Randomization strategies for low-rank matrix equation solvers

Valeria Simoncini

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

Joint works with Martina Iannacito and Davide Palitta (UniBO),
and Catherine Powell (UniManchester)

Multiterm linear matrix equation

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

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

A sample of this equation on different problems:

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

A sample of computational strategies:

- Kronecker form and back on track
- Fixed point iterations (an "evergreen"...)
- ALS (better for tensor eqn)
- Projection-type methods $\Rightarrow$ low rank approximation
- Optimization problems with fixed (low) rank approximation
- Ad-hoc problem-dependent procedures
- etc.

Multiterm linear matrix equation

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

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

A sample of this equation on different problems:

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

A sample of computational strategies:

- Kronecker form and back on track
- Fixed point iterations (an "evergreen"...)
- ALS (better for tensor eqn)
- Projection-type methods $\Rightarrow$ low rank approximation
- Optimization problems with fixed (low) rank approximation
- Ad-hoc problem-dependent procedures
- etc.

Multiterm linear matrix equation

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

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

A sample of this equation on different problems:

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

A sample of computational strategies:

- Kronecker form and back on track
- Fixed point iterations (an "evergreen"...)
- ALS (better for tensor eqn)
- Projection-type methods ⇒ low rank approximation
- Optimization problems with fixed (low) rank approximation
- Ad-hoc problem-dependent procedures
- etc.

Randomized NLA as a key Tool. 1

A general problem:

Given

$$\mathcal{Z} = [Z_1, \ldots, Z_m] \in \mathbb{R}^{n_1 \times km}$$

with $n_1 \gg km$ and $km \gg 1$ that

- cannot be stored all
- is not all available (only one block at the time)

Encountered algebraic tasks:

- Construct (thinner) approximation to row and column bases of $(\mathcal{Z}\mathcal{Y}^T)$
  Key: the true $\mathcal{Z}, \mathcal{Y}$ need **not** be retained
  (here $\mathcal{Y} \in \mathbb{R}^{n_2 \times km}$ similar to $\mathcal{Z}$)
- Construct well-conditioned basis for range($\mathcal{Z}$), range($\mathcal{Y}$)
- Estimate $\|\mathcal{Z}\mathcal{Y}^T\|_F$

Randomized NLA as a key Tool. 1

A general problem:

Given

$$\mathcal{Z} = [Z_1, \ldots, Z_m] \in \mathbb{R}^{n_1 \times km}$$

with $n_1 \gg km$ and $km \gg 1$ that

- cannot be stored all
- is not all available (only one block at the time)

Encountered algebraic tasks:

- Construct (thinner) approximation to row and column bases of $(\mathcal{Z}\mathcal{Y}^T)$
  Key: the true $\mathcal{Z}, \mathcal{Y}$ need **not** be retained
  (here $\mathcal{Y} \in \mathbb{R}^{n_2 \times km}$ similar to $\mathcal{Z}$)
- Construct well-conditioned basis for range($\mathcal{Z}$), range($\mathcal{Y}$)
- Estimate $\|\mathcal{Z}\mathcal{Y}^T\|_F$

Randomized NLA as a key Tool. 2

Construct (thinner) approximation to row and column bases of $(\mathcal{Z}\mathcal{Y}^T)$:

Consider *Gaussian* sketching matrices $G_Z \in RR^{mk\times r_1}$, $G_Y \in RR^{mk\times r_2}$, $r_i \ll n_i$, and compute

1. Left space: $W_1 := \mathcal{Z}\mathcal{Y}^T G_Z = \sum_{i=1}^m Z_i(Y_i^T G_Z)$ (sum sequence)
2. $W_1 = Q_1 R_1$ (reduced QR)
3. Right space: $W_2 := \mathcal{Y}\mathcal{Z}^T G_Y = \sum_{i=1}^m Y_i(Z_i^T G_Y)$ (sum sequence)
4. $W_2 = Q_2 R_2$ (reduced QR)
5. Compute $\boldsymbol{M} = Q_1^T \mathcal{Z}\mathcal{Y}^T Q_2 = \sum_{i=1}^m (Q_1^T Z_i)(Y_i^T Q_2)$
6. Truncated SVD: $U\Sigma V^T = \boldsymbol{M}$
7. Deliver: $\widetilde{\mathcal{Z}} = Q_1 U \Sigma^{1/2}$, $\widetilde{\mathcal{Y}} = Q_2 V \Sigma^{1/2}$

$\mathcal{Z} \mapsto \widetilde{\mathcal{Z}}, \qquad \mathcal{Y} \mapsto \widetilde{\mathcal{Y}}$

Halko, Martinsson, Tropp, 2011

Randomized NLA as a key Tool. 3

Construct well-conditioned bases for range($\mathcal{Z}$), range($\mathcal{Y}$)

Consider *oblivious subspace embedding* matrices[1] $S_1 \in \mathbb{R}^{r_1 \times n_1}$, $S_2 \in \mathbb{R}^{r_2 \times n_2}$

1. Compute reduced QR:

$$S_1\mathcal{Z} = [S_1 Z_1, \ldots, S_1 Z_k] = Q_1 R_1, \qquad S_2\mathcal{Y} = [S_2 Y_1, \ldots, S_2 Y_k] = Q_2 R_2$$

