Preconditioner Updates for Solving Sequences of Linear Systems arising in inexact methods for optimization.

Stefania Bellavia
Università degli Studi di Firenze

Based on works with Valentina De Simone, Daniela di Serafino, Benedetta Morini, Margherita Porcelli

Numerical Methods for Large-Scale Nonlinear Problems and Their Applications, ICERM Providence, RI, USA, Aug. 31- Sept. 4, 2015
Outline

Consider the problem of preconditioning a sequence of linear systems

$$A_k x = b_k, \quad k = 1, \ldots$$

where $A_k \in \mathbb{R}^{n \times n}$ are nonsingular indefinite sparse matrices.

- Computing preconditioners $P_1, P_2, \ldots$, for individual systems separately can be very expensive.
- Reduction of the cost can be achieved by sharing some of the computational effort among subsequent linear systems.
Updating strategies

- Given a preconditioner $P_{\text{seed}}$ for some seed matrix $A_{\text{seed}}$ of the sequence, updated preconditioners for subsequent matrices $A_k$ are computed at a low computational cost.
- **Minimum requirement:** Updates must be able to precondition sequences of **slowly varying systems**. A **periodical or dynamic** refresh of the seed preconditioner may be necessary.
- **Expected behaviour** in terms of linear solver iterations: to be in between the frozen and the recomputed preconditioner.
Updating strategies

- Given a preconditioner $\mathcal{P}_{\text{seed}}$ for some seed matrix $A_{\text{seed}}$ of the sequence, updated preconditioners for subsequent matrices $A_k$ are computed at a low computational cost.

- **Minimum requirement:** Updates must be able to precondition sequences of *slowly varying systems*. A *periodical or dynamic* refresh of the seed preconditioner may be necessary.

- **Expected behaviour** in terms of linear solver iterations: to be in between the frozen and the recomputed preconditioner.

Updating procedures for two classes of systems:

- **nonsymmetric linear systems** arising in Newton-Krylov methods (nearly-matrix free preconditioning strategies);

- **KKT systems** arising in Interior Point methods.
Sequences of systems in Newton-Krylov methods

\[ F(x) = 0 \]

\( F : \mathbb{R}^n \to \mathbb{R}^n \) continuously differentiable, \( J \) Jacobian matrix of \( F \).

Sequence of Newton equations

\[ J(x_k)s = -F(x_k), \quad k = 0, 1, \ldots \]

- By continuity, \( \{J(x_k)\} \) varies slowly if the iterates are close enough.
- \( A_k = J(x_k) \),
- \( A_kv \) provided by an operator or approximated by finite-differences, i.e.

\[
A_kv \simeq \frac{F(x_k + \epsilon v) - F(x_k)}{\epsilon \|v\|} \quad \epsilon > 0.
\]
Preconditioning & Matrix-free setting

- Unpreconditioned Newton-Krylov methods are matrix-free.
  But a truly matrix-free setting is lost when an algebraic preconditioner is used.

- A preconditioning strategy is classified as *nearly matrix-free* if it lies close to a true matrix-free settings. Specifically, if
  - only a few full matrices are formed;
  - for preconditioning most of the systems of the sequence, matrices that are reduced in complexity with respect to the full $A'_k s$ are required.
  - matrix-vector product approximations by finite differences can be used.

  [Knoll, Keyes 2004]
Let $G$ be the function that, evaluated at $v \in \mathbb{R}^n$, provides the product of $A_k$ times $v$.

- $G$ separable: computing one component of $G$ costs about an $n$-th part of the full function evaluation.
- $G$ separable: The cost of evaluating a selected entry of $A_k$ corresponds approximately to the $n$-th part of the cost of performing one matrix-vector product.
- Newton-Krylov: $G$ can be the finite-differences operator, $G$ is separable whenever the nonlinear function itself is separable.
- Nearly matrix-free strategy whenever $G$ is separable and only selected entries of the current matrix $A_k$ are required.
Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:

Preconditioner updates in Newton-Krylov methods

Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:


Recycled Krylov information preconditioners:

- symmetric and nonsymmetric matrices: [Carpentieri, Duff, Giraud 2003], [Knoll, Keyes, 2004], [Parks, de Sturler, Mackey, Jhonson, Maiti, 2006], [Loghin, Ruiz, Tohuami 2006], [Giraud, Gratton, Martin, 2007], [Fasano, Roma 2013], [Soodhalter, Szyld, Xue, 2014].

