Preconditioning Techniques for Multiterm Generalized Sylvester Equations

Precond 2024, Atlanta

Yannis Voet

Chair of Modeling and Numerical Simulation
EPFL

June 10th, 2024

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{m\times m}$ for all $k = 1, \ldots, r$ and $X, E \in \mathbb{R}^{m\times n}$.

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{m\times m}$ for all $k = 1, \ldots, r$ and $X, E \in \mathbb{R}^{m\times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{m \times m}$ for all $k = 1, \ldots, r$ and $X, E \in \mathbb{R}^{m \times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]
- Model reduction (Benner and Breiten 2013; Damm 2008; Shank et al. 2016).

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n \times n}$, $B_k \in \mathbb{R}^{m \times m}$ for all $k = 1, \ldots, r$ and $X, E \in \mathbb{R}^{m \times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]
- Model reduction (Benner and Breiten 2013; Damm 2008; Shank et al. 2016).
- Eigenvalue problems (Ringh et al. 2018).

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{m\times m}$ for all $k = 1, \ldots, r$ and $X, E \in \mathbb{R}^{m\times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]
- Model reduction (Benner and Breiten 2013; Damm 2008; Shank et al. 2016).
- Eigenvalue problems (Ringh et al. 2018).
- Computational neuroscience (Kürschner et al. 2019).

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{m\times m}$ for all $k = 1, \dots, r$ and $X, E \in \mathbb{R}^{m\times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]
- Model reduction (Benner and Breiten 2013; Damm 2008; Shank et al. 2016).
- Eigenvalue problems (Ringh et al. 2018).
- Computational neuroscience (Kürschner et al. 2019).
- ...

Introduction

We consider solving generalized multiterm Sylvester equations

$$\sum_{k=1}^{r} B_k X A_k^T = E$$

where $A_k \in \mathbb{R}^{n\times n}$, $B_k \in \mathbb{R}^{m\times m}$ for all $k = 1, \dots, r$ and $X, E \in \mathbb{R}^{m\times n}$.

Now appear in an increasing number of applications:

- Finite difference (Palitta and Simoncini 2016; Hao and Simoncini 2021) and finite element (Ernst et al. 2009; Ullmann 2010; Mantzaflaris et al. 2017; Scholz et al. 2018) discretizations of (stochastic) PDEs [cf. Catherine Powell's talk at SIAM LA24]
- Model reduction (Benner and Breiten 2013; Damm 2008; Shank et al. 2016).
- Eigenvalue problems (Ringh et al. 2018).
- Computational neuroscience (Kürschner et al. 2019).
- ...

Includes as particular cases the famed (standard) Sylvester, Lyapunov and Stein equations.

Difficulties

Solution strategies critically depend on the number of terms r, the overall structure of the equation and properties of the coefficients.

$r = 1$:

$$B_1 X A_1^T = E$$

Trivial.

Difficulties

Solution strategies critically depend on the number of terms r, the overall structure of the equation and properties of the coefficients.

$r = 2$:

$$B_1 X A_1^T + B_2 X A_2^T = E$$

Much more challenging:

- Direct solution techniques (Bartels and Stewart 1972; Gardiner et al. 1992).
- Block recursive splitting (Jonsson and Kågström 2002a; Jonsson and Kågström 2002b).
- Alternating Direction Implicit (ADI) (Wachspress 1988).
- Data-sparse methods (e.g. low-rank) (Massei et al. 2018; Palitta and Simoncini 2018; Grasedyck 2004; Kressner and Tobler 2010).
- Matrix oriented (truncated) CG/GMRES/... (Hochbruck and Starke 1995).

See e.g. Simoncini 2016; Benner and Saak 2013 for an overview.

Difficulties

Solution strategies critically depend on the number of terms r, the overall structure of the equation and properties of the coefficients.

$r \geq 3$:

$$\sum_{k=1}^{r} B_k X A_k^T = E \tag{1}$$

- Low-rank methods (Benner and Breiten 2013; Kressner and Sirković 2015; Jarlebring et al. 2018).
- ADI (Benner and Saak 2013).
- Matrix oriented (truncated) CG/GMRES/... (Jbilou et al. 1999; Bouhamidi and Jbilou 2008).

Notes:

- No general direct solution method for $r \geq 3$ with complexity $O(n^3 + m^3)$.
- Projection type techniques require solving a small size version of (1).

Kronecker form

Define the linear operator $\mathcal{M}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and its Kronecker representation $M \in \mathbb{R}^{mn\times mn}$ as

$$\mathcal{M}(X) = \sum_{k=1}^{r} B_k X A_k^T, \quad M = \sum_{k=1}^{r} A_k \otimes B_k.$$

Exploit the equivalence

$$\mathcal{M}(X) = Y \iff M\mathbf{x} = \mathbf{y}$$

with $\mathbf{x} = \mathrm{vec}(X)$, $\mathbf{y} = \mathrm{vec}(Y)$.

$\rightarrow$ At the heart of matrix oriented versions of CG, GMRES,...

Kronecker form

Define the linear operator $\mathcal{M}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and its Kronecker representation $M \in \mathbb{R}^{mn\times mn}$ as

$$\mathcal{M}(X) = \sum_{k=1}^{r} B_k X A_k^T, \quad M = \sum_{k=1}^{r} A_k \otimes B_k.$$

Exploit the equivalence

$$\mathcal{M}(X) = Y \iff M\mathbf{x} = \mathbf{y}$$

with $\mathbf{x} = \mathrm{vec}(X)$, $\mathbf{y} = \mathrm{vec}(Y)$.

