PMAA 14
Lugano July 4, 2014

BLAS Level-3 Implementation of Common Solvers for Generalized Quasi-Triangular Lyapunov Equations

Peter Benner Martin Köhler Jens Saak

Computational Methods in Systems and Control Theory
Max Planck Institute for Dynamics of Complex Technical Systems

Introduction

We consider the Generalized Lyapunov Equation

$$A^T X E + E^T X A = Y, \tag{1}$$

where A, E, X and Y are real $n \times n$ matrices. Furthermore the right hand side Y and the solution X are *symmetric*.

Introduction

We consider the Generalized Lyapunov Equation

$$A^T X E + E^T X A = Y, \tag{1}$$

where A, E, X and Y are real $n \times n$ matrices. Furthermore the right hand side Y and the solution X are *symmetric*.

Solvability Condition

Equation (1) is uniquely solvable if and only if

$$\lambda_i + \lambda_j \neq 0$$

holds for any two eigenvalues λ_i, λ_j of (A, E).

Introduction

We consider the Generalized Lyapunov Equation

$$A^T X E + E^T X A = Y, \quad (1)$$

where ... Y are real $n \times n$ matrices. Furthermore the right hand side Y and the solution X are *symmetric*.

Applications:

- Computing stabilizing feedbacks of LTI systems,
- Model Order Reduction,
- Newton's Method to solve Algebraic Riccati Equations,
- Time - integration of Differential Riccati Equations,
- ...

Solvability Condition

Equation (1) is uniquely solvable if and only if

$$\lambda_i + \lambda_j \neq 0$$

holds for any two eigenvalues λ_i, λ_j of (A, E).

Introduction

Basic Solution Techniques

1. Rewrite the Lyapunov Equation into a linear system

$$\left(E^T \otimes A^T + A^T \otimes E^T\right) \operatorname{vec}(X) = \operatorname{vec}(Y),$$

of squared dimension. Here $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ describes the column-wise concatenation of a matrix into a vector.

Introduction

Basic Solution Techniques

1. Rewrite the Lyapunov Equation into a linear system

$$\left(E^T \otimes A^T + A^T \otimes E^T\right) \operatorname{vec}(X) = \operatorname{vec}(Y),$$

of squared dimension. Here $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ describes the column-wise concatenation of a matrix into a vector.

$\rightarrow$ Complexity using LU-Decomposition: $\frac{2}{3}n^6 + 2n^4$ Flops. ↯

Introduction

Basic Solution Techniques

1. Rewrite the Lyapunov Equation into a linear system

$$\left(E^T \otimes A^T + A^T \otimes E^T\right) \operatorname{vec}(X) = \operatorname{vec}(Y),$$

of squared dimension. Here $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ describes the column-wise concatenation of a matrix into a vector.
$\rightarrow$ Complexity using LU-Decomposition: $\frac{2}{3}n^6 + 2n^4$ Flops. ↯

2. Use a Generalized Sylvester Equation solver.
[Gardiner, Laub, Moler '92, Kågström, Westin '89]

Introduction

Basic Solution Techniques

1. Rewrite the Lyapunov Equation into a linear system

$$\left(E^T \otimes A^T + A^T \otimes E^T\right) \operatorname{vec}(X) = \operatorname{vec}(Y),$$

of squared dimension. Here $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ describes the column-wise concatenation of a matrix into a vector.

$\rightarrow$ Complexity using LU-Decomposition: $\frac{2}{3}n^6 + 2n^4$ Flops. ↯

2. Use a Generalized Sylvester Equation solver.

[Gardiner, Laub, Moler '92, Kågström, Westin '89]

Does not guarantee the symmetry of X from a numerical point of view. ↯

Introduction

Basic Solution Techniques

1. Rewrite the Lyapunov Equation into a linear system

$$\left(E^T \otimes A^T + A^T \otimes E^T\right) \operatorname{vec}(X) = \operatorname{vec}(Y),$$

   of squared dimension. Here $\otimes$ denotes the Kronecker product and $\operatorname{vec}(\cdot)$ describes the column-wise concatenation of a matrix into a vector.

   $\rightarrow$ Complexity using LU-Decomposition: $\frac{2}{3}n^6 + 2n^4$ Flops. ↯

2. Use a Generalized Sylvester Equation solver.

   [Gardiner, Laub, Moler '92, Kågström, Westin '89]

   Does not guarantee the symmetry of X from a numerical point of view. ↯

3. Use the Matrix Sign Function: [Quintana-Ortí, Benner '99]
   - Iterative solver $\rightarrow$ only approximate solution, ↯
   - Only feasible for $\Lambda(A, E) \subset \mathbb{C}_-$. ↯

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation

[Penzl '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [Penzl '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

2. Transform Equation (1) using Q and Z:

$$Z^T A_s^T Q X Q^T E_s Z + Z^T E_s^T Q X Q^T A_s Z = Y.$$

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [Penzl '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