Incremental ILU preconditioners:

- nonsymmetric matrices: [Calgaro, Chehab, Saad 2010].
Preconditioner updates in Newton-Krylov methods

Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:


Recycled Krylov information preconditioners:

- symmetric and nonsymmetric matrices: [Carpentieri, Duff, Giraud 2003], [Knoll, Keyes, 2004], [Parks, de Sturler, Mackey, Jhonson, Maiti, 2006], [Loghin, Ruiz, Tohuami 2006], [Giraud, Gratton, Martin, 2007], [Fasano, Roma 2013], [Soodhalter, Szyld, Xue, 2014].

Incremental ILU preconditioners:

- nonsymmetric matrices: [Calgaro, Chehab, Saad 2010].

Updates of factorized preconditioners:

Approximate updates of factorized preconditioners

Consider two linear systems

\[ \mathbf{A}_{seed} \mathbf{x} = b, \quad \mathbf{A}_k \mathbf{x} = b_k \]

and let \( \mathcal{P}_{seed} = \mathbf{LDU} \approx \mathbf{A}_{seed} \).

- It follows

\[ \mathbf{A}_k = \mathbf{A}_{seed} + (\mathbf{A}_k - \mathbf{A}_{seed}) \approx \mathbf{L} (\mathbf{D} + \mathbf{L}^{-1}(\mathbf{A}_k - \mathbf{A}_{seed})\mathbf{U}^{-1})\mathbf{U} \]

- The \textit{ideal} update of the middle-term is costly:
  - the difference matrix \( \mathbf{A}_k - \mathbf{A}_{seed} \) should be formed;
  - in general the ideal update is dense and its factorization is impractical.

- Form an \textit{approximate} and cheap update.
Update of LDU factorizations [Duintjer Tebbens, Tuma 2007, 2010]

Ideal updated preconditioner for $A_k$:

$$A_k \approx L(D + L^{-1}(A_k - A_{seed})U^{-1})U$$

The approximate updated preconditioner is obtained as follows:

1. Neglect either $L^{-1}$ or $U^{-1}$ (closeness of $L$ or $U$ to the identity matrix):

$$A_k \approx L(D + (A_k - A_{seed})U^{-1})U$$

$$A_k \approx L(D + L^{-1}(A_k - A_{seed})U^{-1})U$$

2. Use only a triangular part of the current matrix $A_k$:

$$\mathcal{P}_k = L(DU + \text{triu}(A_k - A_{seed}))$$

$$\mathcal{P}_k = (LD + \text{tril}(A_k - A_{seed}))U$$

$\mathcal{P}_k$ is factorized. This approach is not suitable for symmetric matrices.
Banded approximate factors

Ideal updated preconditioner for $A_k$:

$$A_k \approx L(D + L^{-1}(A_k - A_{seed})U^{-1})U$$

The approximate updated preconditioner is obtained as follows:

1. Let $f(M) = \text{band}(M, k_l, k_u)$, be the banded approximation of $M$ with $k_l$ lower and $k_u$ upper diagonals.

2. Let

$$E_k = f(A_k - A_{seed}), \quad F_k = f(L^{-1}E_k U^{-1}),$$

and

$$P_k = L(D + F_k)U.$$
Motivation: matrices where the entries of the inverse tend to zero away from the main diagonal.

- banded SPD and indefinite matrices [Demko, Moss, Smith 1984][Meurant 1992];
- nonsymmetric block tridiagonal matrices [Nabben 1999];
- matrices $h(A)$ with $A$ symm and banded and $h$ analytic [Benzi, Golub 99].

2D Nonlinear Convection diffusion problem. Sparsity pattern (on the left) and wireframe mesh (on the right) of the inverses of the L and U factors obtained from the ILU factorization of the Jacobian at the null vector ($n = 400$).
Small bandwidth values $k_l$ and $k_u$ are viable.

Only selected elements of $A_k$ are required: nearly matrix-free strategies.

Forming/approximating $L^{-1}$ and $U^{-1}$:
- Use banded approximation of $L^{-1}$ and $U^{-1}$, computable without the need of a complete inversion of $L$ and $U$.
  [B., Morini, Porcelli 2014]

The application of the preconditioner requires the solution of one banded linear system.

The computationally most convenient approximations $E_k$ and $F_k$ are diagonal ($k_l = k_u = 0$).
Diagonally Updated ILU (DU_ILU)

Assume $k_l = k_u = 0$, Let $\mathcal{P}_{seed} = LDU$.