2. Compute truncated SVD: $U\Sigma V^T = R_1 R_2^T$
3. Write

$$\widehat{\mathcal{Z}} = \mathcal{Z} R_1^{-1} U \Sigma^{1/2}, \qquad \widehat{\mathcal{Y}} = \mathcal{Y} R_2^{-1} V \Sigma^{1/2}$$

- $\mathcal{Z}\mathcal{Y}^T \approx \widehat{\mathcal{Z}}\widehat{\mathcal{Y}}^T$
- $\mathrm{cond}(\widehat{\mathcal{Z}}\Sigma^{-1/2}), \mathrm{cond}(\widehat{\mathcal{Y}}\Sigma^{-1/2}) \le \frac{\sqrt{1+\varepsilon_i}}{\sqrt{1-\varepsilon_i}}$ (well-conditioned)

[1] Randomized subsampled trigonometric transformations, with $s_i = \mathcal{O}(\varepsilon_i^{-2} km/\delta_i)$, $i = 1, 2$, which works well in practice.

Randomized NLA as a key Tool. 3

Construct well-conditioned bases for range($\mathcal{Z}$), range($\mathcal{Y}$)

Consider *oblivious subspace embedding* matrices[1] $S_1 \in \mathbb{R}^{r_1 \times n_1}$, $S_2 \in \mathbb{R}^{r_2 \times n_2}$

1. Compute reduced QR:

$$S_1 \mathcal{Z} = [S_1 Z_1, \ldots, S_1 Z_k] = Q_1 R_1, \qquad S_2 \mathcal{Y} = [S_2 Y_1, \ldots, S_2 Y_k] = Q_2 R_2$$

2. Compute truncated SVD: $U \Sigma V^T = R_1 R_2^T$
3. Write

$$\widehat{\mathcal{Z}} = \mathcal{Z} R_1^{-1} U \Sigma^{1/2}, \qquad \widehat{\mathcal{Y}} = \mathcal{Y} R_2^{-1} V \Sigma^{1/2}$$

- $\mathcal{Z}\mathcal{Y}^T \approx \widehat{\mathcal{Z}}\widehat{\mathcal{Y}}^T$
- $\mathrm{cond}(\widehat{\mathcal{Z}}\Sigma^{-1/2}), \mathrm{cond}(\widehat{\mathcal{Y}}\Sigma^{-1/2}) \leq \frac{\sqrt{1+\varepsilon_i}}{\sqrt{1-\varepsilon_i}}$ (well-conditioned)

[1] Randomized subsampled trigonometric transformations, with $s_i = \mathcal{O}(\varepsilon_i^{-2} km/\delta_i)$, $i = 1, 2$, which works well in practice.

Randomized NLA as a key Tool. 4

Estimate $\|\mathcal{Z}\mathcal{Y}^T\|_F$

With the previous procedure employing oblivious subspace embeddings:

$$\|\mathcal{Z}\mathcal{Y}^T\|_F^2 \le (1-\varepsilon_1)^{-1}(1-\varepsilon_2)^{-1}\|S_1\mathcal{Z}\mathcal{Y}^T S_2\|_F^2$$

holding with probability at least $(1-\delta_1)(1-\delta_2)$. (see also M. Meier, 2024)

Since $\|S_1\mathcal{Z}\mathcal{Y}^T S_2\|_F = \|\Sigma\|_F$, it holds

$$\|\mathcal{Z}\mathcal{Y}^T\|_F \le (1-\varepsilon_1)^{-1/2}(1-\varepsilon_2)^{-1/2}\|\Sigma\|_F$$

Back to Multiterm linear matrix equation. Classical device

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

Kronecker formulation[2]

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

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

2

Kronecker product : $M \otimes P = \begin{bmatrix} m_{11}P & \ldots & m_{1n}P \\ \vdots & \ddots & \vdots \\ m_{n1}P & \ldots & m_{nn}P \end{bmatrix}$ and $\mathrm{vec}(AXB) = (B^\top \otimes A)\mathrm{vec}(X)$

CG matricization and truncation

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

$\star$ **Matricization.** Typically,

$$x_{k+1} = x_k + \alpha_k p_k \in \mathbb{R}^{n^2} \qquad \Rightarrow \qquad X_{k+1} = X_k + \alpha_k P_k \in \mathbb{R}^{n\times n}$$

$\star$ **Truncation.** If $X_k = X_k^{(l)}(X_k^{(r)})^\top$ low rank, and similarly for P_k, then

$$X_{k+1} = X_k^{(l)}(X_k^{(r)})^\top + \alpha_k P_k^{(l)}(P_k^{(r)})^\top$$

- $X^{(k+1)}$ low rank:
$$X_{k+1} = [X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}]\,[X_k^{(r)}, \sqrt{\alpha_k}P_k^{(r)}]^\top \qquad (1)$$
(but generally larger than at iteration k)
- Cure: Rank shrinking $[X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}] \Rightarrow X_{k+1}^{(l)}$ $\quad X_{k+1} \approx X_{k+1}^{(l)}(X_{k+1}^{(r)})^\top$

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

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

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

⋆ **Matricization.** Typically,

$$x_{k+1} = x_k + \alpha_k p_k \in \mathbb{R}^{n^2} \qquad \Rightarrow \qquad X_{k+1} = X_k + \alpha_k P_k \in \mathbb{R}^{n\times n}$$

