha I_n + \gamma U U^T)x^{(k+1)} = (\alpha I_n - A)x^{(k+1/2)} + b\,,$$

with $k = 0, 1, \ldots$ This alternating scheme is analogous to that of other well-known iterative methods like ADI, HSS, MHSS, RDF, RPF, etc.

Theorem 1: Assume $A + A^T$ is positive definite. Then the sequence $\{x^{(k)}\}$ converges, as $k \to \infty$, to the unique solution of equation (1), for any choice of $x^{(0)}$ and for all $\alpha > 0$.

To turn this into a practical method, we will use it as a preconditioner for a Krylov-type method rather than as a stationary iterative scheme. This will also allow inexact solves.

The preconditioner (cont.)

To derive the preconditioner we eliminate $x^{(k+1/2)}$ and write the iterative scheme as the fixed-point iteration

$$x^{(k+1)} = T_\alpha x^{(k)} + c = (I_n - P_\alpha^{-1} A_\gamma) x^{(k)} + P_\alpha^{-1} b\,,$$

where we have set $A_\gamma = A + \gamma U U^T$. An easy calculation reveals that the preconditioner P_α is given, in factored form, by

$$P_\alpha = \frac{1}{2\alpha}(A + \alpha I_n)(\alpha I_n + \gamma U U^T).$$

The scalar factor $\frac{1}{2\alpha}$ is immaterial for preconditioning, and can be ignored.

Applying this preconditioner requires two solves involving $A + \alpha I_n$ and $\alpha I_n + \gamma U U^T$ at each Krylov iteration.

Generally speaking, each of these should be considerably simpler than solving systems involving the matrix $A_\gamma = A + \gamma U U^T$.

The preconditioner (cont.)

Consider first solves involving $A + \alpha I_n$. If A is sparse, and/or structured (e.g., block diagonal, Toeplitz, etc.) then so is $A + \alpha I_n$.

Exact solves with $A + \alpha I_n$ can be replaced, if necessary, with inexact solves using either a good preconditioner for $A + \alpha I_n$ or a few steps of an inner iteration (PCG, AMG, or other).

Note the usual trade-off: larger values of α make solves with $A + \alpha I_n$ easier, but may degrade the performance of the preconditioner P_α.

Numerical experiments suggest that the solution of linear systems involving $\alpha I_n + \gamma U U^T$ is more critical. Note that this matrix is SPD for all $\alpha > 0$, but ill-conditioned for small α (or very large γ).

The preconditioner (cont.)

The Sherman-Morrison-Woodbury formula yields

$$(\alpha I_n + \gamma U U^T)^{-1} = \alpha^{-1} I_n - \alpha^{-1} \gamma U (\alpha I_k + \gamma U^T U)^{-1} U^T.$$

The main cost is the solution at each step of a $k \times k$ linear system with matrix $\alpha I_k + \gamma U^T U$, which can be performed by Cholesky factorization (computed once and for all at the outset) or possibly by a suitable inner PCG iteration or maybe an (algebraic) MG method.

Note that for incompressible flow problems, $\alpha I_k + \gamma U^T U$ is essentially a (shifted) discrete pressure Laplacian.

In the numerical solution of the Navier–Stokes equations using (say) Picard iteration, this matrix remains constant, whereas the matrix A changes.

Hence, the cost of a Cholesky factorization can be amortized over many nonlinear (or time) steps.

Outline

Possible variants

Building on the main idea, different variants of the preconditioner can be envisioned.

If A happens to be nonsingular and linear systems with A are not too difficult to solve (e.g., well-conditioned), then it may not be necessary to shift A, leading to a preconditioner of the form

$$P_{\alpha,0} = A(\alpha I_n + \gamma U U^T).$$

Note that Theorem 1, however, is no longer applicable.

When A is symmetric positive semidefinite, we'd like the preconditioner to be SPD so that it can be used with the CG method. In this case we can consider a symmetrized version of the preconditioner, for example

$$P_{\alpha,s} = L(\alpha I + \gamma U U^T)L^T$$

where L is the Cholesky (or incomplete Cholesky) factor of $A + \alpha I$ (or of A itself if A is SPD and not very ill-conditioned).

