MAX PLANCK INSTITUTE
FOR DYNAMICS OF COMPLEX
TECHNICAL SYSTEMS
MAGDEBURG

CSC COMPUTATIONAL METHODS IN
SYSTEMS AND CONTROL THEORY

Numerical Solution of Matrix Equations Arising in DSGE Models

Peter Benner

Centre of Policy Studies (CoPS)
Victoria University, Melbourne
February 8, 2019

The Max Planck Institute (MPI) in Magdeburg

The Max Planck **Society**

- is dedicated to fundamental research;
- operates 84 institutes — 79 in Germany, 2 in Italy (Rome, Florence), 1 each in The Netherlands, Luxembourg, US;
- has $\sim$ 23,000 employees;
- had 18 Nobel Laureates since 1948.

"The first MPI in engineering. . . "

- founded 1998
- 4 departments (directors)
- 10 research groups
- budget $\sim$ 15 Mio. EUR
- $\sim$ 230 employees
- $\sim$ 160 scientific staff,
- doing research in
  - biotechnology
  - chemical engineering
  - process engineering
  - energy conversion
  - applied math
  - HPC

Work on Numerical Methods for Matrix Equations

Selected publications since 1993

- G. S. Ammar, P. Benner, and V. Mehrmann, *A multishift algorithm for the numerical solution of algebraic Riccati equations*, Electron. Trans. Numer. Anal., 1 (1993), pp. 33–48.
- P. Benner and R. Byers, *An exact line search method for solving generalized continuous-time algebraic Riccati equations*, IEEE Trans. Autom. Control, 43 (1998), pp. 101–107.
- P. Benner and E. S. Quintana-Ortí, *Solving stable generalized Lyapunov equations with the matrix sign function*, Numer. Algorithms, 20 (1999), pp. 75–100.
- P. Benner, *Factorized solution of Sylvester equations with applications in control*, in Proc. Intl. Symp. Math. Theory Networks and Syst. MTNS 2004, 2004.
- P. Benner, E. S. Quintana-Ortí, and G. Quintana-Ortí, *Solving stable Sylvester equations via rational iterative schemes*, J. Sci. Comp., 28 (2006), pp. 51–83.
- P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large-scale Lyapunov equations, Riccati equations, and linear-quadratic optimal control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777.
- P. Benner, R.-C. Li, and N. Truhar, *On the ADI method for Sylvester equations*, J. Comput. Appl. Math., 233 (2009), pp. 1035–1045.
- P. Benner and H. Fassbender, *On the numerical solution of large-scale sparse discrete-time Riccati equations*, Adv. Comput. Math., 35 (2011), pp. 119–147.
- P. Benner and J. Saak, *Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey*, GAMM Mitteilungen, 36 (2013), pp. 32–52.
- P. Benner and P. Kürschner, *Computing real low-rank solutions of Sylvester equations by the factored ADI method*, Comput. Math. Appl., 67 (2014), pp. 1656–1672.
- P. Benner, P. Kürschner, and J. Saak, *Low-rank Newton-ADI methods for large nonsymmetric algebraic Riccati equations*, J. Frankl. Inst., 353 (2016), pp. 1147–1167.
- P. Benner, Z. Bujanović, P. Kürschner, and J. Saak, *RADI: A low-rank ADI-type algorithm for large scale algebraic Riccati equations*, Numer. Math., 138 (2018), pp. 301–330.

Problem: Find $X \in \mathbb{R}^{n\times n}$ such that

$$AX^2 + BX + C = 0, \qquad A, B, C \in \mathbb{R}^{n\times n}.$$

Unilateral quadratic matrix equations (UQME) arise in

- solving large-scale Dynamic Stochastic General Equilibrium (DSGE) models;
- quasi-birth-death processes;
- quadratic eigenvalue problems.

The Unilateral Quadratic Matrix Equation

Problem: Find $X \in \mathbb{R}^{n\times n}$ such that

$$AX^2 + BX + C = 0, \qquad A, B, C \in \mathbb{R}^{n\times n}.$$

Unilateral quadratic matrix equations (UQME) arise in

- solving large-scale Dynamic Stochastic General Equilibrium (DSGE) models;
- quasi-birth-death processes;
- quadratic eigenvalue problems.

Explicit formula for solution of scalar quadratic equations does not generalize to UQME, except in special situations: e.g., if $A = I_n$ *and* B, C commute, then

$$X = -\frac{1}{2}B \pm (B^2 - 4C)^{\frac{1}{2}} \qquad \text{if } B^2 - 4C \text{ has a matrix root.}$$

$\rightsquigarrow$ need numerical solution schemes!

N.J. Higham and H.M. Kim. *Numerical analysis of a quadratic matrix equation*. IMA Journal of Numerical Analysis 20(4):499–519, 2000.

The Schur Method for UQME

Theorem ([Higham/Kim 2000])

Given $A, B, C \in \mathbb{R}^{n\times n}$, consider the UQME $AX^2 + BX + C = 0.$

a) *$X \in \mathbb{R}^{n\times n}$ solves the UQME if and only if*

$$\underbrace{\begin{bmatrix} 0 & I_n \\ -C & -B \end{bmatrix}}_{=:F} \begin{bmatrix} I_n \\ X \end{bmatrix} = \underbrace{\begin{bmatrix} I_n & 0 \\ 0 & A \end{bmatrix}}_{=:G} \begin{bmatrix} I_n \\ X \end{bmatrix} X.$$

In mathematical terms, X defines an n-dimensional graph subspace of the matrix pencil $F - \lambda G$, corresponding to the eigenvalues $\Lambda(X) \subset \Lambda(F, G)$.

N.J. Higham and H.M. Kim. *Numerical analysis of a quadratic matrix equation.* IMA Journal of Numerical Analysis 20(4):499–519, 2000.

The Schur Method for UQME

Theorem ([Higham/Kim 2000])

Given $A, B, C \in \mathbb{R}^{n\times n}$, consider the UQME $AX^2 + BX + C = 0.$

b) *Let*

$$F = Q\Sigma Z^T, \quad G = Q\Phi Z^T, \quad \text{with } \Sigma, \Phi = \left[\diagdown \right],$$

where $Q, Z \in \mathbb{R}^{n\times n}$ are orthogonal ($Q^TQ = Z^TZ = I_n$), be the generalized Schur decomposition of $F - \lambda G$.

Then every solution $X \in \mathbb{R}^{n\times n}$ of the UQME has the form

$$X = Z_{21}Z_{11}^{-1} = Q_{11}\Sigma_{11}\Phi_{11}^{-1}Q_{11}^{-1},$$

with M_{ij} denoting the blocks in a uniform 2×2-block partitioning of M.

Note: different solutions X correspond to different orderings of the eigenvalues of $F - \lambda G$, i.e., different orderings of the diagonal elements of Σ, Φ.

N.J. Higham and H.M. Kim. *Numerical analysis of a quadratic matrix equation*. IMA Journal of Numerical Analysis 20(4):499–519, 2000.

The Schur Method for UQME

The characterization of solutions leads directly to a numerical solution method for the UQME:

1. Form the $2n \times 2n$-block matrices F, G.
2. Compute the generalized Schur decomposition of $F - \lambda G$ (e.g., using the QZ algorithm in MATLAB via qz).
3. Re-order the diagonal elements/eigenvalues in the generalized Schur form as needed.
4. Solve $XZ_{11} = Z_{21}$.

Remark

As

$$\mathrm{cond}_2(Z_{11}) \leq 1 + \|X\|_2,$$

scaling $X \to X/\rho$ with $\rho \approx X$ may improve the accuracy of the solution.

+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.
+ Thus, Schur method can be easily implemented, e.g., in MATLAB.

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.
+ Thus, Schur method can be easily implemented, e.g., in MATLAB.
– Structure of matrices from DGSE models cannot be exploited, except, maybe, for initial step (reduction to Hessenberg-triangular form).

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

- \+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.
- \+ Thus, Schur method can be easily implemented, e.g., in MATLAB.
- – Structure of matrices from DGSE models cannot be exploited, except, maybe, for initial step (reduction to Hessenberg-triangular form).
- – Data access pattern and data dependencies make the QZ algorithm a serial, communication-bound algorithm.