1. Consider

$$A_k \approx L(D + L^{-1}(A_k - A_{seed})U^{-1})U \approx LDU + \frac{\text{diag}(A_k - A_{seed})}{\Sigma_k = \text{diag}(\sigma_{11}^k, \ldots, \sigma_{nn}^k)}$$

2. Form the approximate factorization $\mathcal{P}_k = L_k D_k U_k$ for $LDU + \Sigma_k$

$$D_k = D + \Sigma_k,$$
$$L_k = \text{eye}(n), \quad \text{off}(L_k) = \text{off}(L)Z_k,$$
$$U_k = \text{eye}(n), \quad \text{off}(U_k) = Z_k \text{off}(U),$$

$$Z_k = \text{diag}(z_{11}^k, \ldots, z_{nn}^k), \quad z_{ii}^k = \frac{|d_{ii}|}{|d_{ii}| + |\sigma_{ii}^k|}, \quad i = 1, \ldots, n$$

Generalization of [B., De Simone, di Serafino, Morini 2012].
Properties of $DU_{ILU}$

Scaling matrix $Z_k = \text{diag}(z_{11}^k, \ldots, z_{nn}^k)$:

$$z_{ii}^k = \frac{|d_{ii}|}{|d_{ii}| + |\sigma_{ii}^k|}, \quad i = 1, \ldots, n,$$

- Since $z_{ii}^k \in (0, 1]$, the conditioning of $L_k$ and $U_k$ is at least as good as the conditioning of $L$ and $U$ respectively [Lemeire 1975]. The sparsity pattern of $L$ and $U$ is preserved.
Properties of $DU_{ILU}$

Scaling matrix $Z_k = diag(z_{11}^k, \ldots, z_{nn}^k)$:

$$z_{ii}^k = \frac{|d_{ii}|}{|d_{ii}| + |\sigma_{ii}^k|}, \quad i = 1, \ldots, n,$$

- Since $z_{ii}^k \in (0, 1]$, the conditioning of $L_k$ and $U_k$ is at least as good as the conditioning of $L$ and $U$ respectively [Lemeire 1975]. The sparsity pattern of $L$ and $U$ is preserved.
- The preconditioner mimics the behavior of the matrix $LDU + \Sigma_k$:
  - $off(L_k)$ and $off(U_k)$ decrease in absolute value as the entries of $\Sigma_k$ increase, i.e. when the diagonal of $LDU + \Sigma_k$ tends to dominate over the remaining entries.
Properties of $\mathbf{DU_{ILU}}$

Scaling matrix $Z_k = \text{diag}(z_{11}^k, \ldots, z_{nn}^k)$:

$$z_{ii}^k = \frac{|d_{ii}|}{|d_{ii}| + |\sigma_{ii}^k|}, \quad i = 1, \ldots, n,$$

- Since $z_{ii}^k \in (0, 1]$, the conditioning of $L_k$ and $U_k$ is at least as good as the conditioning of $L$ and $U$ respectively [Lemeire 1975]. The sparsity pattern of $L$ and $U$ is preserved.
- The preconditioner mimics the behavior of the matrix $LDU + \Sigma_k$:
  - $\text{off}(L_k)$ and $\text{off}(U_k)$ decrease in absolute value as the entries of $\Sigma_k$ increase, i.e. when the diagonal of $LDU + \Sigma_k$ tends to dominate over the remaining entries.
  - If the entries of $\Sigma_k$ are small then $LDU + \Sigma_k$ is close to $LDU$ and $Z_k$ is close to the identity matrix.

[B., Morini, Porcelli 2014], [B., Porcelli, 2014]
Properties of $\text{DU}_{\text{ILU}}$ (c.ed)

- Quality of $\text{DU}_{\text{ILU}}$ preconditioner

$$\|A_k - P_k\| \leq \|A_{seed} - P_{seed}\| + \|\text{off}(A_k - A_{seed})\| + c\|\Sigma_k\|$$

The upper bound depends on
- $\|A_{seed} - P_{seed}\|$: quality of the seed preconditioner;
- $\|\text{off}(A_k - A_{seed})\|$: information discarded in the update;
- $\|\text{off}(A_k - A_{seed})\|$ and $\|\Sigma_k\|$ small for slowly varying sequences.

- In order to form $\Sigma_k$, $\text{diag}(A_k)$ is needed.

  If $G$ is the finite-differences operator and it is separable then forming $\Sigma_k$ amounts to one $F$-evaluation.

- The update computational overhead is low.
Comparison with Recomputed and Frozen preconditioner

Performance profile in terms of linear iterations (left) and execution time (right)

Linesearch Newton-BiCGSTAB, $LI_{\text{max}} = 400$, dimension from $n = 6400$ to 62500, for a total of 22 test problems.
Nonlinear Convection-Diffusion problem with $n = 22500$ and $Re = 500$: comparison, in terms of LI between the Frozen and the Updated strategy. The seed preconditioner has never been recomputed.
Sequences of KKT matrices

Let $A_k$ be the KKT matrix of the form

$$A_k = \begin{bmatrix} Q + \Theta_k^{(1)} & A^T \\ A & -\Theta_k^{(2)} \end{bmatrix}$$

with

- $Q \in \mathbb{R}^{n \times n}$ symmetric positive semidefinite,
- $A \in \mathbb{R}^{m \times n}$, $0 < m \leq n$, full rank
- $\Theta_k^{(1)} \in \mathbb{R}^{n \times n}$ diagonal SPD,
- $\Theta_k^{(2)} \in \mathbb{R}^{m \times m}$ diagonal positive semidefinite.

