A subspace-conjugate gradient method for linear matrix equations

Valeria Simoncini

Dipartimento di Matematica
Alma Mater Studiorum - Università di Bologna
valeria.simoncini@unibo.it

From joint work with Martina Iannacito and Davide Palitta

Large linear systems

Given a PDE and your preferred discretization strategy,

$$\mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{n \times n}$$

- Krylov subspace methods (CG, MINRES, GMRES, BiCGSTAB, etc.)

$$x \approx x_m = \mathcal{V}_m y_m$$

where $\mathcal{V}_m$ has (orthonormal) columns spanning $\mathbb{K}_m(\mathcal{A}, b) = \mathrm{span}\{b, \mathcal{A}b, \ldots, \mathcal{A}^{m-1}b\}$

- Preconditioners: find P such that

$$\mathcal{A}P^{-1}\widetilde{x} = b \qquad x = P^{-1}\widetilde{x}$$

where $\mathcal{A}P^{-1}$ is "easier" to solve with.

Comfort zone

Large linear systems

Given a PDE and your preferred discretization strategy,

$$\mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{n \times n}$$

- Krylov subspace methods (CG, MINRES, GMRES, BiCGSTAB, etc.)

$$x \approx x_m = \mathcal{V}_m y_m$$

where $\mathcal{V}_m$ has (orthonormal) columns spanning $\mathbb{K}_m(\mathcal{A}, b) = \mathrm{span}\{b, \mathcal{A}b, \ldots, \mathcal{A}^{m-1}b\}$

- Preconditioners: find P such that

$$\mathcal{A}P^{-1}\widetilde{x} = b \qquad x = P^{-1}\widetilde{x}$$

where $\mathcal{A}P^{-1}$ is "easier" to solve with.

Comfort zone

Large linear systems

Given a PDE and your preferred discretization strategy,

$$\mathcal{A}x = b, \quad \mathcal{A} \in \mathbb{R}^{n \times n}$$

- Krylov subspace methods (CG, MINRES, GMRES, BiCGSTAB, etc.)

$$x \approx x_m = \mathcal{V}_m y_m$$

where $\mathcal{V}_m$ has (orthonormal) columns spanning $\mathbb{K}_m(\mathcal{A}, b) = \mathrm{span}\{b, \mathcal{A}b, \ldots, \mathcal{A}^{m-1}b\}$

- Preconditioners: find P such that

$$\mathcal{A}P^{-1}\widetilde{x} = b \qquad x = P^{-1}\widetilde{x}$$

where $\mathcal{A}P^{-1}$ is "easier" to solve with.

Comfort zone

Heterogeneous variable setting

The differential problem may depend on space variable and

- Time (high quality soln of heat-, wave-type equations, dynamical systems generally)
- Parameters (e.g., coefficients with uncertainty, model tuning)

$$\Downarrow$$

Approximation space during the discretization phase: tensor space

$$\mathcal{H} \times \mathcal{S}$$

with ♣ $\mathcal{H}$: spatial variables
♣ $\mathcal{S}$: time/parameter variables

Algebraic system: $\mathcal{A}$ mixes all components, e.g.,

$$\mathcal{A} = I \otimes A + G^T \otimes I$$

Heterogeneous variable setting

The differential problem may depend on space variable and

- Time (high quality soln of heat-, wave-type equations, dynamical systems generally)
- Parameters (e.g., coefficients with uncertainty, model tuning)

$$\Downarrow$$

Approximation space during the discretization phase: tensor space

$$\mathcal{H} \times \mathcal{S}$$

with ♣ $\mathcal{H}$: spatial variables
♣ $\mathcal{S}$: time/parameter variables

Algebraic system: $\mathcal{A}$ mixes all components, e.g.,

$$\mathcal{A} = I \otimes A + G^T \otimes I$$

Heterogeneous variable setting

The differential problem may depend on space variable and

- Time (high quality soln of heat-, wave-type equations, dynamical systems generally)
- Parameters (e.g., coefficients with uncertainty, model tuning)

$$\Downarrow$$

Approximation space during the discretization phase: tensor space

$$\mathcal{H} \times \mathcal{S}$$

with ♣ $\mathcal{H}$: spatial variables
♣ $\mathcal{S}$: time/parameter variables

Algebraic system: $\mathcal{A}$ mixes all components, e.g.,

$$\mathcal{A} = I \otimes A + G^T \otimes I$$

Identity-preserving algebraic formulations

$$\mathcal{A}x = b, \qquad \mathcal{A} = I \otimes A + G^T \otimes I \qquad \mathcal{A} \in \mathbb{R}^{n \times n}, \text{ with } n = n_A n_G$$

$$\Downarrow$$

$$AX + XG = B, \qquad x = \mathrm{vec}(X), \qquad b = \mathrm{vec}(B), \quad X \in \mathbb{R}^{n_A \times n_G}$$

Pros:

- ✓ Matrices of Smaller dimension $\Rightarrow$ Reach more complex problems
- ✓ No mixing – Preserve properties of continuous problem
- ✓ Exploit algebraic structure (symmetries, rank properties...)

Identity-preserving algebraic formulations

$$\mathcal{A}x = b, \qquad \mathcal{A} = I \otimes A + G^T \otimes I \qquad \mathcal{A} \in \mathbb{R}^{n \times n}, \text{ with } n = n_A n_G$$

$$\Downarrow$$

$$AX + XG = B, \qquad x = \mathrm{vec}(X), \qquad b = \mathrm{vec}(B), \quad X \in \mathbb{R}^{n_A \times n_G}$$

Pros:

- ✓ Matrices of Smaller dimension ⇒ Reach more complex problems
- ✓ No mixing – Preserve properties of continuous problem
- ✓ Exploit algebraic structure (symmetries, rank properties...)