⋆ **Truncation.** If $X_k = X_k^{(l)}(X_k^{(r)})^\top$ low rank, and similarly for P_k, then

$$X_{k+1} = X_k^{(l)}(X_k^{(r)})^\top + \alpha_k P_k^{(l)}(P_k^{(r)})^\top$$

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

$$X_{k+1} = [X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}]\,[X_k^{(r)}, \sqrt{\alpha_k}P_k^{(r)}]^\top \qquad (1)$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}] \Rightarrow X_{k+1}^{(l)}$ $\quad X_{k+1} \approx X_{k+1}^{(l)}(X_{k+1}^{(r)})^\top$

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

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

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

⋆ **Matricization.** Typically,

$$x_{k+1} = x_k + \alpha_k p_k \in \mathbb{R}^{n^2} \qquad \Rightarrow \qquad X_{k+1} = X_k + \alpha_k P_k \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X_k = X_k^{(l)}(X_k^{(r)})^\top$ low rank, and similarly for P_k, then

$$X_{k+1} = X_k^{(l)}(X_k^{(r)})^\top + \alpha_k P_k^{(l)}(P_k^{(r)})^\top$$

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

$$X_{k+1} = [X_k^{(l)}, \sqrt{\alpha_k} P_k^{(l)}]\ [X_k^{(r)}, \sqrt{\alpha_k} P_k^{(r)}]^\top \qquad (1)$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_k^{(l)}, \sqrt{\alpha_k} P_k^{(l)}] \Rightarrow X_{k+1}^{(l)}$ $\quad X_{k+1} \approx X_{k+1}^{(l)}(X_{k+1}^{(r)})^\top$

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

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

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

⋆ **Matricization.** Typically,

$$x_{k+1} = x_k + \alpha_k p_k \in \mathbb{R}^{n^2} \qquad \Rightarrow \qquad X_{k+1} = X_k + \alpha_k P_k \in \mathbb{R}^{n\times n}$$

⋆ **Truncation.** If $X_k = X_k^{(l)}(X_k^{(r)})^\top$ low rank, and similarly for P_k, then

$$X_{k+1} = X_k^{(l)}(X_k^{(r)})^\top + \alpha_k P_k^{(l)}(P_k^{(r)})^\top$$

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

$$X_{k+1} = [X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}]\,[X_k^{(r)}, \sqrt{\alpha_k}P_k^{(r)}]^\top \qquad (1)$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_k^{(l)}, \sqrt{\alpha_k}P_k^{(l)}] \Rightarrow X_{k+1}^{(l)}$ $\quad X_{k+1} \approx X_{k+1}^{(l)}(X_{k+1}^{(r)})^\top$

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

Alternative truncation criteria:

♣ Fix lower threshold tolerance ♣ Fix maximum allowed rank

CG matricization and truncation

Assume (for the moment) $\mathcal{A}$ sym pos.def. (spd) $\quad \Rightarrow \quad$ CG

⋆ **Matricization.** Typically,

$$x_{k+1} = x_k + \alpha_k p_k \in \mathbb{R}^{n^2} \qquad \Rightarrow \qquad X_{k+1} = X_k + \alpha_k P_k \in \mathbb{R}^{n \times n}$$

⋆ **Truncation.** If $X_k = X_k^{(l)}(X_k^{(r)})^\top$ low rank, and similarly for P_k, then

$$X_{k+1} = X_k^{(l)}(X_k^{(r)})^\top + \alpha_k P_k^{(l)}(P_k^{(r)})^\top$$

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

$$X_{k+1} = [X_k^{(l)}, \sqrt{\alpha_k} P_k^{(l)}]\ [X_k^{(r)}, \sqrt{\alpha_k} P_k^{(r)}]^\top \qquad (1)$$

  (but generally larger than at iteration k)

- Cure: Rank shrinking $[X_k^{(l)}, \sqrt{\alpha_k} P_k^{(l)}] \Rightarrow X_{k+1}^{(l)}$ $\quad X_{k+1} \approx X_{k+1}^{(l)}(X_{k+1}^{(r)})^\top$

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

Alternative truncation criteria:

♣ Fix lower threshold tolerance

♣ Fix maximum allowed rank

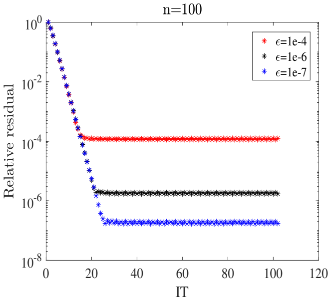
Truncated matrix-oriented CG (TCG) for Kronecker form

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. $||\mathcal{L}(X) - CD^T||_F/||CD^T||_F \leq tol$

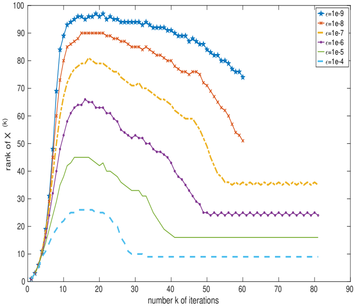
1. $X_0 = 0,\ R_0 = CD^T,\ P_0 = R_0,\ Q_0 = \mathcal{L}(P_0)$
2. $\xi_0 = \langle P_0, Q_0\rangle,\ k = 0$ $\qquad\qquad \langle X, Y\rangle = \mathrm{tr}(X^\top Y)$
3. While $||R_k||_F > tol$
4. $\quad \alpha_k = \langle R_k, P_k\rangle/\xi_k$
5. $\quad X_{k+1} = X_k + \alpha_k P_k,$ $\qquad X_{k+1} \leftarrow \mathcal{T}(X_{k+1})$
6. $\quad R_{k+1} = CD^T - \mathcal{L}(X_{k+1}),$ $\qquad$ 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! $\qquad$ Kressner and Tobler, '11

Typical convergence and rank behavior



(Hao, '20, personal comm.)