Possible variants (cont.)

In some cases (but not always) the performance of the method improves if A_γ is diagonally scaled so that it has unit diagonal prior to forming the preconditioner.

Note that the matrix

$$D_\gamma = \mathsf{diag}(A + \gamma U U^T)$$

can be easily computed:

$$(D_\gamma)_{ii} = a_{ii} + \gamma \|u_i^T\|_2^2 \,,$$

where u_i^T is the ith row of U.

It is easy to see that applying the preconditioner to the diagonally scaled matrix $D_\gamma^{-1/2} A_\gamma D_\gamma^{-1/2}$ is equivalent to using the modified preconditioner

$$(A + \alpha D_\gamma)(\alpha D_\gamma + \gamma U U^T)$$

on the original matrix.

Outline

Bounds on the eigenvalues

Let $A_\gamma := A + \gamma UU^T$ and $P_\alpha := \frac{1}{2\alpha}(A + \alpha I)(\alpha I + \gamma UU^T)$.

WLOG we can assume that $\|A\|_2 = 1$ and $\|U\|_2 = 1$. We also assume that A_γ is nonsingular (that is, $\mathrm{Ker}\,(A) \cap \mathrm{Ker}\,(U^T) = \{0\}$).

Theorem 2. Let $A + A^T$ be positive semidefinite. If (λ, x) is an eigenpair of the preconditioned matrix $P_\alpha^{-1} A_\gamma$, with $\|x\|_2 = 1$, then

$$\mu < Re(\lambda) < 2, \quad |Im(\lambda)| < 1 \tag{2}$$

where

$$\mu = \frac{\alpha\, \lambda_{\min}(A + A^T)}{(1+\alpha)(\alpha+\gamma)}.$$

If (λ, x) is an eigenpair with $x \in \mathrm{Ker}\,(U^T)$, then

$$\lambda = \frac{2x^* Ax}{x^* Ax + \alpha}$$

(independent of γ).

Bounds on the eigenvalues (cont.)

Some comments on this result are in order:

- We see from (2) that the lower bound is uninformative if $A + A^T$ is singular ($\mu = 0$).
- The lower bound (if $\neq 0$) is maximized for $\alpha = \sqrt{\gamma}$.
- Choosing $\alpha = \sqrt{\gamma}$ to maximize the lower bound may not be optimal.
- The lower bound approaches 0 if $\gamma \to \infty$, indicating that the case of large γ *may* be challenging.
- The result assumes the preconditioner is applied exactly (often not true in practice).
- Eigenvalues alone may not be descriptive of GMRES convergence.

Bounds on the eigenvalues (cont.)

γ	α	$\max(Re(\lambda))$	$\min(Re(\lambda))$	lower bound
0.1	0.1	1.818e+00	1.700e-02	5.709e-04
	0.3162	1.519e+00	5.409e-03	7.250e-04
	5.0	3.333e-01	3.430e-04	2.052e-04
1.0	0.5	1.333e+00	6.590e-03	2.791e-04
	1.0	1.000e+00	3.300e-03	3.140e-04
	5.0	4.683e-01	6.609e-04	1.744e-04
50.0	1.0	1.532e+00	3.323e-03	1.231e-05
	7.0711	1.658e+00	4.707e-04	1.928e-05
	10.0	1.606e+00	3.328e-04	1.903e-05

Table: Stokes problem with 64×64 mesh and Q2-Q1 discretization. A and U normalized so that $\|A\|_2 = 1 = \|U\|_2$. In boldface the value $\alpha = \sqrt{\gamma}$.

Bounds on the eigenvalues (cont.)