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.
+ Thus, Schur method can be easily implemented, e.g., in MATLAB.
– Structure of matrices from DGSE models cannot be exploited, except, maybe, for initial step (reduction to Hessenberg-triangular form).
– Data access pattern and data dependencies make the QZ algorithm a serial, communication-bound algorithm.
– Therefore, QZ algorithm is notoriously difficult to parallelize. Hence, it is not efficient on modern multicore architectures.

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

- \+ QZ algorithm [Moler/Stewart 1973] is numerically backward stable and is implemented in LAPACK, the backbone of Intel's MKL, the MATLAB Linear Algebra kernel, etc.
- \+ Thus, Schur method can be easily implemented, e.g., in MATLAB.
- – Structure of matrices from DGSE models cannot be exploited, except, maybe, for initial step (reduction to Hessenberg-triangular form).
- – Data access pattern and data dependencies make the QZ algorithm a serial, communication-bound algorithm.
- – Therefore, QZ algorithm is notoriously difficult to parallelize. Hence, it is not efficient on modern multicore architectures.
- \+ Recent performance improvement using block variant of QZ algorithm [Kågström/Kressner 2008], not yet included in LAPACK.

B. Kågström and D. Kressner. *Multishift variants of the QZ algorithm with aggressive early deflation*. SIAM Journal on Matrix Analysis and Applications 29(1):199–227, 2008.

- Uniform $(-1, 1)$ random matrices
- Compute generalized Schur decomposition only
- 2x8 core Intel Xeon Silver 4110, 192 GB RAM, Intel MKL 2018.1

Dim. n	LAPACK	KKQZ	Speed-up
5 000	873s	482s	1.81
10 000	9 630s	5 647s	1.71
15 000	27 623s	17 195s	1.61
20 000	77 935s	48 189s	1.62
25 000	141 207s	86 009s	1.64

Basic Idea: for solving UQME, only basis of subspace spanned by $\begin{bmatrix} I \\ X \end{bmatrix}$ is needed, that is, a separation of the spectrum into 2 clusters rather than a separation of all eigenvalues as in the generalized Schur decompostion.

Basic Idea: for solving UQME, only basis of subspace spanned by $\begin{bmatrix} I \\ X \end{bmatrix}$ is needed, that is, a separation of the spectrum into 2 clusters rather than a separation of all eigenvalues as in the generalized Schur decompostion.

Therefore, need $Q, Z \in \mathbb{R}^{n\times n}$ (orthogonal) such that

$$Q(F - \lambda G)Z \quad = \quad \begin{bmatrix} F_{11} & F_{12} \\ & F_{22} \end{bmatrix} - \lambda \begin{bmatrix} G_{11} & G_{12} \\ & G_{22} \end{bmatrix}$$

A Sign Function Approach to Solving UQMEs

Basic Idea: for solving UQME, only basis of subspace spanned by $\begin{bmatrix} I \\ X \end{bmatrix}$ is needed, that is, a separation of the spectrum into 2 clusters rather than a separation of all eigenvalues as in the generalized Schur decompostion.

Therefore, need $Q, Z \in \mathbb{R}^{n \times n}$ (orthogonal) such that

$$\begin{aligned} Q(F - \lambda G)Z &= \begin{bmatrix} F_{11} & F_{12} \\ & F_{22} \end{bmatrix} - \lambda \begin{bmatrix} G_{11} & G_{12} \\ & G_{22} \end{bmatrix} \\ &= \begin{bmatrix} \square & \square \\ & \square \end{bmatrix} - \lambda \begin{bmatrix} \square & \square \\ & \square \end{bmatrix}, \end{aligned}$$

i.e., block-triangular decomposition!

A Sign Function Approach to Solving UQMEs

Basic Idea: for solving UQME, only basis of subspace spanned by $\begin{bmatrix} I \\ X \end{bmatrix}$ is needed, that is, a separation of the spectrum into 2 clusters rather than a separation of all eigenvalues as in the generalized Schur decompostion.

Therefore, need $Q, Z \in \mathbb{R}^{n \times n}$ (orthogonal) such that

$$\begin{aligned} Q(F - \lambda G)Z &= \begin{bmatrix} F_{11} & F_{12} \\ & F_{22} \end{bmatrix} - \lambda \begin{bmatrix} G_{11} & G_{12} \\ & G_{22} \end{bmatrix} \\ &= \begin{bmatrix} \square & \square \\ & \square \end{bmatrix} - \lambda \begin{bmatrix} \square & \square \\ & \square \end{bmatrix}, \end{aligned}$$

i.e., block-triangular decomposition!

Then $X = Z_{21} Z_{11}^{-1}$ and Q is not even needed!

A Sign Function Approach to Solving UQMEs

Basic Idea: for solving UQME, only basis of subspace spanned by $\begin{bmatrix} I \\ X \end{bmatrix}$ is needed, that is, a separation of the spectrum into 2 clusters rather than a separation of all eigenvalues as in the generalized Schur decompostion.

Therefore, need $Q, Z \in \mathbb{R}^{n \times n}$ (orthogonal) such that

$$
\begin{aligned}
Q(F - \lambda G)Z &= \begin{bmatrix} F_{11} & F_{12} \\ & F_{22} \end{bmatrix} - \lambda \begin{bmatrix} G_{11} & G_{12} \\ & G_{22} \end{bmatrix} \\
&= \begin{bmatrix} \square & \square \\ & \square \end{bmatrix} - \lambda \begin{bmatrix} \square & \square \\ & \square \end{bmatrix},
\end{aligned}
$$

i.e., block-triangular decomposition!

Then $X = Z_{21} Z_{11}^{-1}$ and Q is not even needed!

This can be computed by spectral projection methods like (generalized) sign and disk function methods.

[Roberts 1971]

Definition (Matrix sign function)

Given $Z \in \mathbb{R}^{n \times n}$ with k / $n-k$ eigenvalues in the open left / right half of the complex plane and Jordan decomposition

$$Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1},$$

where the Jordan blocks corresponding to the eigenvalues

- in the open left half plane are collected in $J^- \in \mathbb{C}^{k \times k}$,
- in the open right half plane are collected in $J^+ \in \mathbb{C}^{n-k \times n-k}$.

Then

$$\operatorname{sign}(Z) := S \begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1},$$

and $\mathrm{range}\,(I_n - \operatorname{sign}(Z))$ is the Z-invariant subspace corresponding to J^-.

Computing the matrix sign function

Applying Newton's method to $F(Z) = Z^2 - I$ with $Z_0 := Z$ yields

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2c_j}(Z_j + c_j^2 Z_j^{-1}), \qquad j = 0, 1, \ldots,$$

with $\lim_{j\to\infty} Z_j = \operatorname{sign}(Z)$ and where c_j is a scaling factor accelerating convergence.

Stable Z-invariant subspace can be computed using pivoted QR decomposition of SVD applied to $I - \operatorname{sign}(Z)$.

Computing the matrix sign function

Applying Newton's method to $F(Z) = Z^2 - I$ with $Z_0 := Z$ yields

$$Z_0 \leftarrow Z, \quad Z_{j+1} \leftarrow \frac{1}{2c_j}(Z_j + c_j^2 Z_j^{-1}), \qquad j = 0, 1, \ldots,$$

with $\lim_{j\to\infty} Z_j = \operatorname{sign}(Z)$ and where c_j is a scaling factor accelerating convergence.

Stable Z-invariant subspace can be computed using pivoted QR decomposition of SVD applied to $I - \operatorname{sign}(Z)$.

Application to matrix pencils $F - \lambda G$**:** apply sign function iteration implicitly to $Z := G^{-1}F$, leading to

$$F_0 \leftarrow F, \quad F_{j+1} \leftarrow \frac{1}{2c_j}(F_j + c_j^2 G F_j^{-1} G), \qquad j = 0, 1, \ldots,$$

and $\mathrm{range}\,(\lim_{j\to\infty} F_j - G)$ provides stable "deflating" subspace.

Note: Usually, $c_j = \left(\frac{|\det(Z_j)|}{|\det(Y)|}\right)^{\frac{1}{n}}$.