$\rightarrow$ At the heart of matrix oriented versions of CG, GMRES,...

Goal: Use the Kronecker form of the equation to build low Kronecker rank approximations of the operator or its inverse, without restrictive assumptions on the coefficients.

Kronecker form

Define the linear operator $\mathcal{M}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and its Kronecker representation $M \in \mathbb{R}^{mn\times mn}$ as

$$\mathcal{M}(X) = \sum_{k=1}^{r} B_k X A_k^T, \quad M = \sum_{k=1}^{r} A_k \otimes B_k.$$

Exploit the equivalence

$$\mathcal{M}(X) = Y \iff M\mathbf{x} = \mathbf{y}$$

with $\mathbf{x} = \mathrm{vec}(X)$, $\mathbf{y} = \mathrm{vec}(Y)$.

$\rightarrow$ At the heart of matrix oriented versions of CG, GMRES,...

Goal: Use the Kronecker form of the equation to build low Kronecker rank approximations of the operator or its inverse, without restrictive assumptions on the coefficients.
👍 Very general solution strategies.

Kronecker form

Define the linear operator $\mathcal{M}\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ and its Kronecker representation $M \in \mathbb{R}^{mn\times mn}$ as

$$\mathcal{M}(X) = \sum_{k=1}^{r} B_k X A_k^T, \quad M = \sum_{k=1}^{r} A_k \otimes B_k.$$

Exploit the equivalence

$$\mathcal{M}(X) = Y \iff M\mathbf{x} = \mathbf{y}$$

with $\mathbf{x} = \mathrm{vec}(X)$, $\mathbf{y} = \mathrm{vec}(Y)$.

→ At the heart of matrix oriented versions of CG, GMRES,...

Goal: Use the Kronecker form of the equation to build low Kronecker rank approximations of the operator or its inverse, without restrictive assumptions on the coefficients.

👍 Very general solution strategies.

👎 Limited to small or medium size equations ($m, n < 10^4$).

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).
- Nearest Kronecker product (Van Loan and Pitsianis 1993): Find the best factor matrices $Y \in \mathbb{R}^{n \times n}$ and $Z \in \mathbb{R}^{m \times m}$ that minimize

$$\phi_M(Y, Z) = \|M - Y \otimes Z\|_F.$$

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).
- Nearest Kronecker product (Van Loan and Pitsianis 1993): Find the best factor matrices $Y \in \mathbb{R}^{n \times n}$ and $Z \in \mathbb{R}^{m \times m}$ that minimize

$$\phi_M(Y, Z) = \|M - Y \otimes Z\|_F.$$

Equivalent to the (matrix) low-rank approximation problem

$$\min \|\mathcal{R}(M) - \operatorname{vec}(Y) \operatorname{vec}(Z)^T\|_F$$

where $\mathcal{R} \colon \mathbb{R}^{nm \times nm} \to \mathbb{R}^{n^2 \times m^2}$ is a rearrangement operator.

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).
- Nearest Kronecker product (Van Loan and Pitsianis 1993): Find the best factor matrices $Y \in \mathbb{R}^{n\times n}$ and $Z \in \mathbb{R}^{m\times m}$ that minimize

$$\phi_M(Y, Z) = \|M - Y \otimes Z\|_F.$$

Equivalent to the (matrix) low-rank approximation problem

$$\min \|\mathcal{R}(M) - \operatorname{vec}(Y) \operatorname{vec}(Z)^T\|_F$$

where $\mathcal{R}\colon \mathbb{R}^{nm\times nm} \to \mathbb{R}^{n^2\times m^2}$ is a rearrangement operator.

Observation: M has Kronecker rank $r \iff \mathcal{R}(M)$ has (matrix) rank r.

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).
- Nearest Kronecker product (Van Loan and Pitsianis 1993): Find the best factor matrices $Y \in \mathbb{R}^{n \times n}$ and $Z \in \mathbb{R}^{m \times m}$ that minimize

$$\phi_M(Y, Z) = \|M - Y \otimes Z\|_F.$$

Equivalent to the (matrix) low-rank approximation problem

$$\min \|\mathcal{R}(M) - \operatorname{vec}(Y) \operatorname{vec}(Z)^T\|_F$$

where $\mathcal{R}\colon \mathbb{R}^{nm \times nm} \to \mathbb{R}^{n^2 \times m^2}$ is a rearrangement operator.

Observation: M has Kronecker rank $r \iff \mathcal{R}(M)$ has (matrix) rank r.

Algorithm:

1. Compute the "SVD representation" of M via the SVD of $\mathcal{R}(M)$

$$\mathcal{R}(M) = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \iff M = \sum_{k=1}^{r} \sigma_k (U_k \otimes V_k).$$

Kronecker product approximation

- Some early attempts for $r = 2$ employing SSOR preconditioning (Hochbruck and Starke 1995; Bouhamidi and Jbilou 2008).
- Nearest Kronecker product (Van Loan and Pitsianis 1993): Find the best factor matrices $Y \in \mathbb{R}^{n\times n}$ and $Z \in \mathbb{R}^{m\times m}$ that minimize

$$\phi_M(Y, Z) = \|M - Y \otimes Z\|_F.$$

Equivalent to the (matrix) low-rank approximation problem

$$\min \|\mathcal{R}(M) - \mathrm{vec}(Y)\,\mathrm{vec}(Z)^T\|_F$$

where $\mathcal{R}\colon \mathbb{R}^{nm\times nm} \to \mathbb{R}^{n^2\times m^2}$ is a rearrangement operator.