2. Transform Equation (1) using Q and Z:

$$A_s^T Q X Q^T E_s + E_s^T Q X Q^T A_s = Z Y Z^T.$$

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [Penzl '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

2. Transform Equation (1) using Q and Z:

$$A_s^T \underbrace{QXQ^T}_{X_s} E_s + E_s^T \underbrace{QXQ^T}_{X_s} A_s = \underbrace{ZYZ^T}_{Y_s}.$$

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [Penzl '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

2. Transform Equation (1) using Q and Z:

$$A_s^T \underbrace{QXQ^T}_{X_s} E_s + E_s^T \underbrace{QXQ^T}_{X_s} A_s = \underbrace{ZYZ^T}_{Y_s}.$$

3. Restore the solution X by:

$$X = Q^T X_s Q.$$

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [PENZL '97]

The overall procedure to solve a Generalized Lyapunov Equation:

1. Compute the real Generalized Schur Decomposition of (A, E):

$$A = Q^T A_s Z \quad \text{and} \quad E = Q^T E_s Z.$$

2. Transform Equation (1) using Q and Z:

$$A_s^T \underbrace{QXQ^T}_{X_s} E_s + E_s^T \underbrace{QXQ^T}_{X_s} A_s = \underbrace{ZYZ^T}_{Y_s}.$$

3. Restore the solution X by:

$$X = Q^T X_s Q.$$

We focus only on solving the second step.

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation [Penzl '97]

The real Generalized Schur Decomposition of (A, E) yields:

$$A_s = \begin{pmatrix} A_{11} & \cdots & A_{1p} \\ & \ddots & \vdots \\ 0 & & A_{pp} \end{pmatrix}, \quad E_s = \begin{pmatrix} E_{11} & \cdots & E_{1p} \\ & \ddots & \vdots \\ 0 & & E_{pp} \end{pmatrix},$$

$$X_s = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{p1} & \cdots & X_{pp} \end{pmatrix}, \quad Y_s = \begin{pmatrix} Y_{11} & \cdots & Y_{1p} \\ \vdots & \ddots & \vdots \\ Y_{p1} & \cdots & Y_{pp} \end{pmatrix},$$

where A_{ij}, E_{ij}, X_{ij} and Y_{ij} are $p \times p$ blocks of size 1×1 or 2×2 according to the eigenvalues of (A, E).

Block Algorithm

Bartels-Stewart Method for the Generalized Lyapunov Equation

[PENZL '97]

The real Generalized Schur Decomposition of (A, E) yields:

$$A_s = \begin{pmatrix} A_{11} & \cdots & A_{1p} \\ & \ddots & \vdots \\ 0 & & A_{pp} \end{pmatrix}, \quad E_s = \begin{pmatrix} E_{11} & \cdots & E_{1p} \\ & \ddots & \vdots \\ 0 & & E_{pp} \end{pmatrix},$$

$$X_s = \begin{pmatrix} X_{11} & \cdots & X_{1p} \\ \vdots & \ddots & \vdots \\ X_{p1} & \cdots & X_{pp} \end{pmatrix}, \quad Y_s = \begin{pmatrix} Y_{11} & \cdots & Y_{1p} \\ \vdots & \ddots & \vdots \\ Y_{p1} & \cdots & Y_{pp} \end{pmatrix},$$

where A_{ij}, E_{ij}, X_{ij} and Y_{ij} are $p \times p$ blocks of size 1×1 or 2×2 according to the eigenvalues of (A, E).

We have to solve Sylvester Equations:

$$A_{kk}^T X_{kl} E_{ll} + E_{kk}^T X_{kl} A_{ll} = \hat{Y}_{kl}$$

with updated right hand sides:

$$\hat{Y}_{kl} = Y_{kl} - \sum_{\substack{i=1, j=1 \\ (i,j) \neq (k,l)}}^{k,l} \left(A_{ik}^T X_{ij} E_{jl} + E_{ik}^T X_{ij} A_{jl} \right).$$

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y_{kl}}$ is performed using a more efficient scheme:

$$
\begin{aligned}
Y_{kl}^{(0)} &= Y_{kl} \\
Y_{kl}^{(2i-1)} &= Y_{kl}^{(2i-2)} - A_{ik}^T X_{i,1:l-1} E_{1:l-1,l} - E_{ik}^T X_{i,1:l-1} A_{1:l-1,l}, && i = 1, \ldots, k \\
Y_{kl}^{(2i)} &= Y_{kl}^{(2i-1)} - A_{ik}^T X_{il} E_{ll} - E_{ik}^T X_{il} A_{ll}, && i = 1, \ldots, k-1 \\
\hat{Y}_{kl} &= Y_{kl}^{(2k-1)}.
\end{aligned}
$$

Block Algorithm

Current Implementation

[PENZL '97, SLICOT]

- The update of $\hat{Y_{kl}}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation:

$$\left(E_{ll}^T \otimes A_{kk}^T + A_{ll}^T \otimes E_{kk}^T\right) \mathrm{vec}\left(X_{kl}\right) = \mathrm{vec}\left(\hat{Y}_{kl}\right),$$

which is at most an 8×8 linear system if A_{kk} and A_{ll} are 2×2 blocks.