Identity-preserving algebraic formulations

$$\mathcal{A}x = b, \qquad \mathcal{A} = I \otimes A + G^T \otimes I \qquad \mathcal{A} \in \mathbb{R}^{n \times n}, \text{ with } n = n_A n_G$$

$$\Downarrow$$

$$AX + XG = B, \qquad x = \mathrm{vec}(X), \qquad b = \mathrm{vec}(B), \quad X \in \mathbb{R}^{n_A \times n_G}$$

Pros:

- ✓ Matrices of Smaller dimension ⇒ Reach more complex problems
- ✓ No mixing – Preserve properties of continuous problem
- ✓ Exploit algebraic structure (symmetries, rank properties...)

Exploiting rank structure

Assume B can be well represented by a low rank matrix.

$$x \in \mathbb{R}^{n_A n_G \times 1} \quad \rightarrow \quad X \approx \widetilde{X} = \begin{bmatrix} X_1 \end{bmatrix} \begin{bmatrix} X_2^T \end{bmatrix}$$

with $X_1 \in \mathbb{R}^{n_A \times k}, X_2 \in \mathbb{R}^{n_G \times k}$ tall, $k \ll n_a, n_G$

Uncover low rank approximate representation!

- Save memory allocations while approximating!
- Different interpretation: approximate soln snapshots (MOR style)
- Recognize roles at the algebraic level: use different approximations for X_1, X_2

Exploiting rank structure

Assume B can be well represented by a low rank matrix.

$$x \in \mathbb{R}^{n_A n_G \times 1} \quad \rightarrow \quad X \approx \widetilde{X} = \begin{bmatrix} X_1 \end{bmatrix} \begin{bmatrix} X_2^T \end{bmatrix}$$

with $X_1 \in \mathbb{R}^{n_A \times k}, X_2 \in \mathbb{R}^{n_G \times k}$ tall, $k \ll n_a, n_G$

Uncover low rank approximate representation!

- Save memory allocations while approximating!
- Different interpretation: approximate soln snapshots (MOR style)
- Recognize roles at the algebraic level: use different approximations for X_1, X_2

Numerical solution of the Sylvester equation

$$AX + XG^T = B$$

Various settings:

- Tiny A and G: Kron will do!
- Small A and G: Bartels-Stewart algorithm (Computes the Schur form of A and G)
- Large A and G: Iterative solution (B low rank)
  - Projection methods
  - ADI (Alternating Direction Iteration)
  - Data sparse approaches (structure-dependent)

[1]

Numerical solution of the Sylvester equation

$$AX + XG^T = B$$

Various settings:

- Tiny A and G: Kron will do!
- Small A and G: Bartels-Stewart algorithm (Computes the Schur form of A and G)
- Large A and G: Iterative solution (B low rank)
  - Projection methods
  - ADI (Alternating Direction Iteration)
  - Data sparse approaches (structure-dependent)

[1]

Numerical solution of the Sylvester equation

$$AX + XG^T = B$$

Various settings:

- Tiny A and G: Kron will do!
- Small A and G: Bartels-Stewart algorithm (Computes the Schur form of A and G)
- Large A and G: Iterative solution (B low rank)
  - Projection methods
  - ADI (Alternating Direction Iteration)
  - Data sparse approaches (structure-dependent)

[1]

Projection-type methods

Assume $B = B_1 B_2^T$.
Given two low dimensional approx spaces $\mathcal{K}_A$, $\mathcal{K}_G$, and V_m, W_m their orthonormal bases let $X_m := V_m Y_m W_m^T$, $X_m \approx X$

Galerkin condition: $R := AX_m + X_m G^T - B_1 B_2^T \quad \perp \quad \mathcal{K}_A \otimes \mathcal{K}_G$

$$V_m^\top R W_m = 0$$

Note: $\mathcal{K}_A$, $\mathcal{K}_G$ tiny wrto $\mathbb{K}(\mathcal{A}, b)$

Projected Sylvester equation:

$$\begin{aligned} V_m^\top (AV_m Y_m W_m^\top + V_m Y_m W_m^\top G^\top &- B_1 B_2^\top) W_m = 0 \\ (V_m^\top A V_m) Y_m + Y_m (V_m^\top G^\top V_m) &- V_m^\top B_1 B_2^\top W_m = 0 \end{aligned}$$

Early contributions: Saad '90, Jaimoukha & Kasenally '94, for $\mathrm{range}(V_m) = \mathcal{K}_A = \mathrm{Range}([B_1, AB_1, \ldots, A^{m-1} B_1])$

Projection-type methods

Assume $B = B_1 B_2^T$.
Given two low dimensional approx spaces $\mathcal{K}_A$, $\mathcal{K}_G$, and V_m, W_m their orthonormal bases let $X_m := V_m Y_m W_m^T$, $X_m \approx X$

Galerkin condition: $R := AX_m + X_m G^T - B_1 B_2^T \quad \perp \quad \mathcal{K}_A \otimes \mathcal{K}_G$

$$V_m^\top R W_m = 0$$

Note: $\mathcal{K}_A$, $\mathcal{K}_G$ tiny wrto $\mathbb{K}(\mathcal{A}, b)$

Projected Sylvester equation:

$$\begin{aligned} V_m^\top (AV_m Y_m W_m^\top + V_m Y_m W_m^\top G^\top \quad &- \quad B_1 B_2^\top) W_m = 0 \\ (V_m^\top A V_m) Y_m + Y_m (V_m^\top G^\top V_m) \quad &- \quad V_m^\top B_1 B_2^\top W_m = 0 \end{aligned}$$

Early contributions: Saad '90, Jaimoukha & Kasenally '94, for
$\text{range}(V_m) = \mathcal{K}_A = \text{Range}([B_1, AB_1, \ldots, A^{m-1} B_1])$

More recent options as approximation space

Enrich space to decrease space dimension

- Extended Krylov subspace

  (Druskin & Knizhnerman '98, Simoncini '07)