(Simoncini & Hao, '22)

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

Back to the roots

Consider

$$\mathcal{A}\boldsymbol{x} = c \qquad \mathcal{A} \quad \text{nonsing, nonsym}$$

One-dimensional projection method: Saad, "Iterative methods for sparse linear systems", SIAM, 2003

$$x_{k+1} = x_k + \alpha_k r_k, \quad r_{k+1} = b - \mathcal{A}x_{k+1}$$

For instance, $\alpha_k := \frac{(\mathcal{A}r_k)^T r_k}{(\mathcal{A}r_k)^T \mathcal{A}r_k}$ minimizes $\|c - \mathcal{A}(x_k + \alpha r_k)\|^2$

Consider $A_1 \boldsymbol{X} B_1 + \ldots + A_\ell \boldsymbol{X} B_\ell = CD^T$ with

$$\mathcal{L}(V) := A_1 V B_1 + \ldots + A_\ell V B_\ell$$

"One-dimensional" subspace projection method:

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha}_k (R_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{q_k \times q_k} \\ \boldsymbol{R}_{k+1} &= CD^T - \mathcal{L}(\boldsymbol{X}_{k+1}), \qquad \boldsymbol{R}_{k+1} =: R_{k+1}^{(l)} (R_{k+1}^{(r)})^T, \end{aligned}$$

At step k, $\boldsymbol{\alpha}_k$ is chosen to minimize the Frobenius norm of the residual, namely

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{q_k \times q_k}} \|CD^T - \mathcal{L}(\boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha} (R_k^{(r)})^T)\|_F^2$$

Back to the roots

Consider

$$\mathcal{A}\boldsymbol{x} = c \qquad \mathcal{A} \quad \text{nonsing, nonsym}$$

One-dimensional projection method: Saad, "Iterative methods for sparse linear systems", SIAM, 2003

$$x_{k+1} = x_k + \alpha_k r_k, \quad r_{k+1} = b - \mathcal{A}x_{k+1}$$

For instance, $\alpha_k := \frac{(\mathcal{A}r_k)^T r_k}{(\mathcal{A}r_k)^T \mathcal{A}r_k}$ minimizes $\|c - \mathcal{A}(x_k + \alpha r_k)\|^2$

Consider $A_1 \boldsymbol{X} B_1 + \ldots + A_\ell \boldsymbol{X} B_\ell = CD^T$ with

$$\mathcal{L}(V) := A_1 V B_1 + \ldots + A_\ell V B_\ell$$

"One-dimensional" subspace projection method:

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha}_k (R_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{q_k \times q_k} \\ \boldsymbol{R}_{k+1} &= CD^T - \mathcal{L}(\boldsymbol{X}_{k+1}), \qquad \boldsymbol{R}_{k+1} =: R_{k+1}^{(l)} (R_{k+1}^{(r)})^T, \end{aligned}$$

At step k, α_k is chosen to minimize the Frobenius norm of the residual, namely

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{q_k \times q_k}} \|CD^T - \mathcal{L}(\boldsymbol{X}_k + R_k^{(l)} \alpha (R_k^{(r)})^T)\|_F^2$$

Back to the roots

Consider

$$\mathcal{A}\boldsymbol{x} = c \qquad \mathcal{A} \quad \text{nonsing, nonsym}$$

One-dimensional projection method: Saad, "Iterative methods for sparse linear systems", SIAM, 2003

$$x_{k+1} = x_k + \alpha_k r_k, \quad r_{k+1} = b - \mathcal{A}x_{k+1}$$

For instance, $\alpha_k := \frac{(\mathcal{A}r_k)^T r_k}{(\mathcal{A}r_k)^T \mathcal{A}r_k}$ minimizes $\|c - \mathcal{A}(x_k + \alpha r_k)\|^2$

Consider $A_1 \boldsymbol{X} B_1 + \ldots + A_\ell \boldsymbol{X} B_\ell = CD^T$ with

$$\mathcal{L}(V) := A_1 V B_1 + \ldots + A_\ell V B_\ell$$

"One-dimensional" subspace projection method:

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha}_k (R_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{q_k \times q_k} \\ \boldsymbol{R}_{k+1} &= CD^T - \mathcal{L}(\boldsymbol{X}_{k+1}), \qquad \boldsymbol{R}_{k+1} =: R_{k+1}^{(l)} (R_{k+1}^{(r)})^T, \end{aligned}$$

At step k, $\boldsymbol{\alpha}_k$ is chosen to minimize the Frobenius norm of the residual, namely

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^{q_k \times q_k}} \|CD^T - \mathcal{L}(\boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha} (R_k^{(r)})^T)\|_F^2$$

Low-rank short matrix recurrences

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha}_k (R_k^{(r)})^T \\ \boldsymbol{R}_{k+1} &= CD^T - \mathcal{L}(\boldsymbol{X}_{k+1}), \qquad \boldsymbol{R}_{k+1} =: R_{k+1}^{(l)} (R_{k+1}^{(r)})^T \end{aligned}$$