Block Algorithm

Current Implementation [PENZL '97, SLICOT]

- The update of $\hat{Y_{kl}}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
  $\rightarrow$ Only solve $\frac{1}{2}p(p+1)$ Sylvester Equations.

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y_{kl}}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
- **But:** A block size of 1 or 2 results in level-2 BLAS operations. Utilization of modern computer architectures and accelerator devices is not optimal. ☹

Block Algorithm

Current Implementation [Penzl '97, SLICOT]

- The update of $\hat{Y}_{kl}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
- **But:** A block size of 1 or 2 results in level-2 BLAS operations.

Our Question:

Are the matrices $A_{kk},\ A_{ll},\ \ldots$ restricted to be 1×1 or 2×2 matrices?

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y}_{kl}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
- **But:** A block size of 1 or 2 results in level-2 BLAS operations.

Our Question:

Are the matrices A_{kk}, A_{ll}, $\ldots$ restricted to be 1×1 or 2×2 matrices?

Answer:

- **No:** The update of $\hat{Y}_{kl}$ is not restricted by block size.

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y}_{kl}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
- **But:** A block size of 1 or 2 results in level-2 BLAS operations.

Our Question:

Are the matrices $A_{kk}, A_{ll}, \ldots$ restricted to be 1×1 or 2×2 matrices?

Answer:

- **No:** The update of $\hat{Y}_{kl}$ is not restricted by block size.
- **Maybe:** The inner Sylvester Equation is solved via its Kronecker representation.

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y}_{kl}$ is performed using a more efficient scheme.

Example:

We consider a Generalized Lyapunov Equation of dimension $n = 960$ and a block size of 64×64.

- $\rightarrow$ 15×15 blocks, 120 inner Sylvester Equations.
- $\rightarrow$ Solution of one of them costs **45 GFlops** $\rightarrow$ **5.4 TFlops** overall.
- $\rightarrow$ Using a block size 1×1 results in **7 GFlops** including right hand side updates.
- $\rightarrow$ **Ratio of 770 between different block sizes.** $\lightning$

Our Question:

Are the matrices A_{kk}, A_{ll}, ... restricted to be 1×1 or 2×2 matrices?

Answer:

- **No:** The update of $\hat{Y}_{kl}$ is not restricted by block size.
- **Maybe:** The inner Sylvester Equation is solved via its Kronecker representation.

Block Algorithm

Current Implementation

[Penzl '97, SLICOT]

- The update of $\hat{Y}_{kl}$ is performed using a more efficient scheme.
- The remaining inner Sylvester Equations are solved using their Kronecker representation.
- The lower half of X is known by symmetry and not computed.
- **But:** A block size of 1 or 2 results in level-2 BLAS operations.

Example:

We consider a Generalized Lyapunov Equation of dimension $n = 960$ and a block size of 64×64.

→ 15×15 blocks, 120 inner Sylvester Equations.

→ Solution of one of them costs **45 GFlops** → **5.4 TFlops** overall.

→ Using a block size 1×1 results in **7 GFlops** including right hand side updates.

→ **Ratio of 770 between different block sizes.** ↯

We have to handle a larger inner Sylvester Equation without increasing the runtime complexity significantly.

Our Question:

Answer:

- **No:** The update of $\hat{Y}_{kl}$ is not restricted by block size.
- **Maybe:** The inner Sylvester Equation is solved via its Kronecker representation.

Block Algorithm

Algorithm and Flop Count

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B
Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

$X_s := Y_s$
for $k = 1, \ldots, P_B$ **do**
  if $k > 1$ **then**
    $X_{k,1:k-1} := X_{1:k-1,k}^T$ {Copy the symmetric part.}
  end if
  for $l = k, \ldots, P_B$ **do**
    if $l > 1$ **then**
      $X_{k:l,l} := X_{k:l,l} - A_{k,k:l}^T X_{k,1:l-1} E_{1:l-1,l}$
      $X_{k:l,l} := X_{k:l,l} - E_{k,k:l}^T X_{k,1:l-1} A_{1:l-1,l}$
    end if
    Solve $A_{k,k}^T X_* E_{l,l} + E_{k,k}^T X_* A_{l,l} = X_{k,l}$
    $X_{k,l} := X_*$
    if $k < l$ **then**
      $X_{k+1:l,l} := X_{k+1:l,l} - A_{k,k+1:l}^T X_{k,l} E_{l,l}$
      $X_{k+1:l,l} := X_{k+1:l,l} - E_{k,k+1:l}^T X_{k,l} A_{l,l}$
    end if
  end for
end for