This matrix arises at the $k$th iteration of an IP method for the convex QP problem

minimize $\frac{1}{2}x^T Q x + c^T x,$

s.t. $A_1 x - s = b_1,$ $A_2 x = b_2,$ $x + v = u,$ $(x, s, v) \geq 0,$
Constraint Preconditioners (CPs)

\[ \mathcal{P}_k = \begin{bmatrix} H_k & A^T \\ A & -\Theta_k^{(2)} \end{bmatrix} \]

- \( H_k \) “simple” symmetric approximation to \( Q + \Theta_k^{(1)} \); here \( H_k = \text{diag}(Q + \Theta_k^{(1)}) \), [Benzi, Golub, Liesen 2005]

- Factorization of CP Factorize the negative Schur complement \( S_k \) of \( H_k \) in \( \mathcal{A}_k \)

\[ S_k = AH_k^{-1}A^T + \Theta_k^{(2)} = L_k D_k L_k^T \]

Cholesky-like factorization

and let

\[ \mathcal{P}_k = \begin{bmatrix} I_n & 0 \\ AH_k^{-1} & I_m \end{bmatrix} \begin{bmatrix} H_k & 0 \\ 0 & -S_k \end{bmatrix} \begin{bmatrix} I_n & H_k^{-1}A^T \\ 0 & I_m \end{bmatrix} = \begin{bmatrix} I_n & 0 \\ AH_k^{-1} & L_k \end{bmatrix} \begin{bmatrix} H_k & 0 \\ 0 & -D_k \end{bmatrix} \begin{bmatrix} I_n & H_k^{-1}A^T \\ 0 & L_k^T \end{bmatrix}, \]
Inexact CPs

In large-scale problems, the factorization of CPs may still account for a large part of the cost of the IP iterations.

- **Approximations of CPs**: based on approximate factorizations of the Schur complement or on sparse approximations of $A$
  
  [Lukšan, Vlček, 1998], [Perugia, Simoncini 2000], [Durazzi, Ruggiero 2002],
  [Bergamaschi, Gondzio, Venturin, Zilli, 2007].

  No exploitation of CPs for previous matrices in the sequence.
Inexact CPs

In large-scale problems, the factorization of CPs may still account for a large part of the cost of the IP iterations.

- **Approximations of CPs**: based on approximate factorizations of the Schur complement or on sparse approximations of $A$
  [Lukšan, Vlček, 1998], [Perugia, Simoncini 2000], [Durazzi, Ruggiero 2002],
  [Bergamaschi, Gondzio, Venturin, Zilli, 2007].

No exploitation of CPs for previous matrices in the sequence.

- **Our focus is on inexact CPs** of the form

\[
(P_k)^{\text{inex}} = \begin{bmatrix}
I_n & 0 \\
AH_k^{-1} & I_m
\end{bmatrix}
\begin{bmatrix}
H_k & 0 \\
0 & -(S_k)^{\text{inex}}
\end{bmatrix}
\begin{bmatrix}
I_n & H_k^{-1}A^T \\
0 & I_m
\end{bmatrix}
\]

where

- $(S_k)^{\text{inex}}$ is a SPD matrix;
- $(S_k)^{\text{inex}}$ is computationally cheaper than $S_k$. 
Inexact CPs built by updating

1. Given
   \[ A_{\text{seed}} = \begin{bmatrix} Q + \Theta_{\text{seed}}^{(1)} & A^T \\ A & -\Theta_{\text{seed}}^{(2)} \end{bmatrix} \]
   \[ S_{\text{seed}} = AH^{-1}A^T + \Theta_{\text{seed}}^{(2)} = LDL^T \]
   \[ P_{\text{seed}} = \begin{bmatrix} I_n & 0 \\ AH^{-1} & I_m \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & -S_{\text{seed}} \end{bmatrix} \begin{bmatrix} I_n & H^{-1}A^T \\ 0 & I_m \end{bmatrix} \text{ seed CP} \]

2. Let
   \[ A = \begin{bmatrix} Q + \Theta^{(1)} & A^T \\ A & -\Theta^{(2)} \end{bmatrix}, \quad G = \text{diag}(Q + \Theta^{(1)}) \]
   \[ S = AG^{-1}A^T + \Theta^{(2)} \]