Algorithm 1 Generalized Sign Function Method

Input: A matrix pencil $F - \lambda G$, $F, G \in \mathbb{R}^{n \times n}$ with no eigenvalues on the imaginary axis.

Output: generalized sign function $F_\infty - \lambda G$.

1: Set $F_0 = F$, $g = |\det G|^{\frac{1}{n}}$.

2: **for** $j = 0, 1, \ldots$ until convergence **do**

3: $\quad F_j = \Pi^T LU$ $\{$*LU factorization: L/U lower/upper triangular,* Π *permutation*$\}$

4: $\quad c_j = \left(\prod_{k=1}^{n} |u_{kk}|^{\frac{1}{n}}\right) / g$.

5: $\quad$ Solve $LW = \Pi G$ by forward substitution.

6: $\quad$ Solve $UX = W$ by backward substitution.

7: $\quad F_{j+1} = \frac{1}{2c_j} F_j + \frac{c_j}{2} GX$.

8: **end for**

Set-up:

- Uniform $(-1, 1)$ random matrices
- 2x8 core Intel Xeon Silver 4110, 192 GB RAM, Intel MKL 2018.1
- sign function run to compute generalized Schur form (not block-triangular form!)

Dim. m	LAPACK	KKQZ	sign fct.	Speed-up	Error [1]
5 000	873s	482s	123s	3.91	$1.42 \cdot 10^{-11}$
10 000	9 630s	5 647s	645s	8.76	$2.69 \cdot 10^{-10}$
15 000	27 623s	17 195s	2 217s	7.75	$8.59 \cdot 10^{-13}$
20 000	77 935s	48 189s	4 838s	9.96	$7.39 \cdot 10^{-11}$
25 000	141 207s	86 009s	8 213s	10.47	$4.08 \cdot 10^{-11}$

[1] $\max\{\frac{||A-QA_sZ^T||}{||A||}, \frac{||B-QB_sZ^T||}{||B||}\}$

A Sign Function Approach to Solving UQMEs

- For DSGE models, want solution $X = \begin{bmatrix} \overline{h}_x & 0 \\ \overline{g}_x & 0 \end{bmatrix}$ with $\rho(\overline{h}_x) < 1$.

- For DSGE models, want solution $X = \begin{bmatrix} \overline{h}_x & 0 \\ \overline{g}_x & 0 \end{bmatrix}$ with $\rho(\overline{h}_x) < 1$.
- As $\Lambda(X) = \Lambda(\overline{h}_x) \cup \{0\}$, need solution X with $\rho(X) < 1$; i.e., invariant "deflating" subspace corresponding to eigenvalues inside unit circle.

- For DSGE models, want solution $X = \begin{bmatrix} \overline{h}_x & 0 \\ \overline{g}_x & 0 \end{bmatrix}$ with $\rho(\overline{h}_x) < 1$.
- As $\Lambda(X) = \Lambda(\overline{h}_x) \cup \{0\}$, need solution X with $\rho(X) < 1$; i.e., invariant "deflating" subspace corresponding to eigenvalues inside unit circle.
- Natural splitting of eigenvalues computed by (generalized) sign function is w.r.t. imaginary axis. Thus, apply sign function method to

$$\tilde{F} \leftarrow F - G = \begin{bmatrix} -I_n & I_n \\ -C & A - B \end{bmatrix}, \quad \tilde{G} \leftarrow F + G = \begin{bmatrix} I_n & I_n \\ -C & -(A + B) \end{bmatrix}.$$

A Sign Function Approach to Solving UQMEs

- For DSGE models, want solution $X = \begin{bmatrix} \bar{h}_x & 0 \\ \bar{g}_x & 0 \end{bmatrix}$ with $\rho(\bar{h}_x) < 1$.
- As $\Lambda(X) = \Lambda(\bar{h}_x) \cup \{0\}$, need solution X with $\rho(X) < 1$; i.e., invariant "deflating" subspace corresponding to eigenvalues inside unit circle.
- Natural splitting of eigenvalues computed by (generalized) sign function is w.r.t. imaginary axis. Thus, apply sign function method to

$$\tilde{F} \leftarrow F - G = \begin{bmatrix} -I_n & I_n \\ -C & A - B \end{bmatrix}, \quad \tilde{G} \leftarrow F + G = \begin{bmatrix} I_n & I_n \\ -C & -(A + B) \end{bmatrix}.$$

- Potential computational savings: matrix product with G:

$$\begin{aligned} \tilde{G}M &= \begin{bmatrix} I_n & I_n \\ -C & -(A + B) \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \\ &= \begin{bmatrix} M_{11} + M_{21} & M_{12} + M_{22} \\ [-N_0, 0]M_{11} - (A + B)M_{21} & [-N_0, 0]M_{12} - (A + B)M_{22} \end{bmatrix} \end{aligned}$$

Implementing this efficiently requires $4n^2(n + n_x)$ flops instead of $16n^3$; this should save $\approx 50\%$ of the operations/time per iteration!

A Sign Function Approach to Solving UQMEs

- For DSGE models, want solution $X = \begin{bmatrix} \bar{h}_x & 0 \\ \bar{g}_x & 0 \end{bmatrix}$ with $\rho(\bar{h}_x) < 1$.
- As $\Lambda(X) = \Lambda(\bar{h}_x) \cup \{0\}$, need solution X with $\rho(X) < 1$; i.e., invariant "deflating" subspace corresponding to eigenvalues inside unit circle.
- Natural splitting of eigenvalues computed by (generalized) sign function is w.r.t. imaginary axis. Thus, apply sign function method to

$$\tilde{F} \leftarrow F - G = \begin{bmatrix} -I_n & I_n \\ -C & A - B \end{bmatrix}, \quad \tilde{G} \leftarrow F + G = \begin{bmatrix} I_n & I_n \\ -C & -(A+B) \end{bmatrix}.$$

- Potential computational savings: matrix product with G:

$$\begin{aligned} \tilde{G}M &= \begin{bmatrix} I_n & I_n \\ -C & -(A+B) \end{bmatrix} \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \\ &= \begin{bmatrix} M_{11} + M_{21} & M_{12} + M_{22} \\ [-N_0, 0]M_{11} - (A+B)M_{21} & [-N_0, 0]M_{12} - (A+B)M_{22} \end{bmatrix} \end{aligned}$$

  Implementing this efficiently requires $4n^2(n + n_x)$ flops instead of $16n^3$; this should save $\approx 50\%$ of the operations/time per iteration!
- No re-ordering needed as in Schur method!

- (Quasi-)Newton's method for $0 = Q(X) = AX^2 + BX + C$:

- (Quasi-)Newton's method for $0 = Q(X) = AX^2 + BX + C$:
  1. Solve Sylvester equation

$$A\Delta_k X_k + (AX_k + B)\Delta_k = -Q(X_k)$$

     for Δ_k.

- (Quasi-)Newton's method for $0 = Q(X) = AX^2 + BX + C$:
  1. Solve Sylvester equation

     $$A\Delta_k X_k + (AX_k + B)\Delta_k = -Q(X_k)$$

     for Δ_k.
  2. Set $X_{k+1} = X_k + t_k\Delta_k$. (Step length $t_k = 1$ for Newton's method.)

- (Quasi-)Newton's method for $0 = Q(X) = AX^2 + BX + C$:
  1. Solve Sylvester equation

$$A\Delta_k X_k + (AX_k + B)\Delta_k = -Q(X_k)$$

     for Δ_k.
  2. Set $X_{k+1} = X_k + t_k \Delta_k$. (Step length $t_k = 1$ for Newton's method.)

- Functional iterations, e.g., Bernoulli iteration

$$X_{k+1} = -A^{-1}(B + CX_k^{-1}).$$

Not applicable for DSGE models, as A is singular.

- (Quasi-)Newton's method for $0 = Q(X) = AX^2 + BX + C$:
  1. Solve Sylvester equation
  $$A\Delta_k X_k + (AX_k + B)\Delta_k = -Q(X_k)$$
  for Δ_k.
  2. Set $X_{k+1} = X_k + t_k\Delta_k$. (Step length $t_k = 1$ for Newton's method.)
- Functional iterations, e.g., Bernoulli iteration
  $$X_{k+1} = -A^{-1}(B + CX_k^{-1}).$$
  Not applicable for DSGE models, as A is singular.
- Cyclic reduction — numerical stability is not guaranteed.