- Rational Krylov subspace

$$\mathcal{K}_A = \mathbb{K}_A := \mathrm{Range}([B_1, (A - s_1 I)^{-1} B_1, \ldots, \prod_{j=1}^{m-1} (A - s_j I)^{-1} B_1])$$

  usually, $\{s_1, \ldots, s_{m-1}\} \subset \mathbb{C}^+$ chosen either a-priori or dynamically
  (form matrix equations, Druskin & Simoncini '11)

In both cases, for $\mathrm{Range}(V_m) = \mathcal{K}_A$, $\mathrm{Range}(W_m) = \mathcal{K}_{G^T}$ projected Lyapunov equation:

$$(V_m^T A V_m) Y_m + Y_m (W_m^T G^T W_m) - V_m^T B_1 B_2^T W_m = 0 \qquad X_m = V_m Y_m W_m^T$$

More recent options as approximation space

Enrich space to decrease space dimension

- Extended Krylov subspace

  (Druskin & Knizhnerman '98, Simoncini '07)

- Rational Krylov subspace

$$\mathcal{K}_A = \mathbb{K}_A := \mathrm{Range}([B_1, (A - s_1 I)^{-1} B_1, \ldots, \prod_{j=1}^{m-1} (A - s_j I)^{-1} B_1])$$

  usually, $\{s_1, \ldots, s_{m-1}\} \subset \mathbb{C}^+$ chosen either a-priori or dynamically
  (form matrix equations, Druskin & Simoncini '11)

In both cases, for $\mathrm{Range}(V_m) = \mathcal{K}_A$, $\mathrm{Range}(W_m) = \mathcal{K}_{G^T}$ projected Lyapunov equation:

$$(V_m^\top A V_m) Y_m + Y_m (W_m^\top G^\top W_m) - V_m^\top B_1 B_2^\top W_m = 0 \qquad X_m = V_m Y_m W_m^\top$$

More recent options as approximation space

Enrich space to decrease space dimension

- Extended Krylov subspace

  (Druskin & Knizhnerman '98, Simoncini '07)

- Rational Krylov subspace

$$\mathcal{K}_A = \mathbb{K}_A := \mathrm{Range}([B_1, (A - s_1 I)^{-1} B_1, \ldots, \prod_{j=1}^{m-1} (A - s_j I)^{-1} B_1])$$

  usually, $\{s_1, \ldots, s_{m-1}\} \subset \mathbb{C}^+$ chosen either a-priori or dynamically
  (form matrix equations, Druskin & Simoncini '11)

In both cases, for $\mathrm{Range}(V_m) = \mathcal{K}_A$, $\mathrm{Range}(W_m) = \mathcal{K}_{G^T}$ projected Lyapunov equation:

$$(V_m^\top A V_m) Y_m + Y_m (W_m^\top G^\top W_m) - V_m^\top B_1 B_2^\top W_m = 0 \qquad X_m = V_m Y_m W_m^\top$$

Multiterm linear matrix equation

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C \tag{1}$$

$A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{m \times m}$, $\boldsymbol{X}$ unknown matrix

Possibly large dimensions, structured coefficient matrices

> *The problem in its full generality is far from tractable, although the transformation to a matrix-vector equation [...] allows us to use the considerable arsenal of numerical weapons currently available for the solution of such problems.*

Peter Lancaster, SIAM Rev. 1970

> *Another generalization of the Sylvester equation, mainly of theoretical interest, is (1)*

Nick Higham, in "Accuracy and Stability of Numerical Algorithms", SIAM, 1996

Multiterm linear matrix equation

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C \tag{1}$$

$A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{m \times m}$, $\boldsymbol{X}$ unknown matrix

Possibly large dimensions, structured coefficient matrices

> *The problem in its full generality is far from tractable, although the transformation to a matrix-vector equation [...] allows us to use the considerable arsenal of numerical weapons currently available for the solution of such problems.*

Peter Lancaster, SIAM Rev. 1970

> *Another generalization of the Sylvester equation, mainly of theoretical interest, is (1)*

Nick Higham, in "Accuracy and Stability of Numerical Algorithms", SIAM, 1996

Multiterm linear matrix equation

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

- Kronecker form and back on track
- Fixed point iterations (an "evergreen"...)
- Projection-type methods $\Rightarrow$ low rank approximation
- Ad-hoc problem-dependent procedures
- etc.

A sample of these methodologies on different problems:

- Stochastic PDEs
- PDEs on polygonal domains, IGA, spectral methods, etc
- Space-time PDEs
- All-at-once PDE-constrained optimization problem
- Bilinear control problems
- ....

Multiterm linear matrix equation

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

- Kronecker form and back on track
- Fixed point iterations (an "evergreen"...)
- Projection-type methods $\Rightarrow$ low rank approximation
- Ad-hoc problem-dependent procedures
- etc.

A sample of these methodologies on different problems:

- Stochastic PDEs
- PDEs on polygonal domains, IGA, spectral methods, etc
- Space-time PDEs
- All-at-once PDE-constrained optimization problem
- Bilinear control problems
- ....

Multiterm linear matrix equation. Classical device

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Kronecker formulation

$$\left(B_1^\top \otimes A_1 + \ldots + B_\ell^\top \otimes A_\ell\right) \boldsymbol{x} = c \qquad \Leftrightarrow \qquad \mathcal{A}\boldsymbol{x} = c$$

Iterative methods: matrix-matrix multiplications and rank truncation

(Benner, Bioli, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Kuerschner, Matthies, Nagy, Onwunta, Palitta, Raydan, Robol, Stoll, Tobler, Wedderburn, Zander, ...)

Current very active area of research

Assume $\mathcal{A}$ is sym pos.def. (spd) $\Rightarrow$ CG

Multiterm linear matrix equation. Classical device

$$A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell = C$$

Kronecker formulation

$$\left(B_1^\top \otimes A_1 + \ldots + B_\ell^\top \otimes A_\ell\right) \boldsymbol{x} = c \qquad \Leftrightarrow \qquad \mathcal{A}\boldsymbol{x} = c$$

Iterative methods: matrix-matrix multiplications and rank truncation

(Benner, Bioli, Breiten, Bouhamidi, Chehab, Damm, Grasedyck, Jbilou, Kressner, Kuerschner, Matthies, Nagy, Onwunta, Palitta, Raydan, Robol, Stoll, Tobler, Wedderburn, Zander, ...)