Form an inexact CP where \( S \) is replaced by a SPD matrix obtained by updating \( S_{\text{seed}} \).
Updating CPs: our strategy

Given the KKT matrix $A_{seed}$ and the corresponding CP $P_{seed}$:

$$P_{seed} = \begin{bmatrix}
I_n & 0 \\
AH^{-1} & I_m
\end{bmatrix} \begin{bmatrix}
H & 0 \\
0 & -S_{seed}
\end{bmatrix} \begin{bmatrix}
I_n & HA^T \\
0 & I_m
\end{bmatrix}$$

$$S_{seed} = AH^{-1}A^T + \Theta_{seed}^{(2)} = LDL^T$$

build an updated preconditioner for a subsequent KKT matrix $A$ as follows:

$$P_{upd} = \begin{bmatrix}
I_n & 0 \\
AG^{-1} & I_m
\end{bmatrix} \begin{bmatrix}
G & 0 \\
0 & -S_{upd}
\end{bmatrix} \begin{bmatrix}
I_n & G^{-1}A^T \\
0 & I_m
\end{bmatrix}$$

$$S_{upd} = \text{factorized update of } S_{seed} \text{ that approximates } S = AG^{-1}A^T + \Theta^{(2)}$$
Updating CPs: our strategy (cont’d)

The real and imag parts of the eigs of $\mathcal{P}^{-1}_{\text{upd}} A$ are bounded in terms of the eigs of $S^{-1}_{\text{upd}} S$.

**Goal:** define an approximation $S_{\text{upd}}$ to $S$ such that

- “good” and easily-computable bounds on the eigs of $S^{-1}_{\text{upd}} S$ can be obtained.
- the factorization of $S_{\text{upd}}$ can be obtained by a low-cost update of the $LDL^T$ factorization of $S_{\text{seed}}$. 
Defining $S_{upd}$ ($\Theta^{(2)}$, $\Theta_{seed}^{(2)} = 0$ for simplicity)

\[
S_{seed} = AH^{-1}A^T, \quad S = AG^{-1}A^T
\]
Defining $S_{\text{upd}}$ $(\Theta^{(2)}, \Theta_{\text{seed}}^{(2)} = 0$ for simplicity)

\[ S_{\text{seed}} = AH^{-1}A^T, \quad S = AG^{-1}A^T \]

\[ S_{\text{upd}} = AJ^{-1}A^T \]

- $J$ diagonal positive definite
Defining $S_{upd}$ ($\Theta^{(2)}$, $\Theta^{(2)}_{seed} = 0$ for simplicity)

$$S_{seed} = AH^{-1}A^T, \quad S = AG^{-1}A^T$$

$$S_{upd} = AJ^{-1}A^T = A(H^{-1} + K)A^T$$

- $J$ diagonal positive definite
- $K$ diagonal with only $q < n$ nonzero entries
Defining $S_{upd}$ \((\Theta^{(2)}, \Theta^{(2)}_{seed} = 0 \; \text{for simplicity})\)

\[
S_{seed} = AH^{-1}A^T, \quad S = AG^{-1}A^T
\]

\[
S_{upd} = AJ^{-1}A^T = A(H^{-1} + K)A^T = AH^{-1}A^T + \bar{A}\bar{K}\bar{A}^T
\]

- $J$ diagonal positive definite
- $K$ diagonal with only $q < n$ nonzero entries
- $\bar{K}$ principal submatrix of $K$ containing these nonzero entries,
  $\bar{A}$ corresponding columns of $A$
Defining $S_{\text{upd}}$ ($\Theta^{(2)}, \Theta^{(2)}_{\text{seed}} = 0$ for simplicity)

\[
S_{\text{seed}} = AH^{-1}A^T, \quad S = AG^{-1}A^T
\]

\[
S_{\text{upd}} = AJ^{-1}A^T = A(H^{-1} + K)A^T = AH^{-1}A^T + \tilde{A}\tilde{K}\tilde{A}^T
\]

- $J$ diagonal positive definite
- $K$ diagonal with only $q < n$ nonzero entries
- $\tilde{K}$ principal submatrix of $K$ containing these nonzero entries, $\tilde{A}$ corresponding columns of $A$

Results:

- $\lambda_{\text{min}}(JG^{-1}) \leq \lambda(S^{-1}_{\text{upd}}S) \leq \lambda_{\text{max}}(JG^{-1})$,
- for small $q$, $S_{\text{upd}}$ is a low-rank correction of $S_{\text{seed}}$
Choosing \( J (S_{\text{seed}} = AH^{-1}A^T, S = AG^{-1}A^T, S_{\text{upd}} = AJ^{-1}A^T) \)