Outline

The following linear matrix equations occur in the recursive solution of DSGE models using perturbation methods [HARDING '19]:

Generalized Sylvester Equation

$$A_0 Y + A Y \overline{C}_z + P_0 + P_1 \overline{C}_z = 0,$$

where

- $A, A_0 \in \mathbb{R}^{n \times n}$, $\overline{C}_z \in \mathbb{R}^{n_z \times n_z}$, $P_0, P_1 \in \mathbb{R}^{n \times n_z}$,
- and $Y \in \mathbb{R}^{n \times n_z}$ is the unknown matrix.

The following linear matrix equations occur in the recursive solution of DSGE models using perturbation methods [Harding '19]:

Generalized Sylvester Equation

$$A_0 Y + A Y \overline{C}_z + P_0 + P_1 \overline{C}_z = 0,$$

where

- $A, A_0 \in \mathbb{R}^{n \times n}$, $\overline{C}_z \in \mathbb{R}^{n_z \times n_z}$, $P_0, P_1 \in \mathbb{R}^{n \times n_z}$,
- and $Y \in \mathbb{R}^{n \times n_z}$ is the unknown matrix.

This can be transformed to a smaller Sylvester equation:

Sylvester Equation

$$TZ + Z\overline{C}_z + W = 0, \qquad T, W \in \mathbb{R}^{n_z \times n_z},$$

where $Z \in \mathbb{R}^{n_z \times n_z}$ is the unknown matrix.

Sylvester equation

James Joseph Sylvester
(September 3, 1814 – March 15, 1897)

$$AX + XB = C.$$

Sylvester equation

James Joseph Sylvester
(September 3, 1814 – March 15, 1897)

$$AX + XB = C.$$

Lyapunov equation

Alexander Michailowitsch Ljapunow
(June 6, 1857 – November 3, 1918)

$$AX + XA^T = C, \quad C = C^T.$$

Generalized Sylvester equation:

$$AXD + EXB = C.$$

Generalized Sylvester equation:

$$AXD + EXB = C.$$

Generalized Lyapunov equation:

$$AXE^T + EXA^T = C, \quad C = C^T.$$

Generalized Sylvester equation:

$$AXD + EXB = C.$$

Generalized Lyapunov equation:

$$AXE^T + EXA^T = C, \quad C = C^T.$$

Stein equation:

$$X - AXB = C.$$

Generalized Sylvester equation:

$$AXD + EXB = C.$$

Generalized Lyapunov equation:

$$AXE^T + EXA^T = C, \quad C = C^T.$$

Stein equation:

$$X - AXB = C.$$

(Generalized) discrete Lyapunov/Stein equation:

$$EXE^T - AXA^T = C, \quad C = C^T.$$

Generalized Sylvester equation:

$$AXD + EXB = C.$$

Generalized Lyapunov equation:

$$AXE^T + EXA^T = C, \quad C = C^T.$$

Stein equation:

$$X - AXB = C.$$

(Generalized) discrete Lyapunov/Stein equation:

$$EXE^T - AXA^T = C, \quad C = C^T.$$

Note:

- Consider only regular cases, having a unique solution!
- Solutions of symmetric cases are symmetric, $X = X^T \in \mathbb{R}^{n \times n}$; otherwise, $X \in \mathbb{R}^{n \times \ell}$ with $n \neq \ell$ in general.

Bilinear Lyapunov equation/Lyapunov-plus-positive equation:

$$AX + XA^T + \sum_{k=1}^{m} N_k X N_k^T = C, \quad C = C^T.$$

Bilinear Lyapunov equation/Lyapunov-plus-positive equation:

$$AX + XA^T + \sum_{k=1}^{m} N_k X N_k^T = C, \quad C = C^T.$$

Bilinear Sylvester equation:

$$AX + XB + \sum_{k=1}^{m} N_k X M_k = C.$$

Bilinear Lyapunov equation/Lyapunov-plus-positive equation:

$$AX + XA^T + \sum_{k=1}^{m} N_k X N_k^T = C, \quad C = C^T.$$

Bilinear Sylvester equation:

$$AX + XB + \sum_{k=1}^{m} N_k X M_k = C.$$

(Generalized) discrete bilinear Lyapunov/Stein-minus-positive eq.:

$$EXE^T - AXA^T - \sum_{k=1}^{m} N_k X N_k^T = C, \quad C = C^T.$$

Note: Again consider only regular cases, symmetric equations have symmetric solutions.

Exemplarily, consider the generalized Sylvester equation

$$AXD + EXB = C. \tag{1}$$

Exemplarily, consider the generalized Sylvester equation

$$AXD + EXB = C. \tag{1}$$

Vectorization (using Kronecker product) $\rightsquigarrow$ representation as linear system:

$$\big(\underbrace{D^T \otimes A + B^T \otimes E}_{=:\mathcal{A}}\big)\underbrace{\mathrm{vec}(X)}_{=:x} = \underbrace{\mathrm{vec}(C)}_{=:c} \quad\Longleftrightarrow\quad \mathcal{A}x = c.$$

Exemplarily, consider the generalized Sylvester equation

$$AXD + EXB = C. \tag{1}$$

Vectorization (using Kronecker product) $\rightsquigarrow$ representation as linear system:

$$\big(\underbrace{D^T \otimes A + B^T \otimes E}_{=:\mathcal{A}}\big)\underbrace{\mathrm{vec}(X)}_{=:x} = \underbrace{\mathrm{vec}(C)}_{=:c} \qquad \Longleftrightarrow \qquad \mathcal{A}x = c.$$

$\Longrightarrow$ "(1) has a unique solution $\Longleftrightarrow$ $\mathcal{A}$ is nonsingular"

Exemplarily, consider the generalized Sylvester equation

$$AXD + EXB = C. \tag{1}$$

Vectorization (using Kronecker product) $\rightsquigarrow$ representation as linear system:

$$\big(\underbrace{D^T \otimes A + B^T \otimes E}_{=:\mathcal{A}}\big)\underbrace{\mathrm{vec}(X)}_{=:x} = \underbrace{\mathrm{vec}(C)}_{=:c} \quad \Longleftrightarrow \quad \mathcal{A}x = c.$$

$\Longrightarrow$ "(1) has a unique solution $\Longleftrightarrow$ $\mathcal{A}$ is nonsingular"

Lemma

$$\Lambda(\mathcal{A}) = \{\alpha_j + \beta_k \mid \alpha_j \in \Lambda(A, E),\ \beta_k \in \Lambda(B, D)\}.$$

Hence, (1) has unique solution $\Longleftrightarrow$ $\Lambda(A, E) \cap -\Lambda(B, D) = \emptyset$.

Exemplarily, consider the generalized Sylvester equation

$$AXD + EXB = C. \tag{1}$$

Vectorization (using Kronecker product) $\rightsquigarrow$ representation as linear system:

$$\underbrace{\left(D^T \otimes A + B^T \otimes E\right)}_{=:\mathcal{A}} \underbrace{\operatorname{vec}(X)}_{=:x} = \underbrace{\operatorname{vec}(C)}_{=:c} \quad \iff \quad \mathcal{A}x = c.$$

$\Longrightarrow$ "(1) has a unique solution $\iff$ $\mathcal{A}$ is nonsingular"

Lemma

$$\Lambda(\mathcal{A}) = \{\alpha_j + \beta_k \mid \alpha_j \in \Lambda(A, E),\ \beta_k \in \Lambda(B, D)\}.$$

Hence, (1) has unique solution $\iff \Lambda(A, E) \cap -\Lambda(B, D) = \emptyset$.

Example: Lyapunov equation $AX + XA^T = C$ has unique solution

$$\iff \nexists\, \mu \in \mathbb{C} \,:\, \pm\mu \in \Lambda(A).$$

Theorem (Lyapunov 1892)

Let $A \in \mathbb{R}^{n \times n}$ and consider the Lyapunov operator $\mathcal{L} : X \rightarrow AX + XA^T$. Then the following are equivalent:

(a) $\forall Y > 0$: $\exists X > 0$: $\mathcal{L}(X) = -Y$,

(b) $\exists Y > 0$: $\exists X > 0$: $\mathcal{L}(X) = -Y$,

(c) $\Lambda(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re z < 0\}$, *i.e., A is (asymptotically) stable or Hurwitz.*

A. M. Lyapunov. *The General Problem of the Stability of Motion* (in Russian). Doctoral dissertation, Univ. Kharkov 1892. English translation: Stability of Motion, Academic Press, New-York & London, 1966.