Current very active area of research

Assume $\mathcal{A}$ is sym pos.def. (spd) $\Rightarrow$ CG

CG matricization and truncation

⋆ **Matricization.** Typically,

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \in \mathbb{R}^{n^2} \qquad \Rightarrow \quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)} \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X^{(k)} = X_1^{(k)} (X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)} (X_1^{(k)})^\top + \alpha_k P_1^{(k)} (P_1^{(k)})^\top$$

- $X^{(k+1)}$ low rank:

$$X^{(k+1)} = [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \, [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]^\top \tag{2}$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \Rightarrow X_1^{(k+1)} \qquad X^{(k+1)} \approx X_1^{(k+1)} (X_1^{(k+1)})^\top$

  Implementation: $\mathcal{T}(X^{(k+1)})$ acts on the QR-SVD of factor in (2)

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

⋆ **Matricization.** Typically,

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \in \mathbb{R}^{n^2} \qquad \Rightarrow \quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)} \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X^{(k)} = X_1^{(k)} (X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)} (X_1^{(k)})^\top + \alpha_k P_1^{(k)} (P_1^{(k)})^\top$$

- $X^{(k+1)}$ low rank:

$$X^{(k+1)} = [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \, [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]^\top \tag{2}$$

(but generally larger than at iteration k)

- Cure: Rank shrinking $[X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \Rightarrow X_1^{(k+1)} \qquad X^{(k+1)} \approx X_1^{(k+1)} (X_1^{(k+1)})^\top$

Implementation: $\mathcal{T}(X^{(k+1)})$ acts on the QR-SVD of factor in (2)

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

⋆ **Matricization.** Typically,

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \in \mathbb{R}^{n^2} \qquad \Rightarrow \quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)} \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X^{(k)} = X_1^{(k)} (X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)} (X_1^{(k)})^\top + \alpha_k P_1^{(k)} (P_1^{(k)})^\top$$

- $X^{(k+1)}$ low rank:

$$X^{(k+1)} = [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \, [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]^\top \tag{2}$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \Rightarrow X_1^{(k+1)} \qquad X^{(k+1)} \approx X_1^{(k+1)} (X_1^{(k+1)})^\top$

  Implementation: $\mathcal{T}(X^{(k+1)})$ acts on the QR-SVD of factor in (2)

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

⋆ **Matricization.** Typically,

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \in \mathbb{R}^{n^2} \qquad \Rightarrow \quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)} \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X^{(k)} = X_1^{(k)}(X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)}(X_1^{(k)})^\top + \alpha_k P_1^{(k)}(P_1^{(k)})^\top$$

- $X^{(k+1)}$ low rank:

$$X^{(k+1)} = [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]\, [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]^\top \tag{2}$$

(but generally larger than at iteration k)

- Cure: Rank shrinking $[X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \Rightarrow X_1^{(k+1)}$ $\quad X^{(k+1)} \approx X_1^{(k+1)}(X_1^{(k+1)})^\top$

Implementation: $\mathcal{T}(X^{(k+1)})$ acts on the QR-SVD of factor in (2)

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

⋆ **Matricization.** Typically,

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} \in \mathbb{R}^{n^2} \qquad \Rightarrow \quad X^{(k+1)} = X^{(k)} + \alpha_k P^{(k)} \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X^{(k)} = X_1^{(k)}(X_1^{(k)})^\top$ with $X_1^{(k)}$ low rank, and similarly for $P^{(k)}$, then

$$X^{(k+1)} = X_1^{(k)}(X_1^{(k)})^\top + \alpha_k P_1^{(k)}(P_1^{(k)})^\top$$

- $X^{(k+1)}$ low rank:

$$X^{(k+1)} = [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]\ [X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}]^\top \tag{2}$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_1^{(k)}, \sqrt{\alpha_k} P_1^{(k)}] \Rightarrow X_1^{(k+1)}$ $\quad X^{(k+1)} \approx X_1^{(k+1)}(X_1^{(k+1)})^\top$

  Implementation: $\mathcal{T}(X^{(k+1)})$ acts on the QR-SVD of factor in (2)

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

Truncated matrix-oriented CG (TCG) for Kronecker form

Input: $\mathcal{L}(\boldsymbol{X}) = A_1 \boldsymbol{X} B_1 + A_2 \boldsymbol{X} B_2 + \ldots + A_\ell \boldsymbol{X} B_\ell$, right-hand side $C \in \mathbb{R}^{n \times n}$ in low-rank format. Truncation operator $\mathcal{T}$.
Output: Matrix $X \in \mathbb{R}^{n \times n}$ in low-rank format s.t. $||\mathcal{L}(X) - C||_F / ||C||_F \leq tol$

1. $X_0 = 0$, $R_0 = C$, $P_0 = R_0$, $Q_0 = \mathcal{L}(P_0)$
2. $\xi_0 = \langle P_0, Q_0 \rangle$, $k = 0$ $\qquad\qquad \langle X, Y \rangle = \mathrm{tr}(X^\top Y)$
3. While $||R_k||_F > tol$
4. $\quad \alpha_k = \langle R_k, P_k \rangle / \xi_k$
5. $\quad X_{k+1} = X_k + \alpha_k P_k,$ $\qquad X_{k+1} \leftarrow \mathcal{T}(X_{k+1})$
6. $\quad R_{k+1} = C - \mathcal{L}(X_{k+1}),$ $\qquad$ Optionally: $R_{k+1} \leftarrow \mathcal{T}(R_{k+1})$
7. $\quad \beta_k = -\langle R_{k+1}, Q_k \rangle / \xi_k$
8. $\quad P_{k+1} = R_{k+1} + \beta_k P_k,$ $\qquad P_{k+1} \leftarrow \mathcal{T}(P_{k+1})$
9. $\quad Q_{k+1} = \mathcal{L}(P_{k+1}),$ $\qquad$ Optionally: $Q_{k+1} \leftarrow \mathcal{T}(Q_{k+1})$
10. $\quad \xi_{k+1} = \langle P_{k+1}, Q_{k+1} \rangle$
11. $\quad k = k + 1$
12. end while