Observation: M has Kronecker rank $r \iff \mathcal{R}(M)$ has (matrix) rank r.

Algorithm:

1. Compute the "SVD representation" of M via the SVD of $\mathcal{R}(M)$

$$\mathcal{R}(M) = \sum_{k=1}^{r} \sigma_k \mathbf{u}_k \mathbf{v}_k^T \iff M = \sum_{k=1}^{r} \sigma_k (U_k \otimes V_k).$$

2. Retain the leading $q \leq 2$ terms and use it as preconditioner

$$P = \sum_{k=1}^{q} \sigma_k (U_k \otimes V_k).$$

Theoretical results

- For $q = 1$, P is block-banded, nonnegative and positive definite if M is (Van Loan and Pitsianis 1993, Theorems 5.1, 5.3 and 5.8).

Theoretical results

- For $q = 1$, P is block-banded, nonnegative and positive definite if M is (Van Loan and Pitsianis 1993, Theorems 5.1, 5.3 and 5.8).
- For $q \geq 2$, all factor matrices of P are banded and symmetric if those of M are.

Theoretical results

- For $q = 1$, P is block-banded, nonnegative and positive definite if M is (Van Loan and Pitsianis 1993, Theorems 5.1, 5.3 and 5.8).
- For $q \geq 2$, all factor matrices of P are banded and symmetric if those of M are.

Theorem

If $M, P \in \mathbb{R}^{n \times n}$ are symmetric positive definite

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(1 - \frac{1}{\lambda_i(M,P)}\right)^2} \leq \kappa(M)\sqrt{\sum_{k=q+1}^{r}\left(\frac{\sigma_k}{\sigma_1}\right)^2}.$$

$\rightarrow$ Effective preconditioner if $\mathcal{R}(M)$ features a fast singular value decay.

Kronecker approximate inverse

Find factor matrices $C \in \mathbb{R}^{n\times n}$ and $D \in \mathbb{R}^{m\times m}$ such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.

Kronecker approximate inverse

Find factor matrices $C \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{m \times m}$ such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.
Consider the minimization problem

$$\min_{C,D} \|I - \sum_{k=1}^{r} A_k C \otimes B_k D\|_F.$$

Kronecker approximate inverse

Find factor matrices $C \in \mathbb{R}^{n \times n}$ and $D \in \mathbb{R}^{m \times m}$ such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.
Consider the minimization problem

$$\min_{C,D} \|I - \sum_{k=1}^{r} A_k C \otimes B_k D\|_F.$$

Use the reshaping that transforms

$$M = \begin{pmatrix} M_{11} & \dots & M_{1n} \\ \vdots & \ddots & \vdots \\ M_{n1} & \dots & M_{nn} \end{pmatrix} \quad \text{to} \quad \tilde{M} = \begin{pmatrix} M_{11} \\ M_{21} \\ \vdots \\ M_{nn} \end{pmatrix}$$

$\rightarrow$ Alternately solve a sequence of least squares problems for reshaped quantities

$$\min_{D} \|\tilde{I} - \mathcal{B} D\|_F \quad \text{and} \quad \min_{C} \|\hat{I} - \mathcal{A} C\|_F$$

$$\begin{aligned} \mathcal{B} &= (U \otimes I_m)B, & U &= [\mathrm{vec}(A_1 C), \dots, \mathrm{vec}(A_r C)], & B &= [B_1; B_2; \dots; B_r], \\ \mathcal{A} &= (V \otimes I_n)A, & V &= [\mathrm{vec}(B_1 D), \dots, \mathrm{vec}(B_r D)], & A &= [A_1; A_2; \dots; A_r]. \end{aligned}$$

Algorithm (via normal equations)

Input: Initial guess $C \in \mathbb{R}^{n \times n}$, tolerance $\epsilon > 0$ and number of iterations $N \in \mathbb{N}$
Output: C and D such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$

1: Set $r = \infty$, $j = 0$ ▷ Initialization
2: **while** $\sqrt{r} > \epsilon$ **and** $j \leq N$ **do**
▷ Optimizing for D
3: Compute $\beta_{kl} = \langle A_k^T A_l, CC^T \rangle_F$ for $k, l = 1, \dots, r$ ▷ $O(rn^3 + r^2 n^2)$
4: Compute $\delta_k = \langle A_k^T, C \rangle_F$ for $k = 1, \dots, r$ ▷ $O(rn^2)$
5: Form $\mathcal{B}^T \mathcal{B} = \sum_{k,l=1}^{r} \beta_{kl} B_k^T B_l$ ▷ $O(rm^3 + r^2 m^2)$
6: Form $\mathcal{B}^T \tilde{I} = \sum_{k=1}^{r} \delta_k B_k^T$ ▷ $O(rm^2)$
7: Solve $\mathcal{B}^T \mathcal{B} D = \mathcal{B}^T \tilde{I}$ ▷ $O(m^3)$
▷ Optimizing for C
8: Compute $\alpha_{kl} = \langle B_k^T B_l, DD^T \rangle_F$ for $k, l = 1, \dots, r$ ▷ $O(rm^3 + r^2 m^2)$
9: Compute $\gamma_k = \langle B_k^T, D \rangle_F$ for $k = 1, \dots, r$ ▷ $O(rm^2)$
10: Form $\mathcal{A}^T \mathcal{A} = \sum_{k,l=1}^{r} \alpha_{kl} A_k^T A_l$ ▷ $O(rn^3 + r^2 n^2)$
11: Form $\mathcal{A}^T \tilde{I} = \sum_{k=1}^{r} \gamma_k A_k^T$ ▷ $O(rn^2)$
12: Solve $\mathcal{A}^T \mathcal{A} C = \mathcal{A}^T \tilde{I}$ ▷ $O(n^3)$