P. Lancaster, M. Tismenetsky. *The Theory of Matrices* (2nd edition). Academic Press, Orlando, FL, 1985. [Chapter 13]

Theorem (Lyapunov 1892)

Let $A \in \mathbb{R}^{n\times n}$ and consider the Lyapunov operator $\mathcal{L} : X \to AX + XA^T$. Then the following are equivalent:

(a) $\forall Y > 0$: $\exists X > 0$: $\mathcal{L}(X) = -Y$,

(b) $\exists Y > 0$: $\exists X > 0$: $\mathcal{L}(X) = -Y$,

(c) $\Lambda(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re z < 0\}$, *i.e., A is (asymptotically) stable or Hurwitz.*

The proof (c) $\Rightarrow$ (a) is trivial from the necessary and sufficient condition for existence and uniqueness, apart from the positive definiteness. The latter is shown by studying $z^H Y z$ for all eigenvectors z of A.

A. M. Lyapunov. *The General Problem of the Stability of Motion* (in Russian). Doctoral dissertation, Univ. Kharkov 1892. English translation: Stability of Motion, Academic Press, New-York & London, 1966.

P. Lancaster, M. Tismenetsky. *The Theory of Matrices* (2nd edition). Academic Press, Orlando, FL, 1985. [Chapter 13]

Theorem (Lyapunov 1892)

Let $A \in \mathbb{R}^{n\times n}$ and consider the Lyapunov operator $\mathcal{L} : X \to AX + XA^T$. Then the following are equivalent:

(a) $\forall Y > 0\colon \exists X > 0\colon \mathcal{L}(X) = -Y$,

(b) $\exists Y > 0\colon \exists X > 0\colon \mathcal{L}(X) = -Y$,

(c) $\Lambda(A) \subset \mathbb{C}^- := \{z \in \mathbb{C} \,|\, \Re z < 0\}$, *i.e., A is (asymptotically) stable or Hurwitz.*

Important in applications: the nonnegative case:

$$\mathcal{L}(X) = AX + XA^T = -WW^T, \quad \text{where} \quad W \in \mathbb{R}^{n\times n_W},\ n_W \ll n.$$

A Hurwitz $\Rightarrow$ $\exists$ unique solution $X = ZZ^T$ for $Z \in \mathbb{R}^{n\times n_X}$ with $1 \le n_X \le n$.

A. M. Lyapunov. *The General Problem of the Stability of Motion* (in Russian). Doctoral dissertation, Univ. Kharkov 1892. English translation: Stability of Motion, Academic Press, New-York & London, 1966.

P. Lancaster, M. Tismenetsky. *The Theory of Matrices* (2nd edition). Academic Press, Orlando, FL, 1985. [Chapter 13]

From Lyapunov's theorem, one immediately obtains a characterization of asymptotic stability of linear dynamical systems

$$\dot{x}(t) = Ax(t). \qquad (2)$$

Theorem (Lyapunov)

The following are equivalent:

- *For (2), the zero state is asymptotically stable.*
- *The Lyapunov equation $AX + XA^T = Y$ has a unique solution $X = X^T > 0$ for all $Y = Y^T < 0$.*
- *A is Hurwitz.*

A. M. Lyapunov. The General Problem of the Stability of Motion (In Russian). Doctoral dissertation, Univ. Kharkov 1892. English translation: Stability of Motion, Academic Press, New-York & London, 1966.

Solving AREs by Newtons's Method

Feedback control design often involves solution of

$$A^T X + XA - XGX + H = 0, \quad G = G^T, H = H^T.$$

$\rightsquigarrow$ In each Newton step, solve Lyapunov equation

$$(A - GX_j)^T X_{j+1} + X_{j+1}(A - GX_j) = -X_j G X_j - H.$$

Solving AREs by Newtons's Method

Feedback control design often involves solution of

$$A^T X + XA - XGX + H = 0, \quad G = G^T, H = H^T.$$

$\rightsquigarrow$ In each Newton step, solve Lyapunov equation

$$(A - GX_j)^T X_{j+1} + X_{j+1}(A - GX_j) = -X_j G X_j - H.$$

Decoupling of dynamical systems, e.g., in slow/fast modes, requires solution of nonsymmetric ARE

$$AX + XF - XGX + H = 0.$$

$\rightsquigarrow$ In each Newton step, solve Sylvester equation

$$(A - X_j G)X_{j+1} + X_{j+1}(F - GX_j) = -X_j G X_j - H.$$

Also occurs in solving DSGE models, but how to compute desired solution?

Model Reduction via Balanced Truncation

For linear dynamical system

$$\dot{x}(t) = Ax(t) + Bu(t), \quad y(t) = Cx_r(t), \qquad x(t) \in \mathbb{R}^n$$

find reduced-order system

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t), \quad y_r(t) = C_r x_r(t), \qquad x(t) \in \mathbb{R}^r, \quad r \ll n$$

such that $\|y(t) - y_r(t)\| < \delta$.

The popular method balanced truncation requires the solution of the dual Lyapunov equations

$$AX + XA^T + BB^T = 0, \qquad A^T Y + YA + C^T C = 0.$$

Sylvester equation $AX - XB = C$ is equivalent to linear system of equations

$$\left(I_m \otimes A - B^T \otimes I_n\right) \operatorname{vec}(X) = \operatorname{vec}(C).$$

Sylvester equation $AX - XB = C$ is equivalent to linear system of equations

$$\left(I_m \otimes A - B^T \otimes I_n\right) \operatorname{vec}(X) = \operatorname{vec}(C).$$

This cannot be used for numerical solutions unless $nm \leq 100$ (or so), as

- direct solver requires $\mathcal{O}(n^2 m^2)$ of storage and $\mathcal{O}(n^3 m^3)$ flops;

Sylvester equation $AX - XB = C$ is equivalent to linear system of equations

$$\left(I_m \otimes A - B^T \otimes I_n\right) \mathrm{vec}(X) = \mathrm{vec}(C).$$

This cannot be used for numerical solutions unless $nm \leq 100$ (or so), as

- direct solver requires $\mathcal{O}(n^2 m^2)$ of storage and $\mathcal{O}(n^3 m^3)$ flops;
- (potential) low (tensor-)rank of right-hand side is ignored;

Sylvester equation $AX - XB = C$ is equivalent to linear system of equations

$$\left(I_m \otimes A - B^T \otimes I_n\right) \mathrm{vec}(X) = \mathrm{vec}(C).$$

This cannot be used for numerical solutions unless $nm \leq 100$ (or so), as

- direct solver requires $\mathcal{O}(n^2 m^2)$ of storage and $\mathcal{O}(n^3 m^3)$ flops;
- (potential) low (tensor-)rank of right-hand side is ignored;
- in Lyapunov case, symmetry and possible definiteness are not respected.

Sylvester equation $AX - XB = C$ is equivalent to linear system of equations

$$\left(I_m \otimes A - B^T \otimes I_n\right) \operatorname{vec}(X) = \operatorname{vec}(C).$$

This cannot be used for numerical solutions unless $nm \leq 100$ (or so), as

- direct solver requires $\mathcal{O}(n^2 m^2)$ of storage and $\mathcal{O}(n^3 m^3)$ flops;
- (potential) low (tensor-)rank of right-hand side is ignored;
- in Lyapunov case, symmetry and possible definiteness are not respected.

Possible solvers:

- Hessenberg-Schur or Bartels-Stewart method [Bartels/Stewart '72, Golub/Nash/Van Loan '79]
- Sign function method [Roberts '71, B '04]
- Krylov subspace solvers in operator from [Hochbruck, Starke, Reichel, ...]
- Block-Tensor-Krylov subspace methods with truncation [Kressner/Tobler, Bollhöfer/Eppler, B./Breiten, ...]
- Galerkin-type methods based on (extended, rational) Krylov subspace methods [Jaimoukha, Kasenally, Jbilou, Simoncini, Druskin, Knizhermann,...]
- ADI methods [Wachspress, Reichel, Li^2, Penzl, B, Saak, Kürschner, Truhar, Tomljanović...]