MINRES (SS-MR): $\boldsymbol{\alpha}_k$ is such that $(\boldsymbol{R}_k = R_k^{(l)}(R_k^{(r)})^T)$

$$\boxed{(R_k^{(l)})^T \mathcal{L}^* \left(\mathcal{L}(R_k^{(l)} \boldsymbol{\alpha} (R_k^{(r)})^T) \right) R_k^{(r)} = (R_k^{(l)})^T \mathcal{L}^*(\boldsymbol{R}_k) R_k^{(r)}.}$$

Moreover, $\mathrm{vec}(\boldsymbol{R}_{k+1}) \perp \mathcal{A} \cdot \mathrm{Range}(R_k^{(r)} \otimes R_k^{(l)})$

GCR (SS-GCR): (Generalized Conjugate-Residual with one-term recurrence)[3]

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P_k^{(l)} \alpha_k (P_k^{(r)})^T$$

where

$$P_{k+1}^{(l)}(P_{k+1}^{(r)})^T \leftarrow R_{k+1}^{(l)}(R_{k+1}^{(r)})^T + P_k^{(l)} \boldsymbol{\beta}_k (P_k^{(r)})^T \qquad (\boldsymbol{P}_{k+1} = \boldsymbol{R}_{k+1} + P_k^{(l)} \boldsymbol{\beta}_k (P_k^{(r)})^T)$$

and β_k solves

$$\boxed{(P_k^{(l)})^T \mathcal{L}^* \left(\mathcal{L}(P_k^{(l)} \boldsymbol{\beta} (P_k^{(r)})^T) \right) P_k^{(r)} = -(P_k^{(l)})^T \mathcal{L}^*(\mathcal{L}(\boldsymbol{R}_{k+1})) P_k^{(r)}}$$

[3] α computed with $R_k^{(l)}, R_k^{(r)}$ replaced by $P_k^{(l)}, P_k^{(r)}$.

Low-rank short matrix recurrences

$$\begin{aligned} \boldsymbol{X}_{k+1} &= \boldsymbol{X}_k + R_k^{(l)} \boldsymbol{\alpha}_k (R_k^{(r)})^T \\ \boldsymbol{R}_{k+1} &= CD^T - \mathcal{L}(\boldsymbol{X}_{k+1}), \qquad \boldsymbol{R}_{k+1} =: R_{k+1}^{(l)} (R_{k+1}^{(r)})^T \end{aligned}$$

MINRES (SS-MR): $\boldsymbol{\alpha}_k$ is such that $(\boldsymbol{R}_k = R_k^{(l)}(R_k^{(r)})^T)$

$$\boxed{(R_k^{(l)})^T \mathcal{L}^* \left(\mathcal{L}(R_k^{(l)} \boldsymbol{\alpha} (R_k^{(r)})^T) \right) R_k^{(r)} = (R_k^{(l)})^T \mathcal{L}^* (\boldsymbol{R}_k) R_k^{(r)}.}$$

Moreover, $\mathrm{vec}(\boldsymbol{R}_{k+1}) \perp \mathcal{A} \cdot \mathrm{Range}(R_k^{(r)} \otimes R_k^{(l)})$

GCR (SS-GCR): (Generalized Conjugate-Residual with one-term recurrence)[3]

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

where

$$P_{k+1}^{(l)} (P_{k+1}^{(r)})^T \leftarrow R_{k+1}^{(l)} (R_{k+1}^{(r)})^T + P_k^{(l)} \boldsymbol{\beta}_k (P_k^{(r)})^T \qquad (\boldsymbol{P}_{k+1} = \boldsymbol{R}_{k+1} + P_k^{(l)} \boldsymbol{\beta}_k (P_k^{(r)})^T)$$

and $\boldsymbol{\beta}_k$ solves

$$\boxed{(P_k^{(l)})^T \mathcal{L}^* \left(\mathcal{L}(P_k^{(l)} \boldsymbol{\beta} (P_k^{(r)})^T) \right) P_k^{(r)} = -(P_k^{(l)})^T \mathcal{L}^* (\mathcal{L}(\boldsymbol{R}_{k+1})) P_k^{(r)}}$$

[3] $\boldsymbol{\alpha}$ computed with $R_k^{(l)}, R_k^{(r)}$ replaced by $P_k^{(l)}, P_k^{(r)}$.

Similarly for a **subspace** CG

Assume that $\mathcal{A}$ is spd. We define $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}$, $\qquad \Phi(\boldsymbol{X}) = \frac{1}{2}\langle \boldsymbol{X}, \mathcal{L}(\boldsymbol{X})\rangle - \langle \boldsymbol{X}, \boldsymbol{C}\rangle$

The min 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

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P^{(l)}{}_k \boldsymbol{\alpha}_k (P_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{s_k\times s_k}$$

Residual and direction update: $\qquad (\boldsymbol{P}_{k+1} = P_{k+1}^{(l)}(P_{k+1}^{(r)})^T)$

$$\boldsymbol{R}_{k+1} = \boldsymbol{R}_k - \mathcal{L}(P_k^{(l)}\alpha_k(P_k^{(r)})^T), \qquad \boldsymbol{P}_{k+1} = \boldsymbol{R}_{k+1} + P_k^{(l)}\beta_k(P_k^{(r)})^T$$