♣ Iterates kept in factored form! Kressner and Tobler, '11

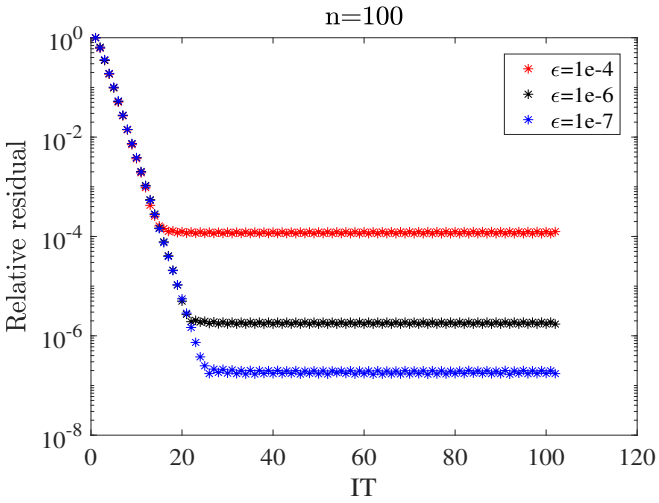
Truncated matrix-oriented CG (TCG) for Kronecker form

Input: $\mathcal{L}(\boldsymbol{X}) = A_1\boldsymbol{X}B_1 + A_2\boldsymbol{X}B_2 + \ldots + A_\ell\boldsymbol{X}B_\ell$, right-hand side $C \in \mathbb{R}^{n\times n}$ in low-rank format.
Truncation operator $\mathcal{T}$.
Output: Matrix $X \in \mathbb{R}^{n\times n}$ in low-rank format s.t. $||\mathcal{L}(X) - C||_F/||C||_F \leq tol$

1. $X_0 = 0$, $R_0 = C$, $P_0 = R_0$, $Q_0 = \mathcal{L}(P_0)$
2. $\xi_0 = \langle P_0, Q_0\rangle$, $k = 0$ $\qquad\qquad \langle X, Y\rangle = \mathrm{tr}(X^\top Y)$
3. While $||R_k||_F > tol$
4. $\quad \alpha_k = \langle R_k, P_k\rangle/\xi_k$
5. $\quad X_{k+1} = X_k + \alpha_k P_k$, $\qquad X_{k+1} \leftarrow \mathcal{T}(X_{k+1})$
6. $\quad R_{k+1} = C - \mathcal{L}(X_{k+1})$, $\qquad$ Optionally: $R_{k+1} \leftarrow \mathcal{T}(R_{k+1})$
7. $\quad \beta_k = -\langle R_{k+1}, Q_k\rangle/\xi_k$
8. $\quad P_{k+1} = R_{k+1} + \beta_k P_k$, $\qquad P_{k+1} \leftarrow \mathcal{T}(P_{k+1})$
9. $\quad Q_{k+1} = \mathcal{L}(P_{k+1})$, $\qquad$ Optionally: $Q_{k+1} \leftarrow \mathcal{T}(Q_{k+1})$
10. $\quad \xi_{k+1} = \langle P_{k+1}, Q_{k+1}\rangle$
11. $\quad k = k + 1$
12. end while

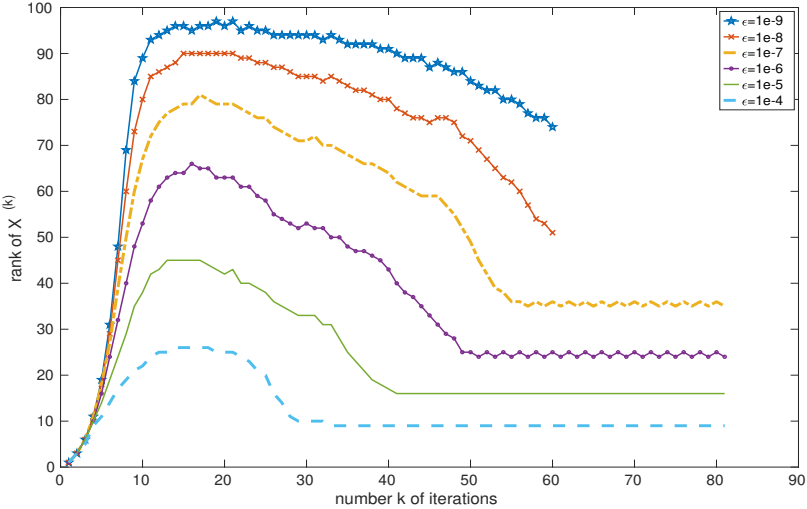
♣ Iterates kept in factored form! Kressner and Tobler, '11

Typical convergence behavior



(Hao, '20, personal comm.)

Typical iterate rank behavior



(Simoncini & Hao, '22)

Within the CG framework, can we do better?

Considerations:

1. At best, convergence as for Kronecker problem
2. Rank of iterates hard to control to maintain convergence
3. Coeffs α, β under exploited

$$\boldsymbol{p}_k = \mathrm{vec}(P_k), \boldsymbol{r}_k = \mathrm{vec}(R_k) \quad \Rightarrow \quad \{\boldsymbol{r}_0, \ldots, \boldsymbol{r}_k\}, \{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_k\} \quad \text{orth prop}$$

Recalling CG basics: $\mathcal{A}\boldsymbol{x} = \boldsymbol{b}$

Problem: Minimize the convex function

$$\Phi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \mathcal{A}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \qquad \alpha_k \, s.t. \, \min_{\alpha} \Phi(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)$$

with residual and direction updates:

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \mathcal{A}\boldsymbol{p}_k \alpha_k, \quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \boldsymbol{p}_k \beta_k.$$

Within the CG framework, can we do better?