Block Algorithm

Algorithm and Flop Count

Outer Algorithm

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B
Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

```
X_s := Y_s
for k = 1, ..., P_B do
    if k > 1 then
        X_{k,1:k-1} := X^T_{1:k-1,k}   {Copy the symmetric part.}
    end if
    for l = k, ..., P_B do
        if l > 1 then
            X_{k:l,l} := X_{k:l,l} - A^T_{k,k:l} X_{k,1:l-1} E_{1:l-1,l}
            X_{k:l,l} := X_{k:l,l} - E^T_{k,k:l} X_{k,1:l-1} A_{1:l-1,l}
        end if
        Solve A^T_{k,k} X_* E_{l,l} + E^T_{k,k} X_* A_{l,l} = X_{k,l}
        X_{k,l} := X_*
        if k < l then
            X_{k+1:l,l} := X_{k+1:l,l} - A^T_{k,k+1:l} X_{k,l} E_{l,l}
            X_{k+1:l,l} := X_{k+1:l,l} - E^T_{k,k+1:l} X_{k,l} A_{l,l}
        end if
    end for
end for
```

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(8lN_B^3 - 4lN_B^3\right) - 4N_B^3$$

Block Algorithm

Algorithm and Flop Count

Outer Algorithm | Inner Algorithm

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B
Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

```
X_s := Y_s
for k = 1, ..., P_B do
    if k > 1 then
        X_{k,1:k-1} := X^T_{1:k-1,k}   {Copy the symmetric part.}
    end if
    for l = k, ..., P_B do
        if l > 1 then
            X_{k:l,l} := X_{k:l,l} - A^T_{k,k:l} X_{k,1:l-1} E_{1:l-1,l}
            X_{k:l,l} := X_{k:l,l} - E^T_{k,k:l} X_{k,1:l-1} A_{1:l-1,l}
        end if
        Solve A^T_{k,k} X_* E_{l,l} + E^T_{k,k} X_* A_{l,l} = X_{k,l}
        X_{k,l} := X_*
        if k < l then
            X_{k+1:l,l} := X_{k+1:l,l} - A^T_{k,k+1:l} X_{k,l} E_{l,l}
            X_{k+1:l,l} := X_{k+1:l,l} - E^T_{k,k+1:l} X_{k,l} A_{l,l}
        end if
    end for
end for
```

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(8lN_B^3 - 4lN_B^3\right) - 4N_B^3$$

$$\frac{1}{2}\left({P_B}^2 + P_B\right)F(N_B)$$

Block Algorithm

Algorithm and Flop Count

Outer Algorithm | **Inner Algorithm**

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B

Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

$X_s := Y_s$
for $k = 1, \ldots, P_B$ **do**
  if $k > 1$ **then**
    $X_{k,1:k-1} := X_{1:k-1,k}^T$ {Copy the symmetric part.}
  end if
  for $l = k, \ldots, P_B$ **do**
    if $l > 1$ **then**
      $X_{k:l,l} := X_{k:l,l} - A_{k,k:l}^T X_{k,1:l-1} E_{1:l-1,l}$
      $X_{k:l,l} := X_{k:l,l} - E_{k,k:l}^T X_{k,1:l-1} A_{1:l-1,l}$
    end if
    Solve $A_{k,k}^T X_* E_{l,l} + E_{k,k}^T X_* A_{l,l} = X_{k,l}$
    $X_{k,l} := X_*$
    if $k < l$ **then**
      $X_{k+1:l,l} := X_{k+1:l,l} - A_{k,k+1:l}^T X_{k,l} E_{l,l}$
      $X_{k+1:l,l} := X_{k+1:l,l} - E_{k,k+1:l}^T X_{k,l} A_{l,l}$
    end if
  end for
end for

Flop count of lines 8–9:

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} (8lN_B^3 - 4lN_B^3) - 4N_B^3$$

Flop count of line 11:

$$\frac{1}{2}({P_B}^2 + P_B)F(N_B)$$

Flop count of lines 14–15:

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} (4lN_B^3 - 4kN_B^3 + 4N_B^3) - 4N_B^3 P_B$$

Block Algorithm

Algorithm and Flop Count

Outer Algorithm | Inner Algorithm

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B
Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

$X_s := Y_s$
for $k = 1, \ldots, P_B$ **do**
  if $k > 1$ **then**
    $X_{k,1:k-1} := X^T_{1:k-1,k}$ {Copy the symmetric part.}
  end if
  for $l = k, \ldots, P_B$ **do**
    if $l > 1$ **then**
      $X_{k:l,l} := X_{k:l,l} - A^T_{k,k:l} X_{k,1:l-1} E_{1:l-1,l}$
      $X_{k:l,l} := X_{k:l,l} - E^T_{k,k:l} X_{k,1:l-1} A_{1:l-1,l}$
    end if
    Solve $A^T_{k,k} X_* E_{l,l} + E^T_{k,k} X_* A_{l,l} = X_{k,l}$
    $X_{k,l} := X_*$
    if $k < l$ **then**
      $X_{k+1:l,l} := X_{k+1:l,l} - A^T_{k,k+1:l} X_{k,l} E_{l,l}$
      $X_{k+1:l,l} := X_{k+1:l,l} - E^T_{k,k+1:l} X_{k,l} A_{l,l}$
    end if
  end for
end for