1. Construct $\boldsymbol{\alpha}_k$ so that $\min_{\alpha\in\mathbb{R}^{s_k\times s_k}} \Phi(\boldsymbol{X}_k + P_k^{(l)}\alpha(P_k^{(r)})^T)$ that is

$$\boxed{(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\alpha}(P_k^{(r)})^T\right)P_k^{(r)} = (P_k^{(l)})^T\boldsymbol{R}_k P_k^{(r)}.}$$

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

$$\boxed{(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\beta}(P_k^{(r)})^T\right)P_k^{(r)} = -(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\beta_k(P_k^{(r)})^T\right)P_k^{(r)}}$$

Similarly for a **subspace** CG

Assume that $\mathcal{A}$ is spd. 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 min 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

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P^{(l)}{}_k \boldsymbol{\alpha}_k (P_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{s_k\times s_k}$$

Residual and direction update: $(\boldsymbol{P}_{k+1} = P_{k+1}^{(l)}(P_{k+1}^{(r)})^T)$

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

1. Construct $\boldsymbol{\alpha}_k$ so that $\min_{\alpha\in\mathbb{R}^{s_k\times s_k}} \Phi(\boldsymbol{X}_k + P_k^{(l)}\alpha(P_k^{(r)})^T)$ that is

$$(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\alpha}(P_k^{(r)})^T\right)P_k^{(r)} = (P_k^{(l)})^T\boldsymbol{R}_k P_k^{(r)}.$$

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

$$(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\beta}(P_k^{(r)})^T\right)P_k^{(r)} = -(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\beta}_k(P_k^{(r)})^T\right)P_k^{(r)}$$

Similarly for a **subspace** CG

Assume that $\mathcal{A}$ is spd. We define $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}$, $\qquad \Phi(\boldsymbol{X}) = \frac{1}{2}\langle \boldsymbol{X}, \mathcal{L}(\boldsymbol{X})\rangle - \langle \boldsymbol{X}, \boldsymbol{C}\rangle$

The min 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

$$\boldsymbol{X}_{k+1} = \boldsymbol{X}_k + P^{(l)}{}_k \boldsymbol{\alpha}_k (P_k^{(r)})^T \qquad \boldsymbol{\alpha}_k \in \mathbb{R}^{s_k\times s_k}$$

Residual and direction update: $\qquad (\boldsymbol{P}_{k+1} = P_{k+1}^{(l)}(P_{k+1}^{(r)})^T)$

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

1. Construct $\boldsymbol{\alpha}_k$ so that $\min_{\boldsymbol{\alpha}\in\mathbb{R}^{s_k\times s_k}} \Phi(\boldsymbol{X}_k + P_k^{(l)}\boldsymbol{\alpha}(P_k^{(r)})^T)$ that is

$$\boxed{(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\alpha}(P_k^{(r)})^T\right)P_k^{(r)} = (P_k^{(l)})^T\boldsymbol{R}_k P_k^{(r)}.}$$

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

$$\boxed{(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\beta}(P_k^{(r)})^T\right)P_k^{(r)} = -(P_k^{(l)})^T\mathcal{L}\left(P_k^{(l)}\boldsymbol{\beta}_k(P_k^{(r)})^T\right)P_k^{(r)}}$$

Making the idea practical. 1

$$\boldsymbol{X}_{k+1} = X_k^{(l)}\boldsymbol{\tau}_k(X_k^{(l)})^T + P_k^{(l)}\boldsymbol{\alpha}_k P_k^{(r)})^T = [X_k^{(l)}, P_k^{(l)}]\boldsymbol{\tau}_{k+1}[X_k^{(r)}, P_k^{(r)}]^T$$

Similarly for $\boldsymbol{P}_{k+1}$

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

$\Rightarrow$ Rank truncation

- Preconditioning strategies (problem dependent)

- Inner solution to obtain α_k, β_k:
  - Kronecker form if dimension allows
  - Matrix-oriented iterative solvers for large `maxrank`

Making the idea practical. 1

$$\boldsymbol{X}_{k+1} = X_k^{(l)} \boldsymbol{\tau}_k (X_k^{(l)})^T + P_k^{(l)} \boldsymbol{\alpha}_k P_k^{(r)})^T = [X_k^{(l)}, P_k^{(l)}] \boldsymbol{\tau}_{k+1} [X_k^{(r)}, P_k^{(r)}]^T$$

Similarly for $\boldsymbol{P}_{k+1}$

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

$\Rightarrow$ Rank truncation

- Preconditioning strategies (problem dependent)

- Inner solution to obtain α_k, β_k:
  - Kronecker form if dimension allows
  - Matrix-oriented iterative solvers for large `maxrank`

Making the idea practical. 1

$$\boldsymbol{X}_{k+1} \quad = \quad X_k^{(l)} \boldsymbol{\tau}_k (X_k^{(l)})^T + P_k^{(l)} \boldsymbol{\alpha}_k P_k^{(r)})^T = [X_k^{(l)}, P_k^{(l)}] \boldsymbol{\tau}_{k+1} [X_k^{(r)}, P_k^{(r)}]^T$$

Similarly for $\boldsymbol{P}_{k+1}$

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

$\Rightarrow$ Rank truncation

- Preconditioning strategies (problem dependent)