13: Update β_{kl} and δ_k following lines 3 and 4, respectively ▷ Residual
14: Compute $r = nm - 2\sum_{k=1}^{r} \gamma_k \delta_k + \sum_{k,l=1}^{r} \alpha_{kl} \beta_{kl}$ ▷ $O(r^2)$
15: Update $j = j + 1$
16: **end while**

Algorithm: ALS for Kronecker rank 1 approximate inverse

Algorithm (via normal equations)

Input: Initial guess $C \in \mathbb{R}^{n\times n}$, tolerance $\epsilon > 0$ and number of iterations $N \in \mathbb{N}$

Output: C and D such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$

1: Set $r = \infty$, $j = 0$ ▷ Initialization
2: **while** $\sqrt{r} > \epsilon$ **and** $j \leq N$ **do**
▷ Optimizing for D
3: Compute $\beta_{kl} = \langle A_k^T A_l, CC^T\rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rn^3 + r^2n^2)$
4: Compute $\delta_k = \langle A_k^T, C\rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rn^2)$
5: Form $\mathcal{B}^T\mathcal{B} = \sum_{k,l=1}^{r} \beta_{kl} B_k^T B_l$ ▷ $O(rm^3 + r^2m^2)$
6: Form $\mathcal{B}^T\tilde{I} = \sum_{k=1}^{r} \delta_k B_k^T$ ▷ $O(rm^2)$
7: Solve $\mathcal{B}^T\mathcal{B}D = \mathcal{B}^T\tilde{I}$ ▷ $O(m^3)$
▷ Optimizing for C
8: Compute $\alpha_{kl} = \langle B_k^T B_l, DD^T\rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rm^3 + r^2m^2)$
9: Compute $\gamma_k = \langle B_k^T, D\rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rm^2)$
10: Form $\mathcal{A}^T\mathcal{A} = \sum_{k,l=1}^{r} \alpha_{kl} A_k^T A_l$ ▷ $O(rn^3 + r^2n^2)$
11: Form $\mathcal{A}^T\tilde{I} = \sum_{k=1}^{r} \gamma_k A_k^T$ ▷ $O(rn^2)$
12: Solve $\mathcal{A}^T\mathcal{A}C = \mathcal{A}^T\tilde{I}$ ▷ $O(n^3)$

13: Update β_{kl} and δ_k following lines 3 and 4, respectively ▷ Residual
14: Compute $r = nm - 2\sum_{k=1}^{r} \gamma_k\delta_k + \sum_{k,l=1}^{r} \alpha_{kl}\beta_{kl}$ ▷ $O(r^2)$
15: Update $j = j + 1$
16: **end while**

Algorithm: ALS for Kronecker rank 1 approximate inverse

Algorithm (via normal equations)

Input: Initial guess $C \in \mathbb{R}^{n \times n}$, tolerance $\epsilon > 0$ and number of iterations $N \in \mathbb{N}$

Output: C and D such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$

1: Set $r = \infty$, $j = 0$ ▷ Initialization

2: **while** $\sqrt{r} > \epsilon$ **and** $j \leq N$ **do**

▷ Optimizing for D

3: Compute $\beta_{kl} = \langle A_k^T A_l, CC^T \rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rn^3 + r^2 n^2)$

4: Compute $\delta_k = \langle A_k^T, C \rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rn^2)$

5: Form $\mathcal{B}^T \mathcal{B} = \sum_{k,l=1}^{r} \beta_{kl} B_k^T B_l$ ▷ $O(rm^3 + r^2 m^2)$

6: Form $\mathcal{B}^T \tilde{I} = \sum_{k=1}^{r} \delta_k B_k^T$ ▷ $O(rm^2)$

7: Solve $\mathcal{B}^T \mathcal{B} D = \mathcal{B}^T \tilde{I}$ ▷ $O(m^3)$

▷ Optimizing for C

8: Compute $\alpha_{kl} = \langle B_k^T B_l, DD^T \rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rm^3 + r^2 m^2)$

9: Compute $\gamma_k = \langle B_k^T, D \rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rm^2)$

10: Form $\mathcal{A}^T \mathcal{A} = \sum_{k,l=1}^{r} \alpha_{kl} A_k^T A_l$ ▷ $O(rn^3 + r^2 n^2)$

11: Form $\mathcal{A}^T \tilde{I} = \sum_{k=1}^{r} \gamma_k A_k^T$ ▷ $O(rn^2)$

12: Solve $\mathcal{A}^T \mathcal{A} C = \mathcal{A}^T \tilde{I}$ ▷ $O(n^3)$

13: Update β_{kl} and δ_k following lines 3 and 4, respectively ▷ Residual

14: Compute $r = nm - 2\sum_{k=1}^{r} \gamma_k \delta_k + \sum_{k,l=1}^{r} \alpha_{kl} \beta_{kl}$ ▷ $O(r^2)$

