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

$$\mathcal{A}_k x = b_k, \quad k = 1, \ldots$$

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

- Computing preconditioners $\mathcal{P}_1, \mathcal{P}_2, \ldots$, 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 $\mathcal{P}_{seed}$ for some *seed* matrix $\mathcal{A}_{seed}$ of the sequence, updated preconditioners for subsequent matrices $\mathcal{A}_k$ 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 $\mathcal{P}_{seed}$ for some *seed* matrix $\mathcal{A}_{seed}$ of the sequence, updated preconditioners for subsequent matrices $\mathcal{A}_k$ 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.
- $\mathcal{A}_k = J(x_k)$,
- $\mathcal{A}_k v$ provided by an operator or approximated by finite-differences, i.e.

$$\mathcal{A}_k v \simeq \frac{F(x_k + \epsilon v) - F(x_k)}{\epsilon \|v\|} \quad \epsilon > 0. \tag{1}$$

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 $\mathcal{A}_k's$ are required.
  - matrix-vector product approximations by finite differences can be used.

[Knoll, Keyes 2004]

Preconditioning & Matrix-free setting c.ed

Let $\mathcal{G}$ be the function that, evaluated at $v \in \mathbb{R}^n$, provides the product of $\mathcal{A}_k$ times v.

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

Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:

- symmetric positive definite (SPD) matrices and nonsymmetric matrices arising in Newton methods: [Morales, Nocedal 2000], [Bergamaschi, Bru, Martinez, Putti 2006], [Gratton, Sartenaer, Tshimanga 2011], [Gower, Gondzio 2014].

Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:

- symmetric positive definite (SPD) matrices and nonsymmetric matrices arising in Newton methods: [Morales, Nocedal 2000], [Bergamaschi, Bru, Martinez, Putti 2006], [Gratton, Sartenaer, Tshimanga 2011], [Gower, Gondzio 2014].

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].

Updating frameworks in literature

Limited-memory Quasi-Newton preconditioners:

- symmetric positive definite (SPD) matrices and nonsymmetric matrices arising in Newton methods: [Morales, Nocedal 2000], [Bergamaschi, Bru, Martinez, Putti 2006], [Gratton, Sartenaer, Tshimanga 2011], [Gower, Gondzio 2014].

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:

- SPD matrices and nonsymmetric matrices: [Meurant 2001], [Benzi, Bertaccini 2003], [Duintjer Tebbens, Tuma 2007, 2010], [B., Bertaccini, Morini 2011], [B., De Simone, di Serafino, Morini 2011-2015],[B., Morini, Porcelli 2014]

Approximate updates of factorized preconditioners

Consider two linear systems

$$\mathcal{A}_{seed} x = b, \qquad \mathcal{A}_k x = b_k$$

and let $\mathcal{P}_{seed} = LDU \approx \mathcal{A}_{seed}$.

- It follows

$$\mathcal{A}_k = \mathcal{A}_{seed} + (\mathcal{A}_k - \mathcal{A}_{seed}) \approx L(\, D + \underbrace{L^{-1}(\mathcal{A}_k - \mathcal{A}_{seed})U^{-1}}_{\text{ideal update}}\,)U$$

- The *ideal* update of the middle-term is costly:
  - the difference matrix $\mathcal{A}_k - \mathcal{A}_{seed}$ should be formed;
  - in general the ideal update is dense and its factorization is impractical.
- Form an *approximate* and cheap update.