Sylvester Equations

Find $X \in \mathbb{R}^{n \times m}$ solving

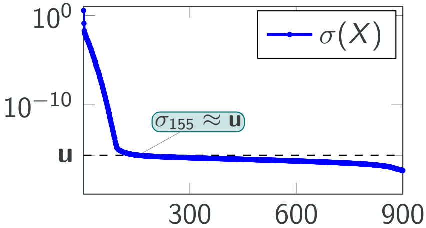
$$AX - XB = FG^T,$$

where $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times m}$, $F \in \mathbb{R}^{n \times r}$, $G \in \mathbb{R}^{m \times r}$.

If n, m large, but $r \ll n, m$
$\rightsquigarrow$ X has a small numerical rank.
[Penzl 1999, Grasedyck 2004, Antoulas/Sorensen/Zhou 2002]

$$\operatorname{rank}(X, \tau) = f \ll \min(n, m)$$

singular values of 1600 × 900 example



$\rightsquigarrow$ Compute low-rank solution factors $Z \in \mathbb{R}^{n \times f}$, $Y \in \mathbb{R}^{m \times f}$, $D \in \mathbb{R}^{f \times f}$, such that $X \approx ZDY^T$ with $f \ll \min(n, m)$.

Lyapunov Equations

Find $X \in \mathbb{R}^{n \times n}$ solving

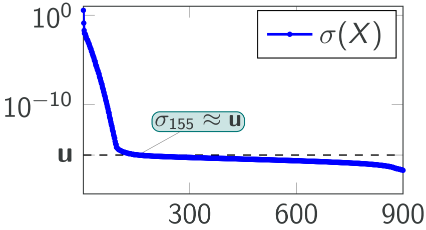
$$AX + XA^T = -FF^T,$$

where $A \in \mathbb{R}^{n \times n}$, $F \in \mathbb{R}^{n \times r}$.

If n large, but $r \ll n$
⇝ X has a small numerical rank.
[Penzl 1999, Grasedyck 2004, Antoulas/Sorensen/Zhou 2002]

$$\operatorname{rank}(X, \tau) = f \ll n$$

singular values of 1600×900 example



⇝ Compute low-rank solution factors $Z \in \mathbb{R}^{n \times f}$, $D \in \mathbb{R}^{f \times f}$, such that $X \approx ZDZ^T$ with $f \ll n$.

Consider $AX - XB + FG^T = 0$ with $\Lambda(A) \subset \mathbb{C}^-$ and $\Lambda(B) \subset \mathbb{C}^+$.

Definition

Recall: the matrix sign function of $M \in \mathbb{R}^{n \times n}$ with no purely imaginary eigenvalues is

$$\operatorname{sign}(M) = \operatorname{sign}\left(T \begin{bmatrix} J_- & 0 \\ 0 & J_+ \end{bmatrix} T^{-1} \right) = T \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} T^{-1}$$

with $J_\pm$ containing all Jordan blocks of M corresponding to eigenvalues with positive/negative real parts.

Consider $AX - XB + FG^T = 0$ with $\Lambda(A) \subset \mathbb{C}^-$ and $\Lambda(B) \subset \mathbb{C}^+$.

Definition

Recall: the matrix sign function of $M \in \mathbb{R}^{n\times n}$ with no purely imaginary eigenvalues is

$$\operatorname{sign}(M) = \operatorname{sign}\left(T \begin{bmatrix} J_- & 0 \\ 0 & J_+ \end{bmatrix} T^{-1}\right) = T \begin{bmatrix} -I & 0 \\ 0 & I \end{bmatrix} T^{-1}$$

with $J_\pm$ containing all Jordan blocks of M corresponding to eigenvalues with positive/negative real parts.

Observations

1. $\operatorname{sign}\left(\begin{bmatrix} A & FG^T \\ 0 & B \end{bmatrix}\right) = \begin{bmatrix} -I & 2X \\ 0 & I \end{bmatrix}$.
2. $\operatorname{sign}(M) = \lim_{k\to\infty} M_k$ with $M_{k+1} = \frac{1}{2}(M_k + M_k^{-1})$ if $M_0 = M$.

Consider $AX - XB + FG^T = 0$ with $\Lambda(A) \subset \mathbb{C}^-$ and $\Lambda(B) \subset \mathbb{C}^+$.

Observations

1. $\text{sign}\left(\begin{bmatrix} A & FG^T \\ 0 & B \end{bmatrix}\right) = \begin{bmatrix} -I & 2X \\ 0 & I \end{bmatrix}$.
2. $\text{sign}(M) = \lim_{k\to\infty} M_k$ with $M_{k+1} = \frac{1}{2}(M_k + M_k^{-1})$ if $M_0 = M$.

Sign function iteration for solving Sylvester equations

$M_0 = \begin{bmatrix} A_0 & F_0 G_0^T \\ 0 & B_0 \end{bmatrix} = \begin{bmatrix} A & FG^T \\ 0 & B \end{bmatrix}$, and inversion formula for block-triangular matrices:

$$A_{k+1} \leftarrow \frac{1}{2}(A_k + A_k^{-1}), \qquad B_{k+1} \leftarrow \frac{1}{2}(B_k + B_k^{-1}),$$

$$F_{k+1}G_{k+1}^T \leftarrow \frac{1}{2}(F_k G_k^T + A_k^{-1} F_k G_k^T B_k^{-1}) = \frac{1}{2}[\,F_k,\ A_k^{-1}F_k\,][\,G_k,\ B_k^{-T}G_k]^T,$$

so that $F_k G_k^T \to 2X$.

Consider $AX - XB + FG^T = 0$ with $\Lambda(A) \subset \mathbb{C}^-$ and $\Lambda(B) \subset \mathbb{C}^+$.

Factored sign function iteration for Sylvester equations [B. 2004]

$$\begin{aligned} A_{k+1} &\leftarrow \frac{1}{2}(A_k + A_k^{-1}), & B_{k+1} &\leftarrow \frac{1}{2}(B_k + B_k^{-1}), \\ F_{k+1} &\leftarrow \frac{1}{\sqrt{2}}[\, F_k,\ A_k^{-1} F_k \,], & G_{k+1} &\leftarrow \frac{1}{\sqrt{2}}[\, G_k,\ B_k^{-T} G_k \,] \end{aligned}$$

Problem: number of columns in F_k, G_k doubles each iteration!

Consider $AX - XB + FG^T = 0$ with $\Lambda(A) \subset \mathbb{C}^-$ and $\Lambda(B) \subset \mathbb{C}^+$.

Factored sign function iteration for Sylvester equations [B. 2004]

$$A_{k+1} \leftarrow \frac{1}{2}(A_k + A_k^{-1}), \qquad B_{k+1} \leftarrow \frac{1}{2}(B_k + B_k^{-1}),$$
$$F_{k+1} \leftarrow \frac{1}{\sqrt{2}}[\,F_k,\ A_k^{-1}F_k\,], \quad G_{k+1} \leftarrow \frac{1}{\sqrt{2}}[\,G_k,\ B_k^{-T}G_k\,]$$

Problem: number of columns in F_k, G_k doubles each iteration!

Cure: truncation operator

$$F_{k+1} \leftarrow \mathcal{T}_\varepsilon\left(\frac{1}{\sqrt{2}}[\,F_k,\ A_k^{-1}F_k\,]\right)$$

with, e.g., $\mathcal{T}_\varepsilon$ returning the scaled left singular vectors of the truncated SVD w.r.t. the numerical rank tolerance ε, similar for G_{k+1}.