γ	α	$\max(Re(\lambda))$	$\min(Re(\lambda))$	lower bound
0.1	0.1	1.818e+00	5.894e-03	4.380e-05
	0.3162	1.519e+00	1.868e-03	5.562e-05
	5.0	3.333e-01	1.182e-04	1.575e-05
1.0	0.5	1.333e+00	8.587e-04	2.141e-05
	1.0	1.000e+00	4.295e-04	2.409e-05
	5.0	3.430e-01	8.591e-05	1.338e-05
50.0	1.0	1.898e+00	3.788e-04	9.447e-07
	7.0711	1.744e+00	5.358e-05	1.479e-06
	10.0	1.662e+00	3.789e-05	1.460e-06

Table: Oseen problem with 64×64 mesh, $\nu = 0.01$, and Q2-Q1 discretizations. A and U normalized so that $\|A\|_2 = 1 = \|U\|_2$. In boldface the value $\alpha = \sqrt{\gamma}$.

To compute the lower bound we use $\lambda_{\min}(\frac{A+A^T}{2})$ since $A \neq A^T$.

Bounds on the eigenvalues (cont.)

α	$\max(Re(\lambda))$	$\min(Re(\lambda))$	lower bound
0.001	1.998e+00	4.247e-03	1.597e-03
0.01	1.980e+00	4.167e-02	1.568e-02
0.1	1.818e+00	3.477e-01	1.322e-01
0.5	1.333e+00	9.087e-01	3.556e-01
1.0	1.000e+00	8.889e-01	4.000e-01
5.0	5.383e-01	2.759e-01	2.222e-01
10.0	3.180e-01	1.481e-01	1.322e-01
20.0	1.738e-01	7.692e-02	7.256e-02
50.0	7.350e-02	3.150e-02	3.076e-02

Table: Problem `lp_fit2p` from SuiteSparse Matrix Collection (sparse-dense least-squares problem), $\gamma = 1$. A and U normalized so that $\|A\|_2 = 1 = \|U\|_2$.

Bounds on the eigenvalues (cont.)

α	$\max(Re(\lambda))$	$\min(Re(\lambda))$	lower bound
0.001	1.998e+00	6.508e-03	7.343e-04
0.01	1.980e+00	6.321e-02	7.213e-03
0.1	1.818e+00	4.834e-01	6.081e-02
0.5	1.333e+00	8.484e-01	1.635e-01
1.0	1.000e+00	5.384e-01	1.839e-01
5.0	4.335e-01	1.372e-01	1.022e-01
10.0	2.457e-01	7.106e-02	6.081e-02
20.0	1.313e-01	3.617e-02	3.337e-02
50.0	5.470e-02	1.463e-02	1.414e-02

Table: Problem `mosarqp1` from Maros and Mészáros collection (KKT systems in constrained optimization), $\gamma = 1$. A and U normalized so that $\|A\|_2 = 1 = \|U\|_2$.

Outline

Numerical experiments with matrices from Stokes and Oseen problems (leaky-lid driven cavity)

We tested inexact variants of the proposed preconditioner

$$P_\alpha = \frac{1}{2\alpha}(A + \alpha I)(\alpha I + \gamma U U^T)$$

on a number of linear systems of the form

$$(A + \gamma B^T W^{-1} B)x = b$$

associated with 2D steady Stokes and Oseen problems, varying γ, α, the mesh size h, the viscosity ν and the type of discretization used. Note that $U = B^T W^{-1/2}$ and that W is diagonal. Also, A is block diagonal.

For efficiency, we replace the factor $(A + \alpha I)$ with its no-fill Cholesky or ILU factorization, denoted by M_α. The factor $(\alpha I + \gamma B^T W^{-1} B)$ is inverted exactly via the SMW formula and the Cholesky factorization of the $k \times k$ matrix $\alpha I_k + \gamma W^{-1/2} B B^T W^{-1/2}$.

We also describe a simple but effective heuristic for the choice of α.