- Let \( \lambda_i = \lambda_i(HG^{-1}) \) and assume \( \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \).
- Choose \( q_1 \) and \( q_2 \) integers such that \( q = q_1 + q_2 \leq n \) and set

\[
\Gamma = \{ \text{indices } i \text{ corresponding to the } q_1 \text{ largest } \lambda_i > 1 \}
\]

\[
\{ q_2 \text{ smallest } \lambda_i < 1 \} \} \]
Choosing $J$ ($S_{seed} = AH^{-1}A^T$, $S = AG^{-1}A^T$, $S_{upd} = AJ^{-1}A^T$)

- Let $\lambda_i = \lambda_i(HG^{-1})$ and assume $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.
- Choose $q_1$ and $q_2$ integers such that $q = q_1 + q_2 \leq n$ and set
  \[ \Gamma = \{ \text{indices } i \text{ corresponding to the } q_1 \text{ largest } \lambda_i > 1 \text{ and } q_2 \text{ smallest } \lambda_i < 1 \} \]
- Set
  \[ J_{ii} = \begin{cases} 
  G_{ii}, & \text{if } i \in \Gamma \\
  H_{ii}, & \text{otherwise}
  \end{cases} \]
Choosing $J$ ($S_{seed} = AH^{-1}A^T$, $S = AG^{-1}A^T$, $S_{upd} = AJ^{-1}A^T$)

- Let $\lambda_i = \lambda_i(HG^{-1})$ and assume $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$.
- Choose $q_1$ and $q_2$ integers such that $q = q_1 + q_2 \leq n$ and set
  \[
  \Gamma = \{ \text{indices } i \text{ corresponding to the } q_1 \text{ largest } \lambda_i > 1, q_2 \text{ smallest } \lambda_i < 1 \} \]
- Set
  \[
  J_{ii} = \begin{cases} 
  G_{ii}, & \text{if } i \in \Gamma \\
  H_{ii}, & \text{otherwise}
  \end{cases}
  \]

Then
\[
\lambda_{\min}(JG^{-1}) = \min \{1, \min_{j \notin \Gamma} H_{jj}/G_{jj}\} = \min \{1, \lambda_{q_2+1}\}
\]
\[
\lambda_{\max}(JG^{-1}) = \max \{1, \max_{j \notin \Gamma} H_{jj}/G_{jj}\} = \max \{1, \lambda_{n-q_1}\}
\]
Choosing $J$ (cont’d)
Choosing $J$ (cont’d)

\[ \lambda(S_{upd}^{-1}S) \]
Choosing $J$ (cont’d)

\[
\lambda_{\min}(JG^{-1}) \leq \lambda(S_{\text{upd}}^{-1}S) \leq \lambda_{\max}(JG^{-1})
\]
Choosing $J$ (cont’d)

\[
\min \{1, \lambda_{q_2+1}\} = \lambda_{\min}(JG^{-1}) \leq \lambda(S^{-1}_{\text{upd}}S) \leq \lambda_{\max}(JG^{-1}) = \max \{1, \lambda_{n-q_1}\}
\]
Choosing $J$ (cont’d)

$$\min \{1, \lambda_{q_2+1}\} = \lambda_{\min}(JG^{-1}) \leq \lambda(S_{\text{upd}}^{-1}S) \leq \lambda_{\max}(JG^{-1}) = \max \{1, \lambda_{n-q_1}\}$$

$$\lambda_1 \leq \cdots \leq \lambda_{q_2} \leq \lambda_{q_2+1} \leq \cdots \leq \lambda_{n-q_1} \leq \lambda_{n-q_1+1} \leq \cdots \leq \lambda_n$$
Choosing \( J \) (cont’d)

\[
\begin{align*}
\min \{1, \lambda_{q_2+1}\} &= \lambda_{\min}(JG^{-1}) \leq \lambda(S_{upd}^{-1}S) \leq \lambda_{\max}(JG^{-1}) = \max \{1, \lambda_{n-q_1}\} \\
\lambda_{q_2+1} &\leq \cdots \leq \lambda_{n-q_1}
\end{align*}
\]
Choosing $J$ (cont’d)

$$\min \left\{ 1, \lambda_{q_2+1} \right\} = \lambda_{\min}(JG^{-1}) \leq \lambda(S_{upd}^{-1}S) \leq \lambda_{\max}(JG^{-1}) = \max \left\{ 1, \lambda_{n-q_1} \right\}$$