Considerations:

1. At best, convergence as for Kronecker problem
2. Rank of iterates hard to control to maintain convergence
3. Coeffs α, β under exploited

$$\boldsymbol{p}_k = \mathrm{vec}(P_k), \boldsymbol{r}_k = \mathrm{vec}(R_k) \quad \Rightarrow \quad \{\boldsymbol{r}_0, \ldots, \boldsymbol{r}_k\}, \{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_k\} \quad \text{orth prop}$$

Recalling CG basics: $\mathcal{A}\boldsymbol{x} = \boldsymbol{b}$

Problem: Minimize the convex function

$$\Phi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T \mathcal{A}\boldsymbol{x} - \boldsymbol{b}^T \boldsymbol{x}$$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \qquad \alpha_k \; s.t. \; \min_{\alpha} \Phi(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k)$$

with residual and direction updates:

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \mathcal{A}\boldsymbol{p}_k \alpha_k, \quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \boldsymbol{p}_k \beta_k.$$

The new **subspace**-CG

Warning: For the sake of the presentation, we assume a simplified form:

$$\mathcal{L}(\boldsymbol{X}) = \mathcal{L}(\boldsymbol{X})^T, \qquad \text{for any} \quad \boldsymbol{X} = \boldsymbol{X}^T$$

This assumption allows us to write a square $\boldsymbol{X}$ as $\boldsymbol{X} = XX^T$

In practice, the whole derivation holds for $\mathcal{A} = B_1 \otimes A_1 + \cdots + B_\ell \otimes A_\ell$ spd
(that is, $\mathcal{L}$ spd in the matrix inner product)
so that

$$\boldsymbol{X} \in \mathbb{R}^{n_l \times n_r}, \qquad \boldsymbol{X} = X^l (X^r)^T$$

The new **subspace**-CG

Warning: For the sake of the presentation, we assume a simplified form:

$$\mathcal{L}(\boldsymbol{X}) = \mathcal{L}(\boldsymbol{X})^T, \qquad \text{for any} \quad \boldsymbol{X} = \boldsymbol{X}^T$$

This assumption allows us to write a square $\boldsymbol{X}$ as $\boldsymbol{X} = XX^T$

In practice, the whole derivation holds for $\mathcal{A} = B_1 \otimes A_1 + \cdots + B_\ell \otimes A_\ell$ spd
(that is, $\mathcal{L}$ spd in the matrix inner product)
so that

$$\boldsymbol{X} \in \mathbb{R}^{n_l \times n_r}, \qquad \boldsymbol{X} = X^l (X^r)^T$$

The new **subspace** CG

We define $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}$,

$$\Phi(\boldsymbol{X}) = \frac{1}{2}\langle \boldsymbol{X}, \mathcal{L}(\boldsymbol{X})\rangle - \langle \boldsymbol{X}, \boldsymbol{C}\rangle$$

The new minimization problem: *Find* $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ *such that*

$$\boldsymbol{X} = \arg \min_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} \Phi(\boldsymbol{X})$$

with iteration

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P_k \boldsymbol{\alpha}_k P_k^T$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^{s_k\times s_k}$ and $P_k \in \mathbb{R}^{n\times s_k}$

Residual and direction computation:

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k - \mathcal{L}(P_k \boldsymbol{\alpha}_k P_k^T), \qquad \boldsymbol{P}_{k+1} = \boldsymbol{R}_{k+1} + P_k \boldsymbol{\beta}_k P_k^T,$$

where $\boldsymbol{P}_{k+1} = P_{k+1} P_{k+1}^T$

The new **subspace** CG

We define $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}$,

$$\Phi(\boldsymbol{X}) = \frac{1}{2}\langle \boldsymbol{X}, \mathcal{L}(\boldsymbol{X})\rangle - \langle \boldsymbol{X}, \boldsymbol{C}\rangle$$

The new minimization problem: *Find* $\boldsymbol{X} \in \mathbb{R}^{n\times n}$ *such that*

$$\boldsymbol{X} = \arg \min_{\boldsymbol{X}\in\mathbb{R}^{n\times n}} \Phi(\boldsymbol{X})$$

with iteration

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P_k \boldsymbol{\alpha}_k P_k^T$$

where $\boldsymbol{\alpha}_k \in \mathbb{R}^{s_k\times s_k}$ and $P_k \in \mathbb{R}^{n\times s_k}$

Residual and direction computation:

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k - \mathcal{L}(P_k \boldsymbol{\alpha}_k P_k^T), \qquad \boldsymbol{P}_{k+1} = \boldsymbol{R}_{k+1} + P_k \boldsymbol{\beta}_k P_k^T,$$

where $\boldsymbol{P}_{k+1} = P_{k+1} P_{k+1}^T$

Defining the coefficients. I

At the kth iteration:

1. Construct $\boldsymbol{\alpha}_k$ so that

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{s_k \times s_k}} \Phi(\boldsymbol{X}_k + P_k \boldsymbol{\alpha} P_k^T)$$

2. Impose a descent direction requirement for $\boldsymbol{P}_k = P_k P_k^T$:

$$\langle \nabla \Phi(\boldsymbol{X}_k), \boldsymbol{P}_k \rangle < 0$$

3. Construct β_k so that the new direction $\boldsymbol{P}_{k+1}$ is $\mathcal{L}$-orthogonal with respect to the previous ones:

$$(P_k \otimes P_k)^T \mathrm{vec}(\mathcal{L}(\boldsymbol{P}_{k+1})) = 0$$

Defining the coefficients. II

Let $\widetilde{A}_k^{(i)} = P_k^T A_i P_k, \widetilde{B}_k^{(i)} = P_k^T B_i P_k$, $i = 1, \ldots, \ell$