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(8lN_B^3 - 4lN_B^3\right) - 4N_B^3$$

$$\frac{1}{2}\left({P_B}^2 + P_B\right) F(N_B)$$

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(4lN_B^3 - 4kN_B^3 + 4N_B^3\right) - 4N_B^3 P_B$$

Overall Flop Count

$$F_{\text{overall}}(n, N_B) := \frac{1}{2}\left(P_B^2 + P_B\right) F(N_B) + N_B^3 \left(\frac{8}{3} P_B^3 + 4P_B^2 - \frac{8}{3} P_B - 4\right)$$

Block Algorithm

Algorithm and Flop Count

Outer Algorithm | Inner Algorithm

Algorithm 1: Solution of the Generalized (Quasi-)Triangular Lyapunov Equation

Input: (A_s, E_s) and Y_s partitioned in P_B blocks of size N_B
Output: X_s solving the Generalized (Quasi-)Triangular Lyapunov Equation

$X_s := Y_s$
for $k = 1, \ldots, P_B$ **do**
  if $k > 1$ **then**
    $X_{k,1:k-1} := X^T_{1:k-1,k}$ {Copy the symmetric part.}
  end if
  for $l = k, \ldots, P_B$ **do**
    if $l > 1$ **then**
      $X_{k:l,l} := X_{k:l,l} - A^T_{k,k:l} X_{k,1:l-1} E_{1:l-1,l}$
      $X_{k:l,l} := X_{k:l,l} - E^T_{k,k:l} X_{k,1:l-1} A_{1:l-1,l}$
    end if
    Solve $A^T_{k,k} X_* E_{l,l} + E^T_{k,k} X_* A_{l,l} = X_{k,l}$
    $X_{k,l} := X_*$
    if $k < l$ **then**
      $X_{k+1:l,l} := X_{k+1:l,l} - A^T_{k,k+1:l} X_{k,l} E_{l,l}$
      $X_{k+1:l,l} := X_{k+1:l,l} - E^T_{k,k+1:l} X_{k,l} A_{l,l}$
    end if
  end for
end for

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(8lN_B^3 - 4lN_B^3\right) - 4N_B^3$$

$$\frac{1}{2}({P_B}^2 + P_B)F(N_B)$$

$$\sum_{k=1}^{P_B} \sum_{l=k}^{P_B} \left(4lN_B^3 - 4kN_B^3 + 4N_B^3\right) - 4N_B^3 P_B$$

Overall Flop Count

$$F_{\text{overall}}(n, N_B) := \frac{1}{2}\left(\frac{n^2}{N_B^2} + \frac{n}{N_B}\right) F(N_B) + \left(\frac{8}{3}n^3 + 4N_B n^2 - \frac{8}{3}N_B^2 n - 4N_B^3\right)$$

Inner Sylvester Equations

We have to solve Generalized Sylvester Equations

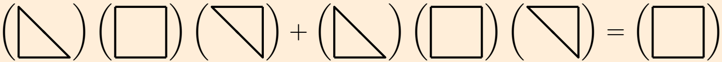
$$A_{kk}^T X_{kl} E_{ll} + E_{kk}^T X_{kl} A_{ll} = \hat{Y}_{kl}$$

Inner Sylvester Equations

We have to solve Generalized Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y},$$

where $\hat{A}$, $\hat{C} \in \mathbb{R}^{\hat{n} \times \hat{n}}$, $\hat{B}$, $\hat{D} \in \mathbb{R}^{\hat{m} \times \hat{m}}$ and $\hat{X}$, $\hat{Y} \in \mathbb{R}^{\hat{n} \times \hat{m}}$ with the structure



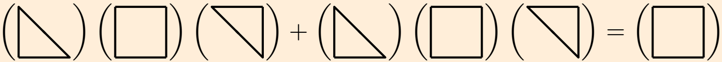
efficiently.

Inner Sylvester Equations

We have to solve Generalized Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y},$$

where $\hat{A}$, $\hat{C} \in \mathbb{R}^{\hat{n} \times \hat{n}}$, $\hat{B}$, $\hat{D} \in \mathbb{R}^{\hat{m} \times \hat{m}}$ and $\hat{X}$, $\hat{Y} \in \mathbb{R}^{\hat{n} \times \hat{m}}$ with the structure



efficiently.

Existing implementations:

- LAPACK: `DTGSY2` (Level-2) / `DTGSYL` (Level-3),
- SLICOT: `SB04OD` (advanced wrapper around DTGSYL)

Inner Sylvester Equations

We have to solve Generalized Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y},$$

where $\hat{A}$, $\hat{C} \in \mathbb{R}^{\hat{n}\times\hat{n}}$, $\hat{B}$, $\hat{D} \in \mathbb{R}^{\hat{m}\times\hat{m}}$ and $\hat{X}$, $\hat{Y} \in \mathbb{R}^{\hat{n}\times\hat{m}}$ with the structure

Why one should NOT use them:

1. Solve the coupled Generalized Sylvester Equation:

$$\begin{aligned} \check{A}R - L\check{B} &= \check{C} \\ \check{D}R - L\check{E} &= \check{F}. \end{aligned}$$

efficiently.