$$\lambda_{q_2+1} \leq \cdots \leq \lambda_{n-q_1}$$

The more $\lambda_{q_2+1}(HG^{-1})$ and $\lambda_{n-q_1}(HG^{-1})$ are separated from $\lambda_{q_2}(HG^{-1})$ and $\lambda_{n-q_1+1}(H)G^{-1})$ the better the bounds on the eigenvalues of $S_{upd}^{-1}S$ are
Computing the factorization of $S_{upd}$

$$S_{upd} = S_{seed} + AKA^T = LDL^T + \bar{A}\bar{K}\bar{A}^T$$

- $K_{ii} = G_{ii}^{-1} - H_{ii}^{-1}$, if $i \in \Gamma$, $K_{ii} = 0$ otherwise.
- $\bar{A}\bar{K}\bar{A}^T$ has rank $q = q_1 + q_2$ and, if $q \ll n$, the factorization
  $$S_{upd} = L_{upd}D_{upd}L_{upd}^T$$
  can be computed at low cost by a rank-$q$ update of $S_{seed} = LDL^T$

Spectra of $A$, $P_{rec}^{-1}A$, $P_{upd}^{-1}A$ CVXQP1 ($n=1000$, $m=500$), $q_1 = q_2 = 25$
Numerical results

- Updating strategy integrated into the Fortran IP solver PRQP (Potential Reduction solver for Quadratic Programming) [Cafieri, D'Apuzzo, De Simone, di Serafino, Toraldo, 2007-2010]
- Solution of KKT systems by SQMR
- Sparse $LDL^T$ and low-rank update of Schur complement by CHOLMOD [Davis, Hager, 2009]
- Adaptive criterion for choosing when to recompute $P_{rec}$ (based on time and iterations)
- Convex quadratic problems from CUTEst
### Numerical results (extremely sparse Schur complement)

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n, m$</th>
<th>nnz($S$)</th>
<th>$\mathcal{P}_{\text{rec}}$</th>
<th>$\mathcal{P}_{\text{upd}}$ ($q=50$)</th>
<th>$\mathcal{P}_{\text{upd}}$ ($q=100$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IPits</td>
<td>its</td>
<td>time</td>
</tr>
<tr>
<td>CVXQP1</td>
<td>20000</td>
<td>10000</td>
<td>16</td>
<td>209</td>
<td>2.07e+0</td>
</tr>
<tr>
<td>CVXQP3</td>
<td>20000</td>
<td>15000</td>
<td>35</td>
<td>523</td>
<td>8.04e+0</td>
</tr>
<tr>
<td>STCQP2</td>
<td>16385</td>
<td>8190</td>
<td>12</td>
<td>226</td>
<td>1.46e+0</td>
</tr>
<tr>
<td>CVXQP1-M</td>
<td>20000</td>
<td>10000</td>
<td>26</td>
<td>1015</td>
<td>7.65e+0</td>
</tr>
<tr>
<td>CVXQP3-M</td>
<td>15000</td>
<td>11250</td>
<td>30</td>
<td>1261</td>
<td>1.47e+1</td>
</tr>
<tr>
<td>MOSARQP1</td>
<td>22500</td>
<td>20000</td>
<td>16</td>
<td>66</td>
<td>4.68e+0</td>
</tr>
<tr>
<td>QPBAND</td>
<td>50000</td>
<td>25000</td>
<td>12</td>
<td>757</td>
<td>7.13e+0</td>
</tr>
</tbody>
</table>
### Numerical results (less sparse Schur complement)