- Inner solution to obtain $\boldsymbol{\alpha}_k$, $\boldsymbol{\beta}_k$:
  - Kronecker form if dimension allows
  - Matrix-oriented iterative solvers for large `maxrank`

Making the idea practical. 2

Or, when RandNLA comes to rescue

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

$$\boldsymbol{R}_{k+1} = [C, -A_1X^{(l)}_{k+1}, \ldots, -A_\ell X^{(l)}_{k+1}]\boldsymbol{\rho}_{k+1}[D, B_1X^{(r)}_{k+1}, \ldots, B_\ell X^{(r)}_{k+1}]^T$$

Randomized strategies and target rank `maxrankR`

- Construct (thinner) approximation to row and column bases of $\boldsymbol{R}_{k+1}$
  (true $\boldsymbol{R}_{k+1}$ need **not** be retained)
- Construct well-conditioned basis for
  range($[C, -A_1X^{(l)}_{k+1}, \ldots, -A_\ell X^{(l)}_{k+1}]$), range($[D, B_1X^{(r)}_{k+1}, \ldots, B_\ell X^{(r)}_{k+1}]$)
- Estimate $\|\boldsymbol{R}_{k+1}\|_F$

Making the idea practical. 2

Or, when RandNLA comes to rescue

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

$$\boldsymbol{R}_{k+1} = [C, -A_1 X_{k+1}^{(l)}, \ldots, -A_\ell X_{k+1}^{(l)}] \boldsymbol{\rho}_{k+1} [D, B_1 X_{k+1}^{(r)}, \ldots, B_\ell X_{k+1}^{(r)}]^T$$

Randomized strategies and target rank `maxrankR`

- Construct (thinner) approximation to row and column bases of $\boldsymbol{R}_{k+1}$
  (true $\boldsymbol{R}_{k+1}$ need **not** be retained)
- Construct well-conditioned basis for
  range$([C, -A_1 X_{k+1}^{(l)}, \ldots, -A_\ell X_{k+1}^{(l)}])$, range$([D, B_1 X_{k+1}^{(r)}, \ldots, B_\ell X_{k+1}^{(r)}])$
- Estimate $\|\boldsymbol{R}_{k+1}\|_F$

A computational experiment. Parametric Darcy Flow Problem

$$
\begin{aligned}
\kappa(\mathbf{x},\mathbf{y})^{-1}\mathbf{u}(\mathbf{x},\mathbf{y}) + \nabla p(\mathbf{x},\mathbf{y}) &= 0 && \text{in} \quad D,\\
\nabla\cdot\mathbf{u}(\mathbf{x},\mathbf{y}) &= 0 && \text{in} \quad D,\\
p(\mathbf{x},\mathbf{y}) &= g(\mathbf{x}) && \text{on } \partial D_D,\\
\mathbf{u}(\mathbf{x},\mathbf{y})\cdot\mathbf{n} &= 0 && \text{on } \partial D_N.
\end{aligned}
$$

where κ^{-1} is a parameter-dependent function (and represented as a random field) of the form

$$
\kappa(\mathbf{x},\mathbf{y})^{-1} := \kappa_0(\mathbf{x}) + \sum_{r=1}^{M}\kappa_r(\mathbf{x})y_r, \qquad \mathbf{x}\in D, \qquad \mathbf{y}\in\Gamma,
$$

and the parameters $y_r := \xi_r(\omega)$ are images of independent uniform random variables $\xi_r \sim U(-1,1)$ with joint probability density $\rho = (1/2)^M$.

Stochastic Galerkin Approximation:

$$
A_0\mathbf{X}G_0 + A_1\mathbf{X}G_1 + \cdots + A_M\mathbf{X}G_M = fg_0^T,
$$

where $[G_r]_{i,j} := \mathbb{E}\left[y_r\psi_i\psi_j\right], \qquad i,j = 1,\ldots,n_q, \quad r = 0,1,\ldots,M$, and

$$
A_0 = \begin{pmatrix} K_0 & B^T \\ B & 0 \end{pmatrix}, \qquad A_j = \begin{pmatrix} K_r & 0 \\ 0 & 0 \end{pmatrix}, \quad j \geq 1,
$$

Indefinite (constraint) Preconditioner: $\mathcal{P}(X) = \widehat{A}_0^{-1}XG_0^{-1}$, with $\widehat{A}_0 \approx A_0$ ($\widehat{K}_0 \approx K_0$)

A computational experiment. Parametric Darcy Flow Problem

$$\begin{aligned} \kappa(\mathbf{x},\mathbf{y})^{-1}\mathbf{u}(\mathbf{x},\mathbf{y}) + \nabla p(\mathbf{x},\mathbf{y}) &= 0 \qquad \text{in} \quad D, \\ \nabla \cdot \mathbf{u}(\mathbf{x},\mathbf{y}) &= 0 \qquad \text{in} \quad D, \\ p(\mathbf{x},\mathbf{y}) &= g(\mathbf{x}) \qquad \text{on } \partial D_D, \\ \mathbf{u}(\mathbf{x},\mathbf{y}) \cdot \mathbf{n} &= 0 \qquad \text{on } \partial D_N. \end{aligned}$$

where κ^{-1} is a parameter-dependent function (and represented as a random field) of the form

$$\kappa(\mathbf{x},\mathbf{y})^{-1} := \kappa_0(\mathbf{x}) + \sum_{r=1}^{M} \kappa_r(\mathbf{x}) y_r, \qquad \mathbf{x} \in D, \qquad \mathbf{y} \in \Gamma,$$

and the parameters $y_r := \xi_r(\omega)$ are images of independent uniform random variables $\xi_r \sim U(-1,1)$ with joint probability density $\rho = (1/2)^M$.