Update of LDU factorizations [Duintjer Tebbens, Tuma 2007, 2010]

Ideal updated preconditioner for $\mathcal{A}_k$:

$$\mathcal{A}_k \approx L(D + \underbrace{L^{-1}(\mathcal{A}_k - \mathcal{A}_{seed})U^{-1}})U$$

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):

$$\begin{aligned} \mathcal{A}_k &\approx L(\,D + (\mathcal{A}_k - \mathcal{A}_{seed})U^{-1}\,)U \\ \mathcal{A}_k &\approx L(\,D + L^{-1}(\mathcal{A}_k - \mathcal{A}_{seed})\,)U \end{aligned}$$

2. Use only a triangular part of the current matrix $\mathcal{A}_k$:

$$\begin{aligned} \mathcal{P}_k &= L(\,DU + triu(\mathcal{A}_k - \mathcal{A}_{seed})) \\ \mathcal{P}_k &= (L\,D + tril(\mathcal{A}_k - \mathcal{A}_{seed})\,)U \end{aligned}$$

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

Banded approximate factors

Ideal updated preconditioner for $\mathcal{A}_k$:

$$\mathcal{A}_k \approx L(D + \underbrace{L^{-1}(\mathcal{A}_k - \mathcal{A}_{seed})U^{-1}})U$$

The approximate updated preconditioner is obtained as follows:

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

2. Let

$$E_k \;=\; f(\mathcal{A}_k - \mathcal{A}_{seed}), \qquad F_k = f(L^{-1}\, E_k\, U^{-1}),$$

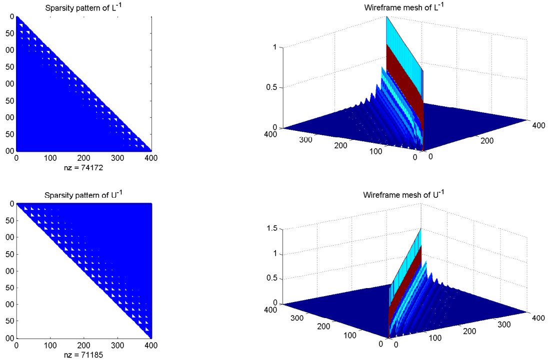
   and

$$\mathcal{P}_k = L\,(D + F_k)\,U.$$

[Benzi, Golub 1999], [Benzi, Bertaccini 2003], [B., Bertaccini, Morini 2011], [B., Morini, Porcelli 2014]

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 the Approximate INVerse (AINV) preconditioner [Benzi, Meyer, Tuma 1996], [Benzi, Tuma 1998],[B.,Morini,Bertaccini, 2011]
  - Use banded approximation of L^{-1} and U^{-1}, computable without the neeed 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

$$\mathcal{A}_k \approx L(D + \underbrace{L^{-1}(\mathcal{A}_k - \mathcal{A}_{seed})U^{-1}})U \simeq LDU + \underbrace{diag(\mathcal{A}_k - \mathcal{A}_{seed})}_{\Sigma_k = diag(\sigma_{11}^k, \ldots, \sigma_{nn}^k)}$$

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

$$\begin{aligned} D_k &= D + \Sigma_k, \\ L_k &= eye(n), \quad off(L_k) = off(L) Z_k \\ U_k &= eye(n), \quad off(U_k) = Z_k off(U) \\ Z_k &= diag(z_{11}^k, \ldots, z_{nn}^k), \quad z_{ii}^k = \frac{|d_{ii}|}{|d_{ii}| + |\sigma_{ii}^k|}, \; i = 1, \ldots, n \end{aligned}$$

Generalization of [B., De Simone, di Serafino, Morini 2012].

Properties of DU_ILU

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

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

- 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 $Z_k = diag(z_{11}^k, \dots, z_{nn}^k)$:

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

- 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 Σ_k increase, i.e. when the diagonal of $LDU + \Sigma_k$ tends to dominate over the remaining entries.