<table>
<thead>
<tr>
<th>Problem</th>
<th>$n$, $m$</th>
<th>$nnz(S)$</th>
<th>$P_{\text{rec}}$</th>
<th>$P_{\text{upd}}$ ($q=50$)</th>
<th>$P_{\text{upd}}$ ($q=100$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$IPits$ $its$ $time$</td>
<td>$IPits$ $its$ $time$</td>
<td>$IPits$ $its$ $time$</td>
</tr>
<tr>
<td>CVXQP1-D</td>
<td>20000,10000</td>
<td>240494</td>
<td>15 239 2.95e+2</td>
<td>15 616 9.91e+1</td>
<td>15 602 1.03e+2</td>
</tr>
<tr>
<td>CVXQP3-D</td>
<td>20000,15000</td>
<td>542296</td>
<td>15 192 1.03e+3</td>
<td>15 526 4.40e+2</td>
<td>15 481 4.55e+2</td>
</tr>
<tr>
<td>CVXQP3-D2</td>
<td>20000,15000</td>
<td>224396</td>
<td>15 288 9.95e+1</td>
<td>17 819 5.26e+1</td>
<td>17 802 5.46e+1</td>
</tr>
<tr>
<td>STCQP2-D</td>
<td>16385,8190</td>
<td>5003908</td>
<td>12 238 6.08e+2</td>
<td>12 262 1.22e+2</td>
<td>12 262 1.22e+2</td>
</tr>
<tr>
<td>CVXQP1-M-D</td>
<td>20000,10000</td>
<td>240494</td>
<td>28 1090 5.85e+2</td>
<td>28 3665 3.23e+2</td>
<td>27 3514 3.24e+2</td>
</tr>
<tr>
<td>CVXQP3-M-D</td>
<td>20000,15000</td>
<td>542296</td>
<td>25 910 1.93e+3</td>
<td>25 3416 8.89e+2</td>
<td>25 3317 9.07e+2</td>
</tr>
<tr>
<td>CVXQP3-M-D2</td>
<td>20000,15000</td>
<td>224396</td>
<td>25 822 1.66e+2</td>
<td>25 2645 1.33e+2</td>
<td>25 2148 1.25e+2</td>
</tr>
<tr>
<td>MOSARQP1-D</td>
<td>22500,20000</td>
<td>573216</td>
<td>24 93 4.94e+1</td>
<td>22 599 3.00e+1</td>
<td>22 440 2.78e+1</td>
</tr>
<tr>
<td>QPBAND-D</td>
<td>50000,25000</td>
<td>149988</td>
<td>11 717 1.06e+3</td>
<td>11 2619 4.36e+2</td>
<td>11 2612 4.51e+2</td>
</tr>
</tbody>
</table>
### Updating constraint preconditioners

#### Numerical results: some details (problem CVXQP3-D)

<table>
<thead>
<tr>
<th>IP it</th>
<th>$P_{\text{rec}}$</th>
<th>$P_{\text{upd}}$ (q=50)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\text{its}$</td>
<td>$T_{\text{fact}}$</td>
</tr>
<tr>
<td>1</td>
<td>30</td>
<td>5.18e+0</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>5.16e+0</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>5.16e+0</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>5.12e+0</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5.14e+0</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>5.16e+0</td>
</tr>
<tr>
<td>7</td>
<td>10</td>
<td>5.15e+0</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>5.16e+0</td>
</tr>
<tr>
<td>9</td>
<td>14</td>
<td>5.13e+0</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>5.18e+0</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>5.14e+0</td>
</tr>
<tr>
<td>12</td>
<td>17</td>
<td>5.17e+0</td>
</tr>
<tr>
<td>13</td>
<td>19</td>
<td>5.14e+0</td>
</tr>
<tr>
<td>14</td>
<td>21</td>
<td>5.15e+0</td>
</tr>
<tr>
<td>15</td>
<td>24</td>
<td>5.15e+0</td>
</tr>
<tr>
<td>16</td>
<td>26</td>
<td>5.13e+0</td>
</tr>
<tr>
<td>17</td>
<td>47</td>
<td>5.27e+0</td>
</tr>
<tr>
<td></td>
<td>288</td>
<td>8.77e+1</td>
</tr>
</tbody>
</table>
More details on the updating technique described so far in

- B., Morini, Porcelli, New updates of incomplete LU factorizations and applications to large nonlinear systems, Optimization Methods and Software, 2014.
- B., De Simone, di Serafino, Morini, Updating constraint preconditioners for KKT systems in quadratic programming via low-rank corrections, SIAM J. Opt., to appear
- B., De Simone, di Serafino, Morini, On the update of constraint preconditioners for regularized KKT systems, 2015, submitted

http://www.optimization-online.org/DB_HTML/2014/03/4283.html

Thank you for your attention!
Application of $DU_{\text{ILU}}$ to Newton-Krylov methods + linesearch

- Implementation in a nearly matrix-free manner, $\text{diag}(J_k)$ is computed by finite differences.

- Safeguard against the risk of singular or nearly singular middle factors $D_k$ in the updated preconditioners,
  
  - If singularity is detected $\Rightarrow$ breakdown
  - If
    
    $$\min_{i=1,\ldots,n} |(D_k)_{ii}| \leq \tau \|J_{\text{seed}}\|_1,$$
    
    for some small positive $\tau \Rightarrow$ preconditioner from the previous Newton iteration is frozen.

[Bellavia, Bertaccini, M. 2011]
The two-dimensional nonlinear convection-diffusion model problem has the form,

$$-\Delta u + Re \, u(u_x + u_y) = f(x, y) \quad \text{in } \Omega = [0, 1] \times [0, 1],$$

$$u = 0 \quad \text{in } \partial \Omega,$$

where \( f(x, y) = 2000x(1 - x)y(1 - y) \), and \( Re \) is the Reynolds number. We discretized this problem using second order centered finite differences on a uniform \( m \times m \) grid.