Existing implementations:

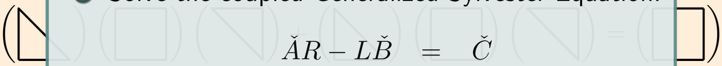
- LAPACK: `DTGSY2` (Level-2) / `DTGSYL` (Level-3),
- SLICOT: `SB04OD` (advanced wrapper around DTGSYL)

Inner Sylvester Equations

We have to solve Generalized Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y},$$

where $\hat{A}$, $\hat{C} \in \mathbb{R}^{\hat{n}\times\hat{n}}$, $\hat{B}$, $\hat{D} \in \mathbb{R}^{\hat{m}\times\hat{m}}$ and $\hat{X}$, $\hat{Y} \in \mathbb{R}^{\hat{n}\times\hat{m}}$ with the structure



efficiently.

Existing implementations:

Why one should NOT use them:

1. Solve the coupled Generalized Sylvester Equation:

$$\begin{aligned} \check{A}R - L\check{B} &= \check{C} \\ \check{D}R - L\check{E} &= \check{F}. \end{aligned}$$

2. Require a different triangular structure.
   $\rightarrow$ Additional work necessary.

- LAPACK: `DTGSY2` (Level-2) / `DTGSYL` (Level-3),
- SLICOT: `SB04OD` (advanced wrapper around DTGSYL)

Inner Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y}$$

Demands on the algorithm to solve the inner Generalized Sylvester Equation:

- The transposition of the matrices $\hat{A}$ and $\hat{C}$ must be done implicitly.

Inner Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y}$$

Demands on the algorithm to solve the inner Generalized Sylvester Equation:

- The transposition of the matrices $\hat{A}$ and $\hat{C}$ must be done implicitly.
- The right hand side $\hat{Y}$ must be overwritten by the solution $\hat{X}$.

Inner Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y}$$

Demands on the algorithm to solve the inner Generalized Sylvester Equation:

- The transposition of the matrices $\hat{A}$ and $\hat{C}$ must be done implicitly.
- The right hand side $\hat{Y}$ must be overwritten by the solution $\hat{X}$.
- The matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ are used read only.

Inner Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y}$$

Demands on the algorithm to solve the inner Generalized Sylvester Equation:

- The transposition of the matrices $\hat{A}$ and $\hat{C}$ must be done implicitly.
- The right hand side $\hat{Y}$ must be overwritten by the solution $\hat{X}$.
- The matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ are used read only.
- At most a cubic (in $\hat{n}$) flop count which does not dominate the flop count of the overall procedure.

Inner Sylvester Equations

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y}$$

Demands on the algorithm to solve the inner Generalized Sylvester Equation:

- The transposition of the matrices $\hat{A}$ and $\hat{C}$ must be done implicitly.
- The right hand side $\hat{Y}$ must be overwritten by the solution $\hat{X}$.
- The matrices $\hat{A}$, $\hat{B}$, $\hat{C}$ and $\hat{D}$ are used read only.
- At most a cubic (in $\hat{n}$) flop count which does not dominate the flop count of the overall procedure.

Two Approaches to solve Generalized Sylvester Equations:

- Extended Bartels-Stewart Method

  [Gardiner, Laub, Amato and Moler, '92]
- Generalized Schur Method

  [Kågström, Westin '89]

Inner Sylvester Equations

Gardiner-Laub Approach

Employing the triangular structure of

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y} \tag{2}$$

we rewrite (2) in terms of the k^{th} column of $\hat{Y}$:

$$\hat{A}^T \sum_{l=1}^{k} \hat{B}_{lk} \hat{X}_{\cdot l} + \hat{C}^T \sum_{l=1}^{k+1} \hat{D}_{lk} \hat{X}_{\cdot l} = \hat{Y}_{\cdot k} \quad \text{for} \quad k = 1, \ldots, m.$$

Inner Sylvester Equations

Gardiner-Laub Approach

Employing the triangular structure of

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y} \tag{2}$$

we rewrite (2) in terms of the k^{th} column of $\hat{Y}$:

$$\hat{A}^T \sum_{l=1}^{k} \hat{B}_{lk} \hat{X}_{\cdot l} + \hat{C}^T \sum_{l=1}^{k+1} \hat{D}_{lk} \hat{X}_{\cdot l} = \hat{Y}_{\cdot k} \quad \text{for} \quad k = 1, \ldots, m.$$