Properties of DU_ILU

Scaling matrix $Z_k = diag(z^k_{11}, \dots, z^k_{nn})$:

$$z^k_{ii} = \frac{|d_{ii}|}{|d_{ii}| + |\sigma^k_{ii}|}, \quad i = 1, \dots, n,$$

- Since $z^k_{ii} \in (0, 1]$, 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 Σ_k increase, i.e. when the diagonal of $LDU + \Sigma_k$ tends to dominate over the remaining entries.
  - If the entries of Σ_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 DU_ILU (c.ed)

- Quality of DU_ILU preconditioner

$$\|\mathcal{A}_k - \mathcal{P}_k\| \leq \|\mathcal{A}_{seed} - \mathcal{P}_{seed}\| + \|off(\mathcal{A}_k - \mathcal{A}_{seed})\| + c\|\Sigma_k\|$$

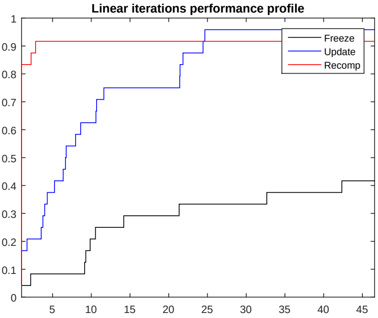
  The upper bound depends on
  - $\|\mathcal{A}_{seed} - \mathcal{P}_{seed}\|$: quality of the seed preconditioner;
  - $\|off(\mathcal{A}_k - \mathcal{A}_{seed})\|$: information discarded in the update;
  - $\|off(\mathcal{A}_k - \mathcal{A}_{seed})\|$ and $\|\Sigma_k\|$ small for slowly varying sequences.

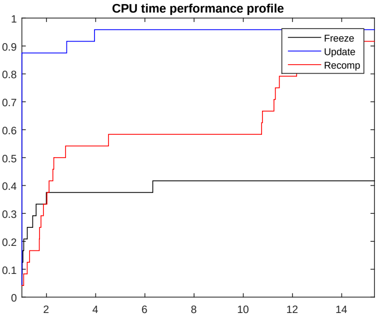
- In order to form Σ_k, $diag(\mathcal{A}_k)$ is needed.

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

- The update computational overhead is low.

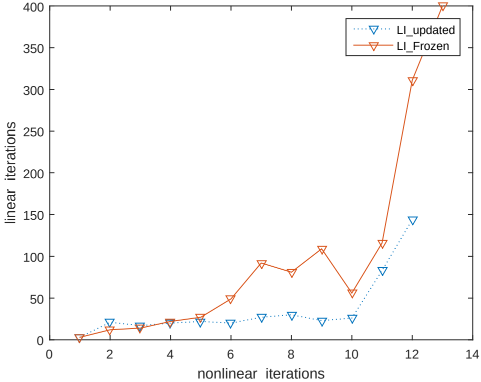
Comparison with Recomputed and Frozen preconditioner





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

Linesearch Newton-BiCGSTAB, $\texttt{LI}_{\texttt{max}} = 400$, 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 $\mathcal{A}_k$ be the KKT matrix of the form

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

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 kth iteration of an IP method for the convex QP problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{2}x^T Q x + c^T x, \\ &\text{s. t.} \qquad A_1 x - s = b_1, \;\; A_2 x = b_2, \;\; x + v = u, \;\; (x, s, v) \geq 0, \end{aligned}$$

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 $H_k = diag(Q + \Theta_k^{(1)})$, [Benzi, Golub, Liesen 2005]
- Factorization of CP Factorize the negative Schur complement S_k of H_k in $\mathcal{A}_k$

$$S_k = AH_k^{-1}A^T + \Theta_k^{(2)} = L_k D_k L_k^T \quad \text{Cholesky-like factorization}$$

and let

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

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

$$(\mathcal{P}_k)_{inex} = \begin{bmatrix} I_n & 0 \\ AH_k^{-1} & I_m \end{bmatrix} \begin{bmatrix} H_k & 0 \\ 0 & -(S_k)_{inex} \end{bmatrix} \begin{bmatrix} I_n & H_k^{-1}A^T \\ 0 & I_m \end{bmatrix}$$

  where

  - $(S_k)_{inex}$ is a SPD matrix;
  - $(S_k)_{inex}$ is computationally cheaper than S_k.