15: Update $j = j + 1$

16: **end while**

Algorithm: ALS for Kronecker rank 1 approximate inverse

Algorithm (via normal equations)

Input: Initial guess $C \in \mathbb{R}^{n\times n}$, tolerance $\epsilon > 0$ and number of iterations $N \in \mathbb{N}$
Output: C and D such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$

1: Set $r = \infty$, $j = 0$ ▷ Initialization
2: **while** $\sqrt{r} > \epsilon$ **and** $j \leq N$ **do**
▷ Optimizing for D
3: Compute $\beta_{kl} = \langle A_k^T A_l, CC^T \rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rn^3 + r^2 n^2)$
4: Compute $\delta_k = \langle A_k^T, C \rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rn^2)$
5: Form $\mathcal{B}^T \mathcal{B} = \sum_{k,l=1}^{r} \beta_{kl} B_k^T B_l$ ▷ $O(rm^3 + r^2 m^2)$
6: Form $\mathcal{B}^T \tilde{I} = \sum_{k=1}^{r} \delta_k B_k^T$ ▷ $O(rm^2)$
7: Solve $\mathcal{B}^T \mathcal{B} D = \mathcal{B}^T \tilde{I}$ ▷ $O(m^3)$
▷ Optimizing for C
8: Compute $\alpha_{kl} = \langle B_k^T B_l, DD^T \rangle_F$ for $k, l = 1, \ldots, r$ ▷ $O(rm^3 + r^2 m^2)$
9: Compute $\gamma_k = \langle B_k^T, D \rangle_F$ for $k = 1, \ldots, r$ ▷ $O(rm^2)$
10: Form $\mathcal{A}^T \mathcal{A} = \sum_{k,l=1}^{r} \alpha_{kl} A_k^T A_l$ ▷ $O(rn^3 + r^2 n^2)$
11: Form $\mathcal{A}^T \tilde{I} = \sum_{k=1}^{r} \gamma_k A_k^T$ ▷ $O(rn^2)$
12: Solve $\mathcal{A}^T \mathcal{A} C = \mathcal{A}^T \tilde{I}$ ▷ $O(n^3)$

13: Update β_{kl} and δ_k following lines 3 and 4, respectively ▷ Residual
14: Compute $r = nm - 2\sum_{k=1}^{r} \gamma_k \delta_k + \sum_{k,l=1}^{r} \alpha_{kl} \beta_{kl}$ ▷ $O(r^2)$
15: Update $j = j + 1$
16: **end while**

Algorithm: ALS for Kronecker rank 1 approximate inverse

Algorithm (via normal equations)

Input: Initial guess $C \in \mathbb{R}^{n \times n}$, tolerance $\epsilon > 0$ and number of iterations $N \in \mathbb{N}$
Output: C and D such that $C \otimes D \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$

1: Set $r = \infty$, $j = 0$ $\triangleright$ Initialization
2: **while** $\sqrt{r} > \epsilon$ **and** $j \leq N$ **do**
$\triangleright$ Optimizing for D
3: Compute $\beta_{kl} = \langle A_k^T A_l, CC^T \rangle_F$ for $k, l = 1, \ldots, r$ $\triangleright$ $O(rn^3 + r^2n^2)$
4: Compute $\delta_k = \langle A_k^T, C \rangle_F$ for $k = 1, \ldots, r$ $\triangleright$ $O(rn^2)$
5: Form $\mathcal{B}^T\mathcal{B} = \sum_{k,l=1}^{r} \beta_{kl} B_k^T B_l$ $\triangleright$ $O(rm^3 + r^2m^2)$
6: Form $\mathcal{B}^T\tilde{I} = \sum_{k=1}^{r} \delta_k B_k^T$ $\triangleright$ $O(rm^2)$
7: Solve $\mathcal{B}^T\mathcal{B}D = \mathcal{B}^T\tilde{I}$ $\triangleright$ $O(m^3)$
$\triangleright$ Optimizing for C
8: Compute $\alpha_{kl} = \langle B_k^T B_l, DD^T \rangle_F$ for $k, l = 1, \ldots, r$ $\triangleright$ $O(rm^3 + r^2m^2)$
9: Compute $\gamma_k = \langle B_k^T, D \rangle_F$ for $k = 1, \ldots, r$ $\triangleright$ $O(rm^2)$
10: Form $\mathcal{A}^T\mathcal{A} = \sum_{k,l=1}^{r} \alpha_{kl} A_k^T A_l$ $\triangleright$ $O(rn^3 + r^2n^2)$
11: Form $\mathcal{A}^T\tilde{I} = \sum_{k=1}^{r} \gamma_k A_k^T$ $\triangleright$ $O(rn^2)$
12: Solve $\mathcal{A}^T\mathcal{A}C = \mathcal{A}^T\tilde{I}$ $\triangleright$ $O(n^3)$

13: Update β_{kl} and δ_k following lines 3 and 4, respectively $\triangleright$ Residual
14: Compute $r = nm - 2\sum_{k=1}^{r} \gamma_k \delta_k + \sum_{k,l=1}^{r} \alpha_{kl}\beta_{kl}$ $\triangleright$ $O(r^2)$
15: Update $j = j + 1$
16: **end while**

Algorithm: ALS for Kronecker rank 1 approximate inverse

Notes

- Complexity **for dense matrices**: $O(Nr(n^3 + m^3) + Nr^2(n^2 + m^2))$.