- Construction of $\boldsymbol{\alpha}_k$:
  $\boldsymbol{\alpha}_k$ is the unique solution of

$$\widetilde{A}_k^{(1)} \boldsymbol{\alpha} \widetilde{B}_k^{(1)} + \ldots + \widetilde{A}_k^{(\ell)} \boldsymbol{\alpha} \widetilde{B}_k^{(\ell)} = P_k^T \boldsymbol{R}_k P_k$$

- Construction of $\boldsymbol{\beta}_k$:
  $\boldsymbol{\beta}_k$ is the unique solution of

$$\widetilde{A}_k^{(1)} \boldsymbol{\beta} \widetilde{B}_k^{(1)} + \ldots + \widetilde{A}_k^{(\ell)} \boldsymbol{\beta} \widetilde{B}_k^{(\ell)} = -P_k^T \mathcal{L}(P_k \boldsymbol{\beta}_k P_k^T) P_k$$

Defining the coefficients. II

Let $\widetilde{A}_k^{(i)} = P_k^T A_i P_k, \widetilde{B}_k^{(i)} = P_k^T B_i P_k,\ i = 1, \ldots, \ell$

- Construction of $\boldsymbol{\alpha}_k$:
  $\boldsymbol{\alpha}_k$ is the unique solution of

$$\widetilde{A}_k^{(1)} \boldsymbol{\alpha} \widetilde{B}_k^{(1)} + \ldots + \widetilde{A}_k^{(\ell)} \boldsymbol{\alpha} \widetilde{B}_k^{(\ell)} = P_k^T \boldsymbol{R}_k P_k$$

- Construction of $\boldsymbol{\beta}_k$:
  $\boldsymbol{\beta}_k$ is the unique solution of

$$\widetilde{A}_k^{(1)} \boldsymbol{\beta} \widetilde{B}_k^{(1)} + \ldots + \widetilde{A}_k^{(\ell)} \boldsymbol{\beta} \widetilde{B}_k^{(\ell)} = -P_k^T \mathcal{L}(P_k \boldsymbol{\beta}_k P_k^T) P_k$$

Making the idea practical

$$\begin{aligned} \boldsymbol{X}_{k+1} &= X_k \boldsymbol{\tau}_k X_k^T + P_k \boldsymbol{\alpha}_k P_k^T = [X_k, P_k] \boldsymbol{\tau}_{k+1} [X_k, P_k]^T \\ \boldsymbol{R}_{k+1} &= [R_0, \boldsymbol{A}_\star \bullet X_{k+1}] \boldsymbol{\rho}_{k+1} [R_0, \boldsymbol{B}_\star \bullet X_{k+1}]^T \end{aligned}$$

where $\boldsymbol{A}_\star \bullet R = [A_1 R, \ldots, A_\ell R]$

- All terms are kept in factored form
- The rank grows

Rank truncation

- Computing $\boldsymbol{R}_{k+1}$ becomes too expensive (CPU time and memory)

Randomized range finder

Given a target rank `maxrankR` and a Gaussian matrix $G^l \in \mathbb{R}^{n_B \times \texttt{maxrankR}}$:

$$\boldsymbol{R}_{k+1} G^l = C_1 (C_2^T G^l) - \sum_{i=1}^{\ell} A_i (X^l_{k+1} \boldsymbol{\tau}_{k+1} ((X^r_{k+1})^T (B_i G^l)))$$

(Analogously for $\boldsymbol{R}_{k+1}^T$)
Then proceed with a cheap evaluation of the reduced residual matrix

Making the idea practical

$$\begin{aligned} \boldsymbol{X}_{k+1} &= X_k \boldsymbol{\tau}_k X_k^T + P_k \boldsymbol{\alpha}_k P_k^T = [X_k, P_k] \boldsymbol{\tau}_{k+1} [X_k, P_k]^T \\ \boldsymbol{R}_{k+1} &= [R_0, \boldsymbol{A}_\star \bullet X_{k+1}] \boldsymbol{\rho}_{k+1} [R_0, \boldsymbol{B}_\star \bullet X_{k+1}]^T \end{aligned}$$

where $\boldsymbol{A}_\star \bullet R = [A_1 R, \ldots, A_\ell R]$

- All terms are kept in factored form
- The rank grows

Rank truncation

- Computing $\boldsymbol{R}_{k+1}$ becomes too expensive (CPU time and memory)

Randomized range finder

Given a target rank `maxrankR` and a Gaussian matrix $G^l \in \mathbb{R}^{n_B \times \texttt{maxrankR}}$:

$$\boldsymbol{R}_{k+1} G^l = C_1 (C_2^T G^l) - \sum_{i=1}^{\ell} A_i (X^l_{k+1} \boldsymbol{\tau}_{k+1} ((X^r_{k+1})^T (B_i G^l)))$$

(Analogously for $\boldsymbol{R}_{k+1}^T$)
Then proceed with a cheap evaluation of the reduced residual matrix

Making the idea practical

$$\begin{aligned} \boldsymbol{X}_{k+1} &= X_k \boldsymbol{\tau}_k X_k^T + P_k \boldsymbol{\alpha}_k P_k^T = [X_k, P_k] \boldsymbol{\tau}_{k+1} [X_k, P_k]^T \\ \boldsymbol{R}_{k+1} &= [R_0, \boldsymbol{A}_\star \bullet X_{k+1}] \boldsymbol{\rho}_{k+1} [R_0, \boldsymbol{B}_\star \bullet X_{k+1}]^T \end{aligned}$$

where $\boldsymbol{A}_\star \bullet R = [A_1 R, \ldots, A_\ell R]$

- All terms are kept in factored form
- The rank grows

Rank truncation

- Computing $\boldsymbol{R}_{k+1}$ becomes too expensive (CPU time and memory)

Randomized range finder

Given a target rank `maxrankR` and a Gaussian matrix $G^l \in \mathbb{R}^{n_B \times \texttt{maxrankR}}$:

$$\boldsymbol{R}_{k+1} G^l = C_1 (C_2^T G^l) - \sum_{i=1}^{\ell} A_i (X^l_{k+1} \boldsymbol{\tau}_{k+1} ((X^r_{k+1})^T (B_i G^l)))$$