Inexact CPs built by updating

1. Given

$$\mathcal{A}_{seed} = \begin{bmatrix} Q + \Theta^{(1)}_{seed} & A^T \\ A & -\Theta^{(2)}_{seed} \end{bmatrix}$$

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

$$\mathcal{P}_{seed} = \begin{bmatrix} I_n & 0 \\ AH^{-1} & I_m \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & -S_{seed} \end{bmatrix} \begin{bmatrix} I_n & H^{-1}A^T \\ 0 & I_m \end{bmatrix} \text{ seed CP}$$

2. Let

$$\mathcal{A} = \begin{bmatrix} Q + \Theta^{(1)} & A^T \\ A & -\Theta^{(2)} \end{bmatrix}, \qquad G = diag(Q + \Theta^{(1)})$$

$$S = AG^{-1}A^T + \Theta^{(2)}$$

Form an inexact CP where S is replaced by a SPD matrix obtained by updating S_{seed}.

Updating CPs: our strategy

Given the KKT matrix $\mathcal{A}_{seed}$ and the corresponding CP $\mathcal{P}_{seed}$:

$$\mathcal{P}_{seed} = \begin{bmatrix} I_n & 0 \\ A\,H^{-1} & I_m \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & -S_{seed} \end{bmatrix} \begin{bmatrix} I_n & H\,A^T \\ 0 & I_m \end{bmatrix}$$

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

build an updated preconditioner for a subsequent KKT matrix $\mathcal{A}$ as follows:

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

S_{upd} = factorized update of S_{seed} that approximates $S = AG^{-1}A^T + \Theta^{(2)}$

Updating CPs: our strategy (cont'd)

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

Goal: define an approximation S_{upd} to S such that

- "good" and easily-computable bounds on the eigs of $S_{upd}^{-1}S$ can be obtained.
- the factorization of S_{upd} can be obtained by a low-cost update of the LDL^T factorization of S_{seed}

Defining S_{upd} ($\Theta^{(2)}, \Theta^{(2)}_{seed} = 0$ for simplicity)

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

Defining S_{upd} $(\Theta^{(2)}, \Theta^{(2)}_{seed} = 0$ for simplicity$)$

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

$$S_{upd} = AJ^{-1}A^T$$

- J diagonal positive definite

Defining S_{upd} $(\Theta^{(2)}, \Theta^{(2)}_{seed} = 0$ for simplicity$)$

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

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

- J diagonal positive definite
- K diagonal with only $q < n$ nonzero entries

Defining S_{upd} $(\Theta^{(2)}, \Theta^{(2)}_{seed} = 0$ for simplicity)

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

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

- 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_{upd} $(\Theta^{(2)}, \Theta^{(2)}_{seed} = 0$ for simplicity$)$

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

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

- 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

Results:

- $\lambda_{min}(JG^{-1}) \leq \lambda(S_{upd}^{-1}S) \leq \lambda_{max}(JG^{-1})$,
- for small q, S_{upd} is a low-rank correction of S_{seed}

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 $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$.
- Choose q_1 and q_2 integers such that $q = q_1 + q_2 \le n$ and set

  $\Gamma = \{$ indices i corresponding to the q_1 largest $\lambda_i > 1$
  q_2 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

  $\Gamma = \{$ indices i corresponding to the q_1 largest $\lambda_i > 1$
  q_2 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 $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_n$.
- Choose q_1 and q_2 integers such that $q = q_1 + q_2 \le n$ and set

  $\Gamma = \{$ indices i corresponding to the q_1 largest $\lambda_i > 1$
  q_2 smallest $\lambda_i < 1$ $\}$
- Set

$$J_{ii} = \begin{cases} G_{ii}, & \text{if } \ i \in \Gamma \\ H_{ii}, & \text{otherwise} \end{cases}$$