Notes

- Complexity **for dense matrices**: $O(Nr(n^3 + m^3) + Nr^2(n^2 + m^2))$.
- Generalizable to Kronecker rank q approximate inverses:
  $\sum_{s=1}^{q} C_s \otimes D_s \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.

Notes

- Complexity **for dense matrices**: $O(Nr(n^3+m^3)+Nr^2(n^2+m^2))$.
- Generalizable to Kronecker rank q approximate inverses: $\sum_{s=1}^{q} C_s \otimes D_s \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.
- Applying the preconditioner only requires matrix-matrix multiplications:

$$\mathcal{P}(X) = \sum_{s=1}^{q} D_s X C_s^T.$$

Notes

- Complexity **for dense matrices**: $O(Nr(n^3+m^3)+Nr^2(n^2+m^2))$.
- Generalizable to Kronecker rank q approximate inverses: $\sum_{s=1}^{q} C_s \otimes D_s \approx \left(\sum_{k=1}^{r} A_k \otimes B_k\right)^{-1}$.
- Applying the preconditioner only requires matrix-matrix multiplications:

$$\mathcal{P}(X) = \sum_{s=1}^{q} D_s X C_s^T.$$

- The residual directly controls the clustering of the eigenvalues of the preconditioned operator (Grote and Huckle 1997, Theorem 3.2). For $M, P \in \mathbb{R}^{n\times n}$

$$\sum_{i=1}^{n} |1-\lambda_i(MP)|^2 \leq \|I - MP\|_F^2.$$

Sparse Kronecker approximate inverse

Define sets of sparse matrices $\mathcal{S}_C \subset \mathbb{R}^{n\times n}$ and $\mathcal{S}_D \subset \mathbb{R}^{m\times m}$ with prescribed sparsity (based e.g. on powers of $\sum_{k=1}^r A_k$ and $\sum_{k=1}^r B_k$ or some variation) (Huckle 1999).

Solve alternately

$$\min_{D\in\mathcal{S}_D} \|\tilde{I} - \mathcal{B}D\|_F, \quad \text{and} \quad \min_{C\in\mathcal{S}_C} \|\hat{I} - \mathcal{A}C\|_F.$$

Sparse Kronecker approximate inverse

Define sets of sparse matrices $\mathcal{S}_C \subset \mathbb{R}^{n\times n}$ and $\mathcal{S}_D \subset \mathbb{R}^{m\times m}$ with prescribed sparsity (based e.g. on powers of $\sum_{k=1}^{r} A_k$ and $\sum_{k=1}^{r} B_k$ or some variation) (Huckle 1999).

Solve alternately

$$\min_{D\in\mathcal{S}_D} \|\tilde{I} - \mathcal{B}D\|_F, \quad \text{and} \quad \min_{C\in\mathcal{S}_C} \|\hat{I} - \mathcal{A}C\|_F.$$

If X is sparse, $\mathcal{P}(X)$ retains some sparsity. Let β_M denote the bandwidth of a matrix M.

Lemma

The Bi-CGSTAB method applied to $\mathcal{M}(X) = Y$ and preconditioned with $\mathcal{P}$ with starting matrix $X_0 = 0$ produces iterates X_j (for a full iteration $j \geq 1$) with bandwidth

$$\beta_{X_j} \leq (2j-1)(\beta_{\mathcal{M}} + \beta_{\mathcal{P}}) + \beta_{\mathcal{P}} + \beta_E$$

where $\beta_{\mathcal{M}} = \max_k\{\beta_{A_k} + \beta_{B_k}\}$ and $\beta_{\mathcal{P}} = \max_s\{\beta_{C_s} + \beta_{D_s}\}$.

RC circuit simulation

Lyapunov-plus-positive equation (Benner and Breiten 2013):

$$AX + XA^T + NXN^T = E, \tag{2}$$

with $m = n = 930$.

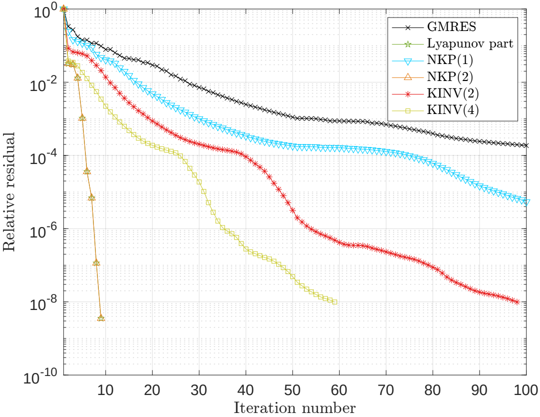


Figure: Convergence history for solving (2) using the (right-preconditioned) GMRES method. The non-preconditioned method converged after 630 iterations.

RC circuit simulation

Timings:

Preconditioner	Setup	GMRES
None	–	26.0 (630)
Lyapunov	–/6.0	12.2 (8)
NKP(1)	0.06/0.02	15.0 (203)
NKP(2)	0.06/5.9	12.2 (8)
KINV(2)	1.4	5.8 (97)
KINV(4)	2.4	5.2 (58)

Table: Timing (in seconds). When writing x/y, x represents the time for computing the SVD representation of the operator and y is the time for computing matrix factorizations (e.g. QZ or LU). The total number of iterations is shown in parenthesis.