Sylvester and Stein equations

Let $\alpha \neq \beta$ with $\alpha \notin \Lambda(B)$, $\beta \notin \Lambda(A)$, then

$$\underbrace{AX - XB = FG^T}_{\text{Sylvester equation}} \quad \Leftrightarrow \quad \underbrace{X = \mathcal{A} X \mathcal{B} + (\beta - \alpha)\mathcal{F}\mathcal{G}^H}_{\text{Stein equation}}$$

with the Cayley like transformations

$$\mathcal{A} := (A - \beta I_n)^{-1}(A - \alpha I_n), \qquad \mathcal{B} := (B - \alpha I_m)^{-1}(B - \beta I_m),$$
$$\mathcal{F} := (A - \beta I_n)^{-1} F, \qquad \mathcal{G} := (B - \alpha I_m)^{-H} G.$$

$\rightsquigarrow$ fix point iteration

$$X_k = \mathcal{A} X_{k-1} \mathcal{B} + (\beta - \alpha)\mathcal{F}\mathcal{G}^H$$

for $k \geq 1$, $X_0 \in \mathbb{R}^{n \times m}$.

Sylvester and Stein equations

Let $\alpha_\mathbf{k} \neq \beta_\mathbf{k}$ with $\alpha_\mathbf{k} \notin \Lambda(B)$, $\beta_\mathbf{k} \notin \Lambda(A)$, then

$$\underbrace{AX - XB = FG^T}_{\text{Sylvester equation}} \quad \Leftrightarrow \quad \underbrace{X = \mathcal{A}_\mathbf{k} X \mathcal{B}_\mathbf{k} + (\beta_\mathbf{k} - \alpha_\mathbf{k}) \mathcal{F}_\mathbf{k} \mathcal{G}_\mathbf{k}^H}_{\text{Stein equation}}$$

with the Cayley like transformations

$$\mathcal{A}_\mathbf{k} := (A - \beta_\mathbf{k} I_n)^{-1}(A - \alpha_\mathbf{k} I_n), \qquad \mathcal{B}_\mathbf{k} := (B - \alpha_\mathbf{k} I_m)^{-1}(B - \beta_\mathbf{k} I_m),$$
$$\mathcal{F}_\mathbf{k} := (A - \beta_\mathbf{k} I_n)^{-1} F, \qquad \mathcal{G}_\mathbf{k} := (B - \alpha_\mathbf{k} I_m)^{-H} G.$$

$\rightsquigarrow$ **alternating directions implicit (ADI)** iteration

$$X_k = \mathcal{A}_\mathbf{k} X_{k-1} \mathcal{B}_\mathbf{k} + (\beta_\mathbf{k} - \alpha_\mathbf{k}) \mathcal{F}_\mathbf{k} \mathcal{G}_\mathbf{k}^H$$

for $k \geq 1$, $X_0 \in \mathbb{R}^{n \times m}$. [Wachspress 1988]

Sylvester ADI iteration [Wachspress 1988]

$$X_k = \mathcal{A}_k X_{k-1} \mathcal{B}_k + (\beta_k - \alpha_k)\mathcal{F}_k \mathcal{G}_k^H,$$

$$\mathcal{A}_k := (A - \beta_k I_n)^{-1}(A - \alpha_k I_n), \quad \mathcal{B}_k := (B - \alpha_k I_m)^{-1}(B - \beta_k I_m),$$

$$\mathcal{F}_k := (A - \beta_k I_n)^{-1} F \in \mathbb{R}^{n\times r}, \quad \mathcal{G}_k := (B - \alpha_k I_m)^{-H} G \in \mathbb{C}^{m\times r}.$$

Now set $X_0 = 0$ and find factorization $X_k = Z_k D_k Y_k^H$

$$X_1 = \mathcal{A}_1 X_0 \mathcal{B}_1 + (\beta_1 - \alpha_1)\mathcal{F}_1 \mathcal{G}_1^H$$

Sylvester ADI iteration [Wachspress 1988]

$$X_k = \mathcal{A}_k X_{k-1} \mathcal{B}_k + (\beta_k - \alpha_k) \mathcal{F}_k \mathcal{G}_k^H,$$
$$\mathcal{A}_k := (A - \beta_k I_n)^{-1}(A - \alpha_k I_n), \quad \mathcal{B}_k := (B - \alpha_k I_m)^{-1}(B - \beta_k I_m),$$
$$\mathcal{F}_k := (A - \beta_k I_n)^{-1} F \in \mathbb{R}^{n \times r}, \quad \mathcal{G}_k := (B - \alpha_k I_m)^{-H} G \in \mathbb{C}^{m \times r}.$$

Now set $X_0 = 0$ and find factorization $X_k = Z_k D_k Y_k^H$

$$X_1 = (\beta_1 - \alpha_1)(A - \beta_1 I_n)^{-1} F G^T (B - \alpha_1 I_m)^{-1}$$
$$\Rightarrow V_1 := Z_1 = (A - \beta_1 I_n)^{-1} F \in \mathbb{R}^{n \times r}, \quad D_1 = (\beta_1 - \alpha_1) I_r \in \mathbb{R}^{r \times r},$$
$$W_1 := Y_1 = (B - \alpha_1 I_m)^{-H} G \in \mathbb{C}^{m \times r}.$$

Sylvester ADI iteration [Wachspress 1988]

$$X_k = \mathcal{A}_k X_{k-1} \mathcal{B}_k + (\beta_k - \alpha_k)\mathcal{F}_k \mathcal{G}_k^H,$$

$$\mathcal{A}_k := (A - \beta_k I_n)^{-1}(A - \alpha_k I_n), \quad \mathcal{B}_k := (B - \alpha_k I_m)^{-1}(B - \beta_k I_m),$$

$$\mathcal{F}_k := (A - \beta_k I_n)^{-1} F \in \mathbb{R}^{n \times r}, \quad \mathcal{G}_k := (B - \alpha_k I_m)^{-H} G \in \mathbb{C}^{m \times r}.$$

Now set $X_0 = 0$ and find factorization $X_k = Z_k D_k Y_k^H$

$$X_2 = \mathcal{A}_2 X_1 \mathcal{B}_2 + (\beta_2 - \alpha_2)\mathcal{F}_2 \mathcal{G}_2^H = \ldots =$$

$$V_2 = V_1 + (\beta_2 - \alpha_1)(A + \beta_2 I)^{-1} V_1 \in \mathbb{R}^{n \times r},$$

$$W_2 = W_1 + \overline{(\alpha_2 - \beta_1)}(B + \alpha_2 I)^{-H} W_1 \in \mathbb{R}^{m \times r},$$

$$Z_2 = [Z_1,\ V_2], \qquad D_2 = \operatorname{diag}(D_1, (\beta_2 - \alpha_2) I_r),$$

$$Y_2 = [Y_1,\ W_2].$$

Solving Large-Scale Sylvester and Lyapunov Equations

LR-ADI Algorithm [B. 2005, Li/Truhar 2008, B./Li/Truhar 2009]

Algorithm 2 Low-rank Sylvester ADI / factored ADI (fADI)

Input: Matrices defining $AX - XB = FG^T$ and shift parameters $\{\alpha_1, \ldots, \alpha_{k_{\max}}\}$, $\{\beta_1, \ldots, \beta_{k_{\max}}\}$.

Output: Z, D, Y such that $ZDY^H \approx X$.

$Z_1 = V_1 = (A - \beta_1 I_n)^{-1} F$.

$Y_1 = W_1 = (B - \alpha_1 I_m)^{-H} G$.

$D_1 = (\beta_1 - \alpha_1) I_r$

for $k = 2, \ldots, k_{\max}$ **do**

  $V_k = V_{k-1} + (\beta_k - \alpha_{k-1})(A - \beta_k I_n)^{-1} V_{k-1}$.

  $W_k = W_{k-1} + (\alpha_k - \beta_{k-1})(B - \alpha_k I_n)^{-H} W_{k-1}$.

  Update solution factors

  $Z_k = [Z_{k-1}, V_k]$, $Y_k = [Y_{k-1}, W_k]$, $D_k = \operatorname{diag}(D_{k-1}, (\beta_k - \alpha_k) I_r)$.

end for

Optimal Shifts

Solution of rational optimization problem

$$\min_{\substack{\alpha_j \in \mathbb{C} \\ \beta_j \in \mathbb{C}}} \max_{\substack{\lambda \in \Lambda(A) \\ \mu \in \Lambda(B)}} \prod_{j=1}^{k} \left| \frac{(\lambda - \alpha_j)(\mu - \beta_j)}{(\lambda - \beta_j)(\mu - \alpha_j)} \right|,$$

for which no analytic solution is known in general.