Then

$$\lambda_{min}(JG^{-1}) = \min\left\{1, \min_{j \notin \Gamma} H_{jj}/G_{jj}\right\} = \min\left\{1, \lambda_{q_2+1}\right\}$$
$$\lambda_{max}(JG^{-1}) = \max\left\{1, \max_{j \notin \Gamma} H_{jj}/G_{jj}\right\} = \max\left\{1, \lambda_{n-q_1}\right\}$$

Choosing J (cont'd)

Choosing J (cont'd)

$$\lambda(S_{upd}^{-1} S)$$

Choosing J (cont'd)

$$\lambda_{min}(JG^{-1}) \leq \lambda(S_{upd}^{-1}S) \leq \lambda_{max}(JG^{-1})$$

Choosing J (cont'd)

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

Choosing J (cont'd)

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

$$\lambda_1 \le \cdots \le \lambda_{q_2} \le \lambda_{q_2+1} \le \cdots \le \lambda_{n-q_1} \le \lambda_{n-q_1+1} \le \cdots \le \lambda_n$$

Choosing J (cont'd)

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

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

Choosing J (cont'd)

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

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

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}

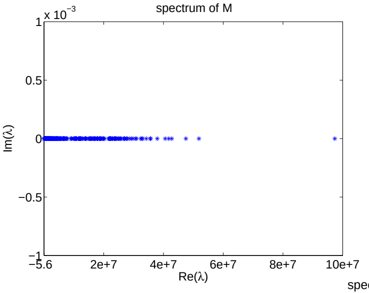
$$S_{upd} = S_{seed} + AKA^T = LDL^T + \bar{A}\bar{K}\bar{A}^T$$

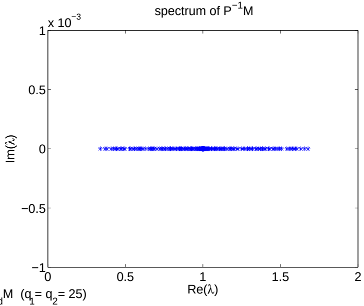
- $K_{ii} = G_{ii}^{-1} - H_{ii}^{-1}$, if $i \in \Gamma$, $K_{ii} = 0$ otherwise.
- $\bar{A}\bar{K}\bar{A}^T$ has rank $q = q_1 + q_2$ and, if $q \ll n$, the factorization

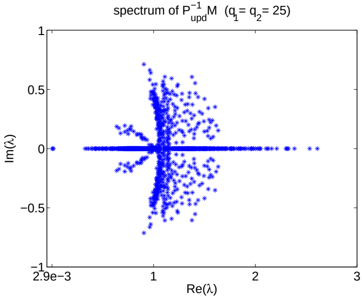
$$S_{upd} = L_{upd} D_{upd} L_{upd}^T$$

  can be computed at low cost by a rank-q update of $S_{seed} = LDL^T$
- Efficient algorithms/software available ([Gill, Golub, Murray & Saunders, 1974; Davis & Hager, 1999, 2001, 2009])

Spectra of $\mathcal{A}, \mathcal{P}_{rec}^{-1}\mathcal{A}, \mathcal{P}_{upd}^{-1}\mathcal{A}$ 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 $\mathcal{P}_{rec}$ (based on time and iterations)
- Convex quadratic problems from CUTEst

Numerical results (extremely sparse Schur complement)

Problem	n, m $nnz(S)$	$\mathcal{P}_{rec}$ *IPits*	*its*	*time*	$\mathcal{P}_{upd}$ (q=50) *IPits*	*its*	*time*	$\mathcal{P}_{upd}$ (q=100) *IPits*	*its*	*time*
CVXQP1	20000, 10000 67976	16	209	2.07e+0	16	335	2.55e+0	16	323	2.49e+0
CVXQP3	20000, 15000 155942	35	523	8.04e+0	35	755	8.86e+0	35	757	8.90e+0
STCQP2	16385, 8190 114660	12	226	1.46e+0	12	235	1.43e+0	12	235	1.43e+0
CVXQP1-M	20000, 10000 67976	26	1015	7.65e+0	26	1812	1.21e+1	26	1845	1.22e+1
CVXQP3-M	15000, 11250 155942	30	1261	1.47e+1	30	2073	2.11e+1	30	2135	2.18e+1
MOSARQP1	22500, 20000 257166	16	66	4.68e+0	16	189	4.83e+0	16	215	5.21e+0
QPBAND	50000, 25000 25000	12	757	7.13e+0	12	1600	1.43e+1	12	1599	1.37e+1