Stochastic Galerkin Approximation:

$$A_0 \boldsymbol{X} G_0 + A_1 \boldsymbol{X} G_1 + \cdots + A_M \boldsymbol{X} G_M = f g_0^T,$$

where $[G_r]_{i,j} := \mathbb{E}\left[y_r \psi_i \psi_j\right], \qquad i,j = 1,\ldots,n_q, \quad r = 0,1,\ldots,M$, and

$$A_0 = \begin{pmatrix} K_0 & B^T \\ B & 0 \end{pmatrix}, \qquad A_j = \begin{pmatrix} K_r & 0 \\ 0 & 0 \end{pmatrix}, \quad j \geq 1,$$

Indefinite (constraint) Preconditioner: $\mathcal{P}(X) = \widehat{A}_0^{-1} X G_0^{-1}$, with $\widehat{A}_0 \approx A_0$ ($\widehat{K}_0 \approx K_0$)

A computational experiment. Fast Decay Case

$n_h = 49,152$											
			SS-MR					MINRES			
M	q	n_q	maxrank	Rank	# its	Res	Time	k	Res	Time
5	4	126	40	35	10	6.1e-07	11.0	18	8.6e-07	10.7
	5	252	40	40	12	7.0e-07	14.2	19	2.7e-07	23.2
	6	462	40	40	13	8.5e-07	16.3	19	2.2e-07	47.7
9	4	715	40	40	11	8.4e-07	32.4	18	8.6e-07	106.1
	5	2,002	40	40	13	9.3e-07	**39.5**	19	2.7e-07	574.8
	6	5,005	40	40	15	7.9e-07	**47.1**	*	*	*
$n_h = 196,608$										
			SS-MR					MINRES		
M	q	n_q	maxrank	Rank	# its	Res	Time	k	Res	Time
5	4	126	30	30	12	7.4e-07	43.9	18	8.6e-07	48.5
	5	252	40	39	12	7.8e-07	59.0	19	2.5e-07	132.8
	6	462	40	40	13	9.0e-07	66.8	19	2.0e-07	390.0
9	4	715	40	40	11	7.7e-07	125.8	18	8.6e-07	875.0
	5	2,002	40	40	13	8.0e-07	**153.9**	*	*	*
	6	5,005	40	40	14	9.1e-07	**163.6**	*	*	*

Table: **Test Problem 1** Performance of SS-MR on the matrix equation formulation with one-term preconditioning, and MINRES on the associated linear system with the same preconditioner, $\texttt{tol} = 10^{-6}$ and $\texttt{toltrank} = 10^{-8}$.

A computational experiment. Slow Decay Case

$n_h = 49,152$										
			SS-MR					MINRES		
M	q	n_q	`maxrank`	# its	PCG	Res	Time	k	Res	Time
	4	495	80	7	[6, 7]	7.5e-07	24.0	10	6.6e-07	32.2
8	5	1,287	100	7	[6, 7]	5.1e-07	29.9	10	5.2e-08	109.6
	6	3,003	100	7	[6, 7]	7.0e-07	**33.2**	9	9.4e-07	381.1
	4	1,820	120	7	[6, 7]	8.0e-07	74.6	10	3.4e-07	233.7
12	5	6,188	140	7	[6, 7]	8.0e-07	**101.8**	*	*	*
	6	18,564	140	8	[5 ,7]	5.1e-07	**139.2**	*	*	*

$n_h = 196,608$										
			SS-MR					MINRES		
M	q	n_q	`maxrank`	# its	PCG	Res	Time	k	Res	Time
	4	495	80	7	[6, 7]	6.8e-07	99.7	10	6.6e-07	251.6
8	5	1,287	100	7	[6, 7]	4.1e-07	165.0	10	5.4e-08	986.7
	6	3,003	100	7	[6, 7]	5.1e-07	**168.0**	*	*	*
	4	1,820	120	7	[6, 7]	6.8e-07	420.7	*	*	*
12	5	6,188	140	7	[6, 7]	6.2e-07	**482.6**	*	*	*
	6	18,564	140	7	[6, 7]	7.9e-07	**490.7**	*	*	*

Table: **Test Problem 2** Performance of SS-MR on the matrix equation formulation with one-term preconditioning, and MINRES on the associated linear system with the same preconditioner, `tol` $= 10^{-6}$ and `toltrank` $= 10^{-8}$.

.

Inner solver: matlab preconditioned CG

Outlook

- Experiments for sym and nonsym problems are very promising
- Reduced problem deserves further attention
- Tensor version also available

References

[1] D. Palitta, M. Iannacito, and V. S., *A subspace-conjugate gradient method for linear matrix equations* SIMAX, 46 (2025)

[2] M. Iannacito, L. Piccinini, and V. S., *Subspace gradient descent methods for linear tensor equations*, ArXiv 2602.21974. To appear in Springer Indam Series.

[3] D. Palitta, C. E. Powell, and V. S., *A class of low-rank short recurrences for nonsymmetric linear matrix equations*, ArXiv 2605.01276. May 2026.