$\hat{D}_{k+1,k} = 0$

$$\left(\hat{B}_{kk} \hat{A} + \hat{D}_{kk} \hat{C}\right)^T \hat{X}_{\cdot k} = \hat{Y}_{\cdot k} - \hat{A}^T \sum_{l=1}^{k-1} \hat{B}_{lk} \hat{X}_{\cdot l} - \hat{C}^T \sum_{l=1}^{k-1} \hat{D}_{lk} \hat{X}_{\cdot l} = \bar{Y}_{\cdot k}$$

Inner Sylvester Equations

Gardiner-Laub Approach

Employing the triangular structure of

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y} \tag{2}$$

we rewrite (2) in terms of the k^{th} column of $\hat{Y}$:

$$\hat{A}^T \sum_{l=1}^{k} \hat{B}_{lk} \hat{X}_{\cdot l} + \hat{C}^T \sum_{l=1}^{k+1} \hat{D}_{lk} \hat{X}_{\cdot l} = \hat{Y}_{\cdot k} \quad \text{for} \quad k = 1, \ldots, m.$$

$\hat{D}_{k+1,k} \neq 0$

$$\begin{pmatrix} \hat{B}_{kk}\hat{A} + \hat{D}_{kk}\hat{C} & \hat{B}_{k,k+1}\hat{A} + \hat{D}_{k,k+1}\hat{C} \\ \hat{B}_{k+1,k+1}\hat{A} + \hat{D}_{k+1,k+1}\hat{C} & \hat{B}_{k+1,k+1}\hat{A} + \hat{D}_{k+1,k+1}\hat{C} \end{pmatrix}^T \begin{pmatrix} \hat{X}_{\cdot k} \\ \hat{X}_{\cdot k+1} \end{pmatrix} = \begin{pmatrix} \bar{Y}_{\cdot k} \\ \bar{Y}_{\cdot k+1} \end{pmatrix}$$

Inner Sylvester Equations

Gardiner-Laub Approach

Employing the triangular structure of

$$\hat{A}^T \hat{X} \hat{B} + \hat{C}^T \hat{X} \hat{D} = \hat{Y} \tag{2}$$

we rewrite (2) in terms of the k^{th} column of $\hat{Y}$:

$$\hat{A}^T \sum_{l=1}^{k} \hat{B}_{lk} \hat{X}_{\cdot l} + \hat{C}^T \sum_{l=1}^{k+1} \hat{D}_{lk} \hat{X}_{\cdot l} = \hat{Y}_{\cdot k} \quad \text{for} \quad k = 1, \ldots, m.$$

$\hat{D}_{k+1,k} \neq 0$

$$\begin{pmatrix} \hat{B}_{kk}\hat{A} + \hat{D}_{kk}\hat{C} & \hat{B}_{k,k+1}\hat{A} + \hat{D}_{k,k+1}\hat{C} \\ \hat{B}_{k+1,k+1}\hat{A} + \hat{D}_{k+1,k+1}\hat{C} & \hat{B}_{k+1,k+1}\hat{A} + \hat{D}_{k+1,k+1}\hat{C} \end{pmatrix}^T \begin{pmatrix} \hat{X}_{\cdot k} \\ \hat{X}_{\cdot k+1} \end{pmatrix} = \begin{pmatrix} \bar{Y}_{\cdot k} \\ \bar{Y}_{\cdot k+1} \end{pmatrix}$$

- Requires $4\hat{n}^2$ extra memory to setup the linear system,
- Additional $2\hat{n}$ extra memory if the leading dimensions of $\hat{X}$ and $\hat{Y}$ are not equal to $\hat{n}$.

Inner Sylvester Equations

Gardiner-Laub Approach – Flop Count

Best Case – $\hat{n} = \hat{m}$ and $D_{k+1,k} = 0 \quad \forall k = 1 \dots \hat{n} - 1$

One solve:

$$F^{(best)}(\hat{n}) = \sum_{k=1}^{\hat{n}} \left(4\hat{n}^2 + 4\hat{n}^2 + \sum_{l=k+1}^{\hat{n}} 4\hat{n} \right) = 10\hat{n}^3 - 2\hat{n}^2$$

Inside Algorithm 1:

$$F_{overall}^{(best)}(n, N_B) = \frac{8}{3}n^3 + 9N_B n^2 + \frac{7}{3}N_B^2 n - 4N_B^3 - n^2 - nN_B$$

Inner Sylvester Equations

Gardiner-Laub Approach – Flop Count

Best Case – $\hat{n} = \hat{m}$ and $D_{k+1,k} = 0 \quad \forall k = 1 \ldots \hat{n} - 1$

One solve:

$$F^{(best)}(\hat{n}) = \sum_{k=1}^{\hat{n}} \left(4\hat{n}^2 + 4\hat{n}^2 + \sum_{l=k+1}^{\hat{n}} 4\hat{n} \right) = 10\hat{n}^3 - 2\hat{n}^2$$

Inside Algorithm 1:

$$F_{overall}^{(best)}(n, N_B) = \frac{8}{3}n^3 + 9N_B n^2 + \frac{7}{3}N_B^2 n - 4N_B^3 - n^2 - nN_B$$