Numerical results (less sparse Schur complement)

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

Numerical results: some details (problem CVXQP3-D)

	$\mathcal{P}_{rec}$				$\mathcal{P}_{upd}$ (q=50)			
IP it	its	T_{fact}	T_{solve}	T_{sum}	its	T_{prec}	T_{solve}	T_{sum}
1	30	5.18e+0	1.19e+0	6.37e+0	30	5.24e+0	1.18e+0	6.42e+0
2	12	5.16e+0	4.87e-1	5.65e+0	14	5.52e-1	5.54e-1	1.11e+0
3	8	5.16e+0	3.40e-1	5.50e+0	15	5.49e-1	6.01e-1	1.15e+0
4	5	5.12e+0	2.24e-1	5.35e+0	13	5.11e-1	5.29e-1	1.04e+0
5	5	5.14e+0	2.27e-1	5.37e+0	36	5.52e-1	1.37e+0	1.92e+0
6	8	5.16e+0	3.37e-1	5.50e+0	48	6.00e-1	1.79e+0	2.39e+0
7	10	5.15e+0	4.15e-1	5.57e+0	10	5.24e+0	4.17e-1	5.66e+0
8	12	5.16e+0	4.93e-1	5.65e+0	15	1.56e-1	5.89e-1	7.45e-1
9	14	5.13e+0	5.61e-1	5.70e+0	22	2.76e-1	8.39e-1	1.11e+0
10	14	5.18e+0	5.58e-1	5.74e+0	41	4.82e-1	1.54e+0	2.02e+0
11	16	5.14e+0	6.33e-1	5.78e+0	78	4.90e-1	2.90e+0	3.39e+0
12	17	5.17e+0	6.68e-1	5.83e+0	139	4.66e-1	5.09e+0	5.56e+0
13	19	5.14e+0	7.40e-1	5.88e+0	19	5.25e+0	7.44e-1	5.99e+0
14	21	5.15e+0	8.11e-1	5.97e+0	31	1.95e-1	1.16e+0	1.36e+0
15	24	5.15e+0	9.24e-1	6.08e+0	62	4.68e-1	2.32e+0	2.79e+0
16	26	5.13e+0	1.39e+0	6.51e+0	86	4.72e-1	3.17e+0	3.64e+0
17	47	5.27e+0	1.76e+0	7.03e+0	160	4.61e-1	5.83e+0	6.29e+0
	288	8.77e+1	1.18e+1	9.95e+1	819	2.20e+1	3.06e+1	5.26e+1

More details on the updating technique described so far in

- B., Bertaccini, Morini, Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems, SIAM J. Sci. Comput., 2011.
- 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 $\mathrm{DU\_ILU}$ to Newton-Krylov methods + linesearch

- Implementation in a nearly matrix-free manner, $diag(J_k)$ 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 ⇒ breakdown
  - If

$$\min_{i=1,\ldots,n} |(D_k)_{ii}| \leq \tau \|J_{seed}\|_1,$$

 for some small positive τ ⇒ preconditioner from the previous Newton iteration is frozen.

[Bellavia, Bertaccini, M. 2011]

2D Nonlinear Convection diffusion problem

The two-dimensional nonlinear convection-diffusion model problem has the form,

$$\begin{aligned} -\Delta u + Re\, u(u_x + u_y) &= f(x,y) \quad \text{in } \Omega = [0,1]\times[0,1], \\ u &= 0 \qquad\quad\;\; \text{in } \partial\Omega, \end{aligned}$$

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.