Optimal Shifts

Solution of rational optimization problem

$$\min_{\substack{\alpha_j \in \mathbb{C} \\ \beta_j \in \mathbb{C}}} \max_{\substack{\lambda \in \Lambda(A) \\ \mu \in \Lambda(B)}} \prod_{j=1}^{k} \left| \frac{(\lambda - \alpha_j)(\mu - \beta_j)}{(\lambda - \beta_j)(\mu - \alpha_j)} \right|,$$

for which no analytic solution is known in general.

Some shift generation approaches:

- generalized Bagby points, [LEVENBERG/REICHEL 1993]
- adaption of Penzl's cheap heuristic approach available [PENZL 1999, LI/TRUHAR 2008]
  $\rightsquigarrow$ approximate $\Lambda(A)$, $\Lambda(B)$ by small number of Ritz values w.r.t. A, A^{-1}, B, B^{-1} via Arnoldi,
- just taking these Ritz values alone also works well quite often.

Disadvantages of Low-Rank ADI as of 2012:

1. No efficient stopping criteria:
   - Difference in iterates $\rightsquigarrow$ norm of added columns/step: not reliable, stops often too late.
   - Residual is a full dense matrix, can not be calculated as such.
2. Requires complex arithmetic for real coefficients when complex shifts are used.
3. Expensive (only semi-automatic) set-up phase to precompute ADI shifts.

Disadvantages of Low-Rank ADI as of 2012:

1. No efficient stopping criteria:
   - Difference in iterates $\rightsquigarrow$ norm of added columns/step: not reliable, stops often too late.
   - Residual is a full dense matrix, can not be calculated as such.
2. Requires complex arithmetic for real coefficients when complex shifts are used.
3. Expensive (only semi-automatic) set-up phase to precompute ADI shifts.

None of these disadvantages exists as of today
$\Longrightarrow$ speed-ups old vs. new LR-ADI can be up to 20!

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Projection-based methods for Lyapunov equations with $A + A^T < 0$:

1. Compute orthonormal basis $\mathrm{range}(Z)$, $Z \in \mathbb{R}^{n \times r}$, for subspace $\mathcal{Z} \subset \mathbb{R}^n$, $\dim \mathcal{Z} = r$.
2. Set $\hat{A} := Z^T A Z$, $\hat{B} := Z^T B$.
3. Solve small-size Lyapunov equation $\hat{A}\hat{X} + \hat{X}\hat{A}^T + \hat{B}\hat{B}^T = 0$.
4. Use $X \approx Z\hat{X}Z^T$.

Examples:

- Krylov subspace methods, i.e., for $m = 1$:

$$\mathcal{Z} = \mathcal{K}(A, B, r) = \mathrm{span}\{B, AB, A^2B, \ldots, A^{r-1}B\}$$

[Saad 1990, Jaimoukha/Kasenally 1994, Jbilou 2002–2008].

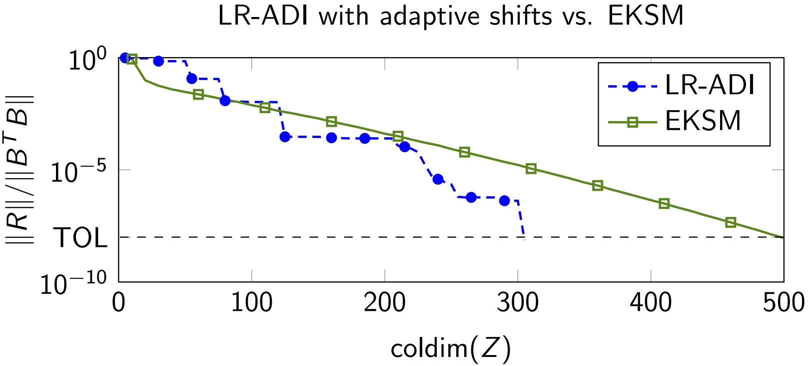
- Extended Krylov subspace method (EKSM) [Simoncini 2007],

$$\mathcal{Z} = \mathcal{K}(A, B, r) \cup \mathcal{K}(A^{-1}, B, r).$$

- Rational Krylov subspace methods (RKSM) [Druskin/Simoncini 2011].

- FEM discretization of a simple 3D ocean circulation model (barotropic, constant depth) $\rightsquigarrow$ stiffness matrix $-A$ with $n = 42,249$, choose artificial constant term $B = \texttt{rand(n,5)}$.

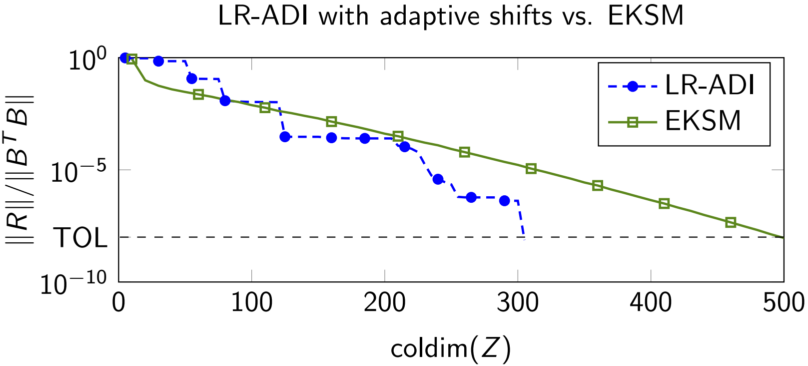
- FEM discretization of a simple 3D ocean circulation model (barotropic, constant depth) $\rightsquigarrow$ stiffness matrix $-A$ with $n = 42,249$, choose artificial constant term $B =$ `rand(n,5)`.
- **Convergence history:**



Numerical Performance

Example: an ocean circulation problem[Van Gijzen et al. 1998]

- FEM discretization of a simple 3D ocean circulation model (barotropic, constant depth) $\rightsquigarrow$ stiffness matrix $-A$ with $n = 42,249$, choose artificial constant term $B = \texttt{rand(n,5)}$.
- **Convergence history:**



- CPU times: LR-ADI $\approx$ 110 sec, EKSM $\approx$ 135 sec.

- Numerical enhancements of low-rank ADI for large Sylvester/Lyapunov equations:
  1. low-rank residuals, reformulated implementation,
  2. compute real low-rank factors in the presence of complex shifts,
  3. self-generating shift strategies (quantification in progress).

For diffusion-convection-reaction example:
332.02 sec. down to **17.24 sec.** $\rightsquigarrow$ acceleration by factor almost **20**.

- Generalized version enables derivation of low-rank solvers for various generalized Sylvester equations.
- Ongoing work:
  - Apply LR-ADI in Newton methods for algebraic Riccati equations

$$\mathcal{D}(X) = AXA^T - EXE^T + GG^T + A^TXF(I_r + F^TXF)^{-1}F^TXA = 0.$$

- For nonlinear AREs see

P. Benner, P. Kürschner, J. Saak. *Low-rank Newton-ADI methods for large nonsymmetric algebraic Riccati equations.* J. Franklin Inst., 353(5):1147–1167, 2016.

Further Reading

P. Benner and T. Breiten.
Low rank methods for a class of generalized Lyapunov equations and related issues.
Numerische Mathematik 124(3):441–470, 2013.

P. Benner and T. Damm.
Lyapunov equations, energy functionals, and model order reduction of bilinear and stochastic systems.
SIAM Journal on Control and Optimization 49(2):686–711, 2011.

P. Benner and P. Kürschner.
Computing real low-rank solutions of Sylvester equations by the factored ADI method.
Computers and Mathematics with Applications 67:1656–1672, 2014.

P. Benner, P. Kürschner, and J. Saak.
Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method.
Numerical Algorithms 62(2):225–251, 2013.

P. Benner, P. Kürschner, and J. Saak.
Self-generating and efficient shift parameters in ADI methods for large Lyapunov and Sylvester equations.
Electronic Transactions on Numerical Analysis, 43:142–162, 2014.

P. Benner and J. Saak.
Numerical solution of large and sparse continuous time algebraic matrix Riccati and Lyapunov equations: a state of the art survey.
GAMM Mitteilungen 36(1):32–52, 2013.