Numerical experiments with steady 2D Oseen problem

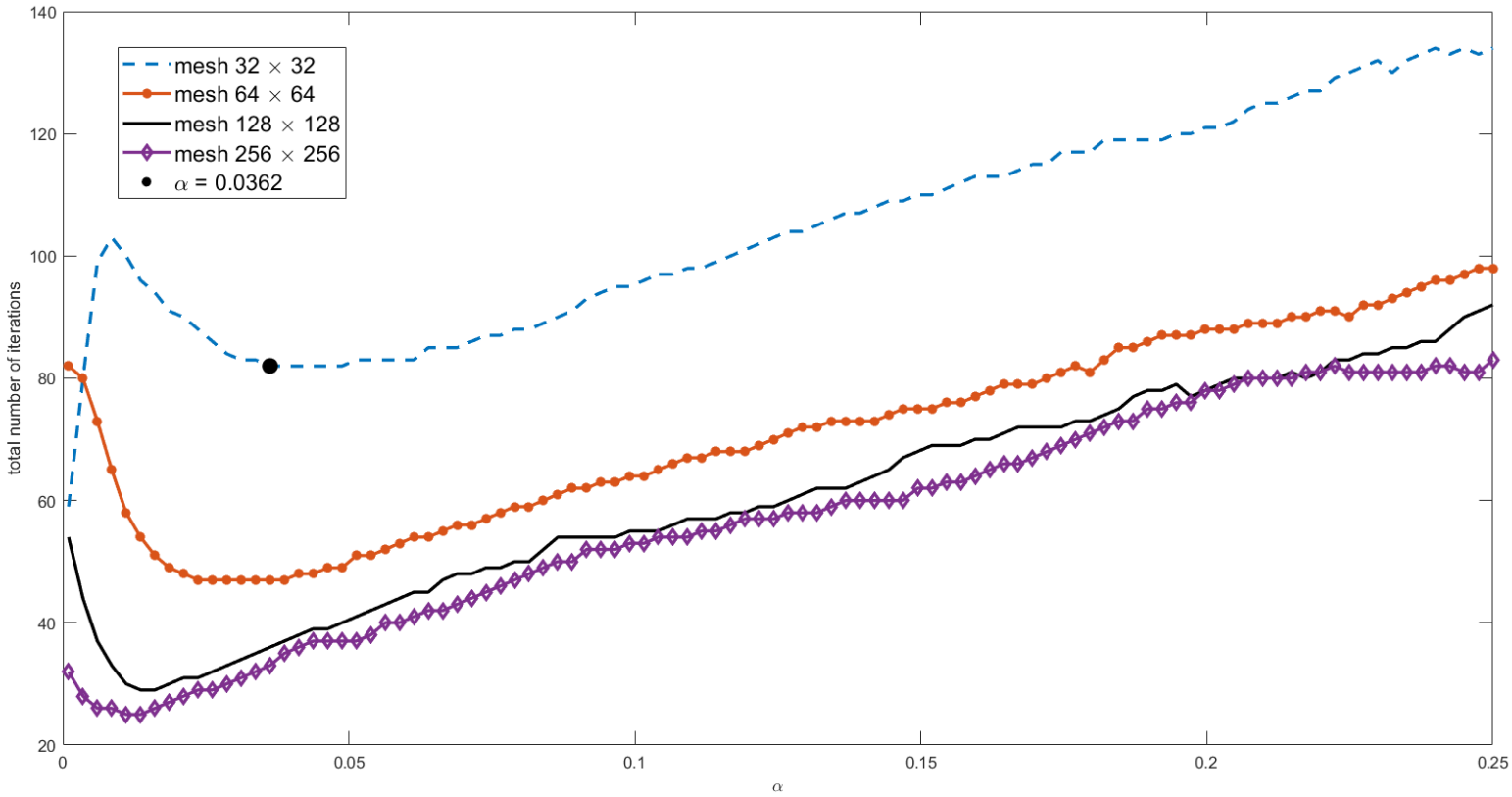

Figure: Number of PGMRES iterations versus α for the 2D Oseen problem with $\nu = 0.01$, $\gamma = 100$, Q2-Q1 finite element discretization and different mesh sizes. GMRES restart $m = 20$, convergence residual tolerance = 1e-06. Diagonal scaling is applied.

A simple heuristic for the choice of α

mesh	α^*	iterations with α^*	optimal α	iterations with optimal α
64×64	0.0256	47	0.0236	47
128×128	0.0181	30	0.0136	29
256×256	0.0128	25	0.0111	25

Table: PGMRES iteration counts for 2D Oseen problem with $\nu = 0.01$, $\gamma = 100$, Q2-Q1 finite element discretization. For 32×32 mesh we find $\alpha^* = 0.0362$, then for the $2^{5+k} \times 2^{5+k}$ mesh we set $\alpha^* = \frac{0.0362}{2^{k/2}}$. That is, we divide α^* by $\sqrt{2}$ each time h is halved.