(Analogously for $\boldsymbol{R}_{k+1}^T$)
Then proceed with a cheap evaluation of the reduced residual matrix

A computational experiment. I

Parameterized diffusion equation (Biolietal, tech.rep.2024)

$$-\nabla \cdot (k \nabla u) = 0 \qquad \text{in} \quad (0,1)^2$$

with homogeneous boundary conditions and semi-separable diffusion coefficient:

$$k(x,y) = \sum_{j=1}^{\ell_k} \delta_j k_{j,x}(x) k_{j,y}(y) = 1 + \sum_{j=1}^{\ell_k - 1} \frac{10^j}{j!} x^j y^j, \qquad \ell_k = 4$$

This gives

$$\sum_{j=1}^{\ell_k} \delta_j (A_{j,x} \boldsymbol{X} D_{j,y} + D_{j,y} \boldsymbol{X} A_{j,y}) = \boldsymbol{C}$$

with $\boldsymbol{C}$ rank-four nonsym matrix accounting for b.c. (total of $\ell = 8$ terms)

Algorithms to be compared:
* SS-CG-determ: new method, residual matrix computed sequentially;
* SS-CG-rand'zed: new method, residual matrix computed using Randfinder
* TPCG: truncated matrix-oriented preconditioned CG (Kressner, Tobler, 2011, and others)
* R-NLTCG: Riemannian, nonlinear CG (Bioli, Kressner, Robol, tech.rep.2024)

A computational experiment. I

Parameterized diffusion equation (Biolietal, tech.rep.2024)

$$-\nabla \cdot (k \nabla u) = 0 \qquad \text{in} \quad (0,1)^2$$

with homogeneous boundary conditions and semi-separable diffusion coefficient:

$$k(x,y) = \sum_{j=1}^{\ell_k} \delta_j k_{j,x}(x) k_{j,y}(y) = 1 + \sum_{j=1}^{\ell_k - 1} \frac{10^j}{j!} x^j y^j, \qquad \ell_k = 4$$

This gives

$$\sum_{j=1}^{\ell_k} \delta_j (A_{j,x} \boldsymbol{X} D_{j,y} + D_{j,y} \boldsymbol{X} A_{j,y}) = \boldsymbol{C}$$

with $\boldsymbol{C}$ rank-four nonsym matrix accounting for b.c. (total of $\ell = 8$ terms)

Algorithms to be compared:
* SS-CG-determ: new method, residual matrix computed sequentially;
* SS-CG-rand'zed: new method, residual matrix computed using Randfinder
* TPCG: truncated matrix-oriented preconditioned CG (Kressner, Tobler, 2011, and others)
* R-NLTCG: Riemannian, nonlinear CG (Bioli, Kressner, Robol, tech.rep.2024)

A computational experiment. II

n type	Precond	maxrank	R-NLCG	TPCG	SS–CG determ.	SS–CG rand'zed
10000	$\mathcal{P}_1$	20	– (100)	– (100)	– (100)	– (100)
	$\mathcal{P}_1$	40	– (100)	– (100)	1.08 (5)	**0.92** (5)
	$\mathcal{P}_1$	60	– (100)	– (100)	2.47 (5)	**2.34** (5)
	$\mathcal{P}_2$	20	**11.25 (36)**	11.42 (38)	– (100)	– (100)
	$\mathcal{P}_2$	40	*42.97 (36)	**15.54** (33)	– (100)	– (100)
	$\mathcal{P}_2$	60	*98.62 (35)	32.39 (28)	9.59 (5)	**8.37** (5)
102400	$\mathcal{P}_1$	20	– (100)	– (100)	– (100)	– (100)
	$\mathcal{P}_1$	40	†	– (100)	18.17 (6)	**8.74** (6)
	$\mathcal{P}_1$	60	†	– (100)	23.50 (5)	**16.93** (5)
	$\mathcal{P}_2$	20	**183.44** (41)	– (100)	– (100)	– (100)
	$\mathcal{P}_2$	40	†	446.94 (47)	– (100)	– (100)
	$\mathcal{P}_2$	60	†	884.20 (26)	115.73 (3)	**101.91** (3)

– no conv. * Lower final residual norm than other methods † Out of Memory

Running time in seconds (# iter's)
Stopping tolerance $\texttt{tol} = 5 \cdot 10^{-6}$ True residual norm at termination

$\mathcal{P}_1$: one-term precond, cheap $\mathcal{P}_2$: two-term precond, expensive (fixed ADI iters)

What is left

What I have not told you about:

- Orthogonality properties of residuals and directions
- Optimality and finite termination properties
- Preconditioning
- More experiments on a variety of application problems

Outlook:

- Experiments are very promising
- The idea can be generalized to other Krylov methods
- Tensor version under investigation

Reference

Davide Palitta, Martina Iannacito, and V. Simoncini
A subspace-conjugate gradient method for linear matrix equations
pp. 1-25, Jan 2025. ArXiv 2501.02938

What is left

What I have not told you about:

- Orthogonality properties of residuals and directions
- Optimality and finite termination properties
- Preconditioning
- More experiments on a variety of application problems

Outlook:

- Experiments are very promising
- The idea can be generalized to other Krylov methods
- Tensor version under investigation

Reference

Davide Palitta, Martina Iannacito, and V. Simoncini

A subspace-conjugate gradient method for linear matrix equations

pp. 1-25, Jan 2025. ArXiv 2501.02938

What is left

What I have not told you about:

- Orthogonality properties of residuals and directions
- Optimality and finite termination properties
- Preconditioning
- More experiments on a variety of application problems

Outlook:

- Experiments are very promising
- The idea can be generalized to other Krylov methods
- Tensor version under investigation

Reference

Davide Palitta, Martina Iannacito, and V. Simoncini
A subspace-conjugate gradient method for linear matrix equations
pp. 1-25, Jan 2025. ArXiv 2501.02938