Convection-diffusion equation

Consider the PDE (Palitta and Simoncini 2016)

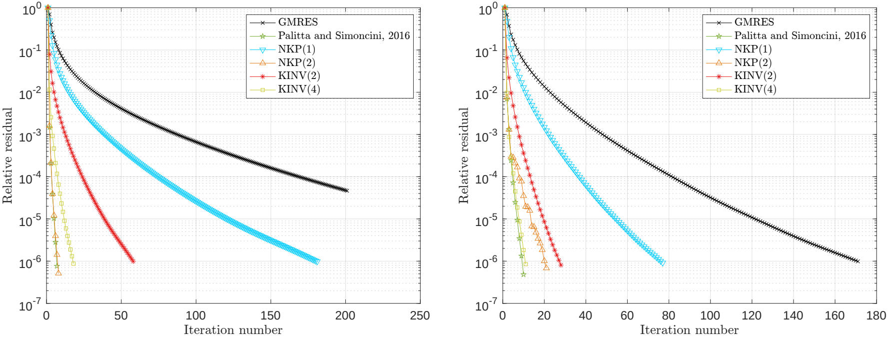
$$-\epsilon \Delta u + \mathbf{w} \cdot \nabla u = f \quad \text{on } \Omega = (0,1)^2.$$

Set suitable boundary conditions, $f = 0$ and

$$\mathbf{w} = \begin{pmatrix} y(1-(2x+1)^2) \\ -2(2x+1)(1-y^2) \end{pmatrix}.$$

Finite difference discretization on the grid $\{x_i\}_{i=1}^n \times \{y_j\}_{j=1}^n$ with $n = 1000$

$$TX + XT^T + (\Phi_1 B) X \Psi_1 + \Phi_2 X (B^T \Psi_2) = F \tag{3}$$



(a) $\epsilon = 1/10$

(b) $\epsilon = 1/30$

Convection-diffusion equation

Timings:

Preconditioner	Setup	$\epsilon = 1/10$	$\epsilon = 1/20$	$\epsilon = 1/30$
None	$-$	103.9 (200*)	105.1 (200*)	73.25 (170)
Palitta and Simoncini	$-$/10.3	10.8 (6)	14.2 (8)	24.4 (9)
NKP(1)	0.04/0.01	94.9 (180)	28.9 (104)	13.8 (76)
NKP(2)	0.04/8.95	13.4 (7)	20.7 (12)	51.1 (20)
KINV(2)	1.04	9.38 (57)	4.41 (35)	2.95 (27)
KINV(4)	1.86	2.04 (17)	1.40 (12)	1.17 (10)

Table: Timing (in seconds). When writing x/y, x represents the time for computing the SVD representation of the operator and y is the time for computing matrix factorizations (e.g. QZ or LU). The total number of iterations is shown in parenthesis, where $*$ indicates that the method did not converge within the maximum number of iterations.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Future work:

- Better theoretical insights on the nearest Kronecker product preconditioner for $q \geq 2$.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Future work:

- Better theoretical insights on the nearest Kronecker product preconditioner for $q \geq 2$.
- When to expect a fast singular value decay of $\mathcal{R}(M^{-1})$?

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Future work:

- Better theoretical insights on the nearest Kronecker product preconditioner for $q \geq 2$.
- When to expect a fast singular value decay of $\mathcal{R}(M^{-1})$?

If interested, please check out [Y. Voet, *Preconditioning techniques for generalized Sylvester matrix equations*] on ArXiv.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Future work:

- Better theoretical insights on the nearest Kronecker product preconditioner for $q \geq 2$.
- When to expect a fast singular value decay of $\mathcal{R}(M^{-1})$?

If interested, please check out [Y. Voet, *Preconditioning techniques for generalized Sylvester matrix equations*] on ArXiv.

Travel support provided by the organizers is kindly acknowledged.

Conclusions

- Algebraic parameter-free preconditioners for multiterm Sylvester equations.
- Based on low Kronecker rank approximations of either the operator or its inverse.
- May complement low-rank solvers for large scale applications.

Future work:

- Better theoretical insights on the nearest Kronecker product preconditioner for $q \geq 2$.
- When to expect a fast singular value decay of $\mathcal{R}(M^{-1})$?

If interested, please check out [Y. Voet, *Preconditioning techniques for generalized Sylvester matrix equations*] on ArXiv.

Travel support provided by the organizers is kindly acknowledged.

Thank you!

References I

[1] Richard H Bartels and George W Stewart. “Algorithm 432 [C2]: Solution of the matrix equation AX+ XB= C [F4]”. In: *Communications of the ACM* 15.9 (1972), pp. 820–826.

[2] Peter Benner and Tobias Breiten. “Low rank methods for a class of generalized Lyapunov equations and related issues”. In: *Numerische Mathematik* 124.3 (2013), pp. 441–470.

[3] Peter Benner and Jens Saak. “Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey”. In: *GAMM-Mitteilungen* 36.1 (2013), pp. 32–52.

[4] Abderrahman Bouhamidi and Khalide Jbilou. “A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications”. In: *Applied Mathematics and Computation* 206.2 (2008), pp. 687–694.

[5] Tobias Damm. “Direct methods and ADI-preconditioned Krylov subspace methods for generalized Lyapunov equations”. In: *Numerical Linear Algebra with Applications* 15.9 (2008), pp. 853–871.