Numerical experiments for steady 2D Oseen problem

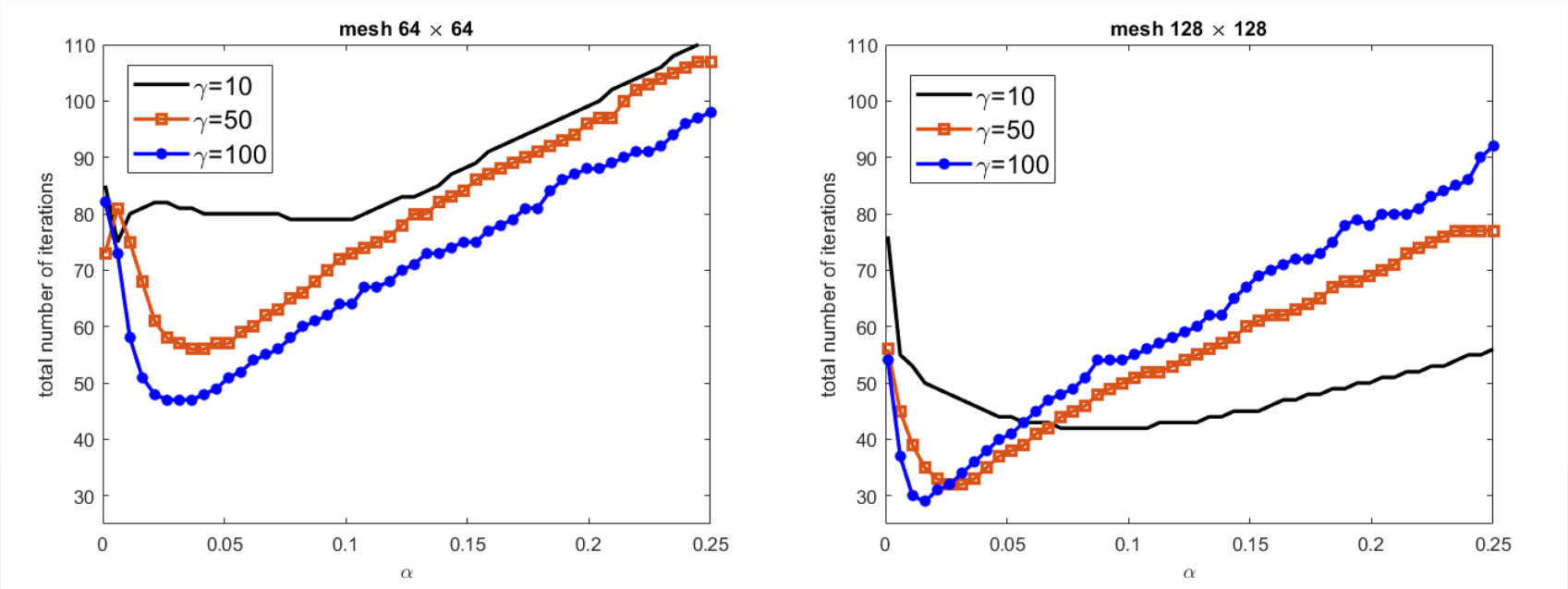

Figure: Number of iterations versus α for the 2D Oseen problem with $\nu = 0.01$, Q2-Q1 finite element discretization on 64×64 mesh (LEFT) and on 128×128 mesh (RIGHT) for different values of γ.

Numerical experiments for steady 2D Stokes problem

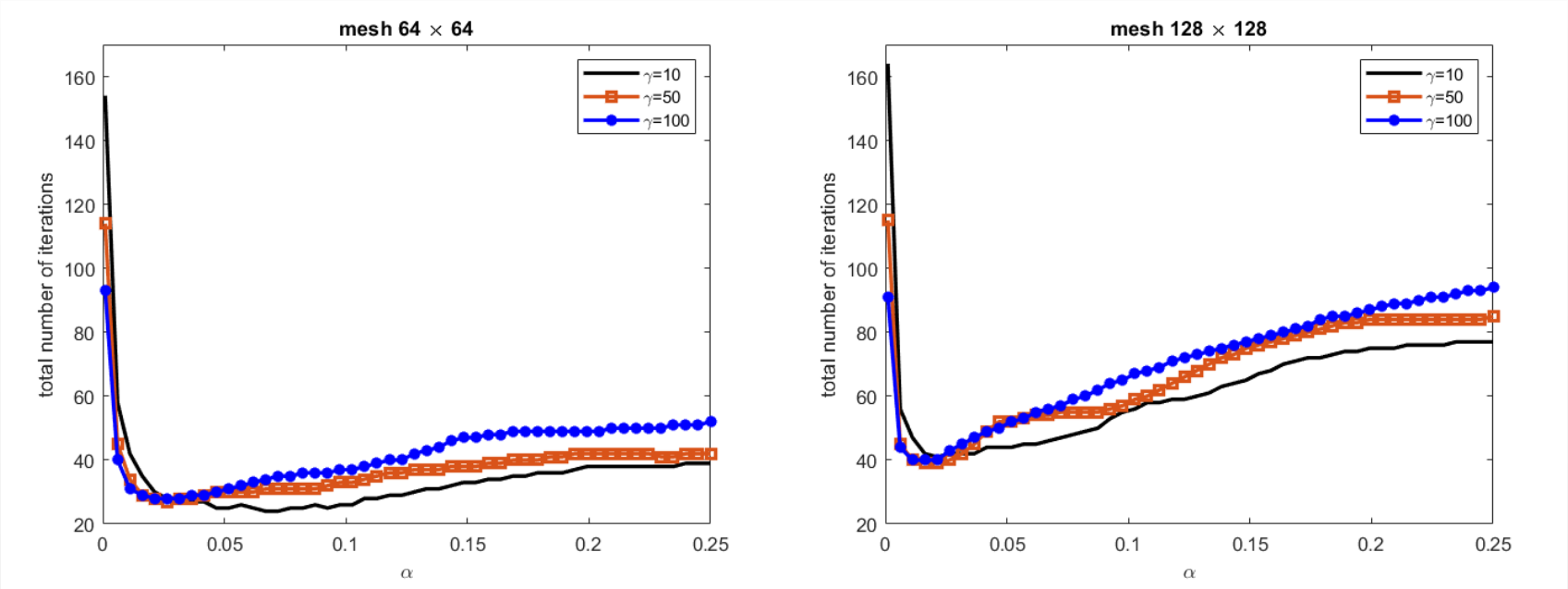

Figure: Number of iterations versus α for the 2D Stokes problem with Q2-Q1 finite element discretization on 64×64 mesh (LEFT) and on 128×128 mesh (RIGHT) for different values of γ.

Numerical experiments on sparse-dense LS problems, I

Problem `lp_fit2p` (SuiteSparse Collection). Here B_1 is 13500×3000, B_2 is 25×3000 (hence $n = 3000$, $k = 25$), $\kappa(B_1^T B_1 + B_2^T B_2) = 2.52 \times 10^9$. **Note:** the norm of $B_2^T B_2$ is 7 orders of magnitude larger than that of $B_1^T B_1$.

α	M_α	P_α	no prec.
0.001	109	10	174
0.01	109	10	
0.1	109	10	
0.5	110	10	
1.0	110	8	
10.0	139	7	
20.0	149	9	

Table: Total number of (P)GMRES iterations for `lp_fit2p`. M_α is replaced by its no-fill incomplete Cholesky approximation. No diagonal scaling is used.

We remark that the cost for P_α is only slightly larger than for M_α.

Numerical experiments on sparse-dense LS problems, II

Problem `scfxm1-2r` (SuiteSparse Collection): B_1 is 65886×37980, B_2 is 57×37980 (so $n = 37980$, $k = 27$), $\kappa(B_1^T B_1 + B_2^T B_2) = 9.32 \times 10^6$.
Note: $A = B_1^T B_1$ is singular.