[6] Oliver G Ernst et al. “Efficient solvers for a linear stochastic Galerkin mixed formulation of diffusion problems with random data”. In: *SIAM Journal on Scientific Computing* 31.2 (2009), pp. 1424–1447.

[7] Judith D Gardiner et al. “Solution of the Sylvester matrix equation AXB T+ CXD T= E”. In: *ACM Transactions on Mathematical Software (TOMS)* 18.2 (1992), pp. 223–231.

[8] Lars Grasedyck. “Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure”. In: *Computing* 72 (2004), pp. 247–265.

[9] Marcus J Grote and Thomas Huckle. “Parallel preconditioning with sparse approximate inverses”. In: *SIAM Journal on Scientific Computing* 18.3 (1997), pp. 838–853.

[10] Yue Hao and Valeria Simoncini. “The Sherman–Morrison–Woodbury formula for generalized linear matrix equations and applications”. In: *Numerical linear algebra with applications* 28.5 (2021), e2384.

References II

[11] Marlis Hochbruck and Gerhard Starke. “Preconditioned Krylov subspace methods for Lyapunov matrix equations”. In: *SIAM Journal on Matrix Analysis and Applications* 16.1 (1995), pp. 156–171.

[12] Thomas Huckle. “Approximate sparsity patterns for the inverse of a matrix and preconditioning”. In: *Applied numerical mathematics* 30.2-3 (1999), pp. 291–303.

[13] Elias Jarlebring et al. “Krylov methods for low-rank commuting generalized Sylvester equations”. In: *Numerical Linear Algebra with Applications* 25.6 (2018), e2176.

[14] Khalide Jbilou, Abderrahim Messaoudi, and Hassane Sadok. “Global FOM and GMRES algorithms for matrix equations”. In: *Applied Numerical Mathematics* 31.1 (1999), pp. 49–63.

[15] Isak Jonsson and Bo Kågström. “Recursive blocked algorithms for solving triangular systems—Part I: One-sided and coupled Sylvester-type matrix equations”. In: *ACM Transactions on Mathematical Software (TOMS)* 28.4 (2002), pp. 392–415.

[16] Isak Jonsson and Bo Kågström. “Recursive blocked algorithms for solving triangular systems—Part II: Two-sided and generalized Sylvester and Lyapunov matrix equations”. In: *ACM Transactions on Mathematical Software (TOMS)* 28.4 (2002), pp. 416–435.

[17] Daniel Kressner and Petar Sirković. “Truncated low-rank methods for solving general linear matrix equations”. In: *Numerical Linear Algebra with Applications* 22.3 (2015), pp. 564–583.

[18] Daniel Kressner and Christine Tobler. “Krylov subspace methods for linear systems with tensor product structure”. In: *SIAM journal on matrix analysis and applications* 31.4 (2010), pp. 1688–1714.

[19] Patrick Kürschner et al. “Greedy low-rank algorithm for spatial connectome regression”. In: *The Journal of Mathematical Neuroscience* 9.1 (2019), pp. 1–22.

[20] Angelos Mantzaflaris et al. “Low rank tensor methods in Galerkin-based isogeometric analysis”. In: *Computer Methods in Applied Mechanics and Engineering* 316 (2017), pp. 1062–1085.

[21] Stefano Massei, Davide Palitta, and Leonardo Robol. “Solving rank-structured Sylvester and Lyapunov equations”. In: *SIAM journal on matrix analysis and applications* 39.4 (2018), pp. 1564–1590.

References III

[22] Davide Palitta and Valeria Simoncini. “Matrix-equation-based strategies for convection–diffusion equations”. In: *BIT Numerical Mathematics* 56 (2016), pp. 751–776.

[23] Davide Palitta and Valeria Simoncini. “Numerical methods for large-scale Lyapunov equations with symmetric banded data”. In: *SIAM Journal on Scientific Computing* 40.5 (2018), A3581–A3608.

[24] Emil Ringh et al. “Sylvester-based preconditioning for the waveguide eigenvalue problem”. In: *Linear Algebra and its Applications* 542 (2018), pp. 441–463.

[25] Felix Scholz, Angelos Mantzaflaris, and Bert Jüttler. “Partial tensor decomposition for decoupling isogeometric Galerkin discretizations”. In: *Computer Methods in Applied Mechanics and Engineering* 336 (2018), pp. 485–506.

[26] Stephen D Shank, Valeria Simoncini, and Daniel B Szyld. “Efficient low-rank solution of generalized Lyapunov equations”. In: *Numerische Mathematik* 134.2 (2016), pp. 327–342.

[27] Valeria Simoncini. “Computational methods for linear matrix equations”. In: *SIAM REVIEW* 58.3 (2016), pp. 377–441.

[28] Elisabeth Ullmann. “A Kronecker product preconditioner for stochastic Galerkin finite element discretizations”. In: *SIAM Journal on Scientific Computing* 32.2 (2010), pp. 923–946.

[29] Charles F Van Loan and Nikos Pitsianis. “Approximation with Kronecker products”. In: *Linear algebra for large scale and real-time applications*. Springer, 1993, pp. 293–314.

[30] Eugene L Wachspress. “Iterative solution of the Lyapunov matrix equation”. In: *Applied Mathematics Letters* 1.1 (1988), pp. 87–90.