α	M_α	P_α	no prec.
0.001	1572	555	240
0.01	693	91	
0.1	183	36	
0.5	154	39	
1.0	155	50	
10.0	213	141	

α	P_α^S	no prec. CG
0.001	1331	184
0.01	415	
0.1	105	
0.5	58	
1.0	65	
10.0	109	

Table: Total number of iterations for `scfxm1-2r` problem. Diagonal scaling is applied. (LEFT) PGMRES. M_α is replaced by its no-fill incomplete Cholesky approximation. (RIGHT) PCG. We consider the symmetrized version of the preconditioner: $P_\alpha^S = L(\alpha I_n + \gamma U U^T)L^T$, where L is the no-fill incomplete Cholesky factor of $M_\alpha = (B_1^T B_1 + \alpha I_n)$.

Numerical experiments on reduced KKT systems, I

Problem `primal4` (Maros ans Mészáros collection), $n = 1489$, $k = 75$. The reduced system matrix $H + C^T(Z^{-1}\Lambda)C$ has condition number 3.41×10^5. **Note:** H is singular.

α	M_α	P_α	no prec.
0.001	2000*	13	2000*
0.01	1458	13	
0.1	1379	11	
0.5	1358	9	
1.0	2000*	2	
10.0	2000*	15	
20.0	2000*	18	

Table: Total number of iterations for `primal4`. No diagonal scaling is applied.

Numerical experiments on reduced KKT systems, II

Problem `mosarqp1` problem (Maros ans Mészáros collection), $n = 2500$, $k = 700$. The reduced system matrix $H + C^T(Z^{-1}\Lambda)C$ has condition number 3.35×10^4.

α	M_α	P_α	no prec.
0.001	2000*	309	2000*
0.01	2000*	66	
0.1	2000*	20	
0.5	2000*	9	
1.0	2000*	6	
10.0	2000*	11	
20.0	2000*	13	

α	P_α^S	no prec. CG
0.001	293	246
0.01	125	
0.1	45	
0.5	20	
1.0	14	
10.0	15	
20.0	17	

Table: Total number of iterations for `mosarqp1`. No diagonal scaling is applied. (LEFT) PGMRES. (RIGHT) PCG. We consider a symmetrized version of the preconditioner: $P_\alpha^S = L(\alpha I_n + \gamma UU^T)L^T$, where L is the no-fill incomplete Cholesky factor of $H + \alpha I_n$.

Numerical experiments on integro-differential problem

γ	α	M_α	P_α	no prec.
0.1	0.01	2000*	176	1711
	0.1	2000*	73	
	1.0	2000*	106	
1.0	0.01	2000*	175	782
	0.1	2000*	68	
	1.0	1709	59	
5.0	0.01	1049	175	173
	0.1	847	56	
	1.0	574	39	

Table: A is the finite difference 2D Laplacian with $n = 10,404$, $U = [u_1, ..., u_k]$, where $u_k = [\sin(kx_1), \ldots, \sin(kx_n)]^T$, $[x_1, \ldots, x_n]$ is a uniform partition of $[0, 2\pi]$, $k = 100$. A is reordered with RCM. We approximate $A + \alpha I$ with `ichol` with zero-fill. No diagonal scaling is applied.

Outline

Conclusions

- Introduced a new solver for a wide class of tough linear systems
- The proposed preconditioner seems to work well in practice
- Solves with $A + \alpha I_n$ can be (very) inexact...
- ... but exact solves with $\alpha I + \gamma U U^T$ seem to be necessary
- SMW formula $\Rightarrow$ only a $k \times k$ solve needed
- Often, much of this work can be reused
- For some PDE problems we found a simple heuristic for choosing α
- Some theory available for ideal case

Future work

- Further investigate the spectrum of $P_\alpha^{-1} A_\gamma$. Clustering?
- Try to find eigenvalue bounds for singular A
- More work to be done on the choice of α
- Investigate other approaches to solving systems with $\alpha I + \gamma U U^T$ (iterative?)
- Are there better ways to symmetrize the preconditioner for use with CG?
- Write the paper!