Worst Case – $\hat{n} = \hat{m}$ and $D_{2k-1,2k} \neq 0 \quad \forall k = 1 \ldots \frac{\hat{n}}{2}$

One solve:

$$F^{(worst)}(\hat{n}) = \sum_{k=1}^{\frac{\hat{n}}{2}} \left(10\hat{n}^2 + 4\hat{n}^2 + \frac{47}{2}\hat{n} + 8\hat{n}^2 + \sum_{l=2k+1}^{\hat{n}} 8\hat{n} \right) = 13\hat{n}^3 + \frac{31}{4}\hat{n}^2$$

Inside Algorithm 1:

$$F_{overall}^{(worst)}(n, N_B) = \frac{8}{3}n^3 + \frac{21}{2}N_B n^2 + \frac{23}{6}N_B^2 n + -4N_B^2 + \frac{31}{2}\left(n^2 + N_B n\right)$$

Inner Sylvester Equations

Kågström-Westin Approach

We consider the coupled Sylvester Equation:

$$\begin{aligned} \check{A}R - L\check{B} &= \check{C} \\ \check{D}R - L\check{E} &= \check{F}, \end{aligned}$$

Inner Sylvester Equations

Kågström-Westin Approach

We consider the coupled Sylvester Equation:

$$\begin{aligned} \hat{A}^T R + L\hat{D} &= \hat{Y} \\ \hat{C}^T R - L\hat{B} &= 0 \quad = \hat{W}, \end{aligned} \qquad (\bigstar)$$

where we have to restore the solution $\hat{X}$ trough:

$$\hat{X} = \hat{C}^{-1} L \quad \text{or} \quad \hat{X} = R\hat{B}^{-T}.$$

Inner Sylvester Equations

Kågström-Westin Approach

We consider the coupled Sylvester Equation:

$$\begin{array}{rcll} \hat{A}^T R + L\hat{D} & = & \hat{Y} & \\ \hat{C}^T R - L\hat{B} & = & 0 & = \hat{W}, \end{array} \qquad (\bigstar)$$

where we have to restore the solution $\hat{X}$ trough:

$$\hat{X} = \hat{C}^{-1} L \quad \text{or} \quad \hat{X} = R\hat{B}^{-T}.$$

Forward Substitution Scheme

Partition $(\bigstar)$ into $p \times q$ blocks of size 1×1 or 2×2 and solve

$$\begin{array}{rll} \hat{A}_{ii}^T R_{ij} + L_{ij}\hat{D}_{jj} & = \hat{Y}_{ij} - \sum_{k=1}^{i-1} \hat{A}_{ik}^T R_{kj} - \sum_{k=1}^{j-1} L_{ik}\hat{D}_{kj} & = \tilde{Y}_{ij} \\ \hat{C}_{ii}^T R_{ij} - L_{ij}\hat{B}_{jj} & = \hat{W}_{ij} - \sum_{k=1}^{i-1} \hat{C}_{ik}^T R_{kj} + \sum_{k=1}^{j-1} L_{ik}\hat{B}_{kj} & = \tilde{W}_{ij} \end{array}$$

for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.

Inner Sylvester Equations

Kågström-Westin Approach

We consider the coupled Sylvester Equation:

(★)

where we

Solution of a linear system

$$\begin{pmatrix} I_{\hat{D}} \otimes \hat{A}_{ii}^T & \hat{D}_{jj}^T \otimes I_{\hat{A}} \\ I_{\hat{D}} \otimes \hat{C}_{ii}^T & -\hat{B}_{jj}^T \otimes I_{\hat{A}} \end{pmatrix} \begin{pmatrix} \operatorname{vec}(R_{ij}) \\ \operatorname{vec}(L_{ij}) \end{pmatrix} = \begin{pmatrix} \operatorname{vec}(\tilde{Y}_{ij}) \\ \operatorname{vec}(\tilde{W}_{ij}) \end{pmatrix},$$

which is at most 8×8.

Forward Substitution Scheme

Partition (★) into $p \times q$ blocks of size 1×1 or 2×2 and solve

$$\begin{aligned} \hat{A}_{ii}^T R_{ij} + L_{ij} \hat{D}_{jj} &= \hat{Y}_{ij} - \sum_{k=1}^{i-1} \hat{A}_{ik}^T R_{kj} - \sum_{k=1}^{j-1} L_{ik} \hat{D}_{kj} = \tilde{Y}_{ij} \\ \hat{C}_{ii}^T R_{ij} - L_{ij} \hat{B}_{jj} &= \hat{W}_{ij} - \sum_{k=1}^{i-1} \hat{C}_{ik}^T R_{kj} + \sum_{k=1}^{j-1} L_{ik} \hat{B}_{kj} = \tilde{W}_{ij} \end{aligned}$$

for $i = 1, \ldots, p$ and $j = 1, \ldots, q$.

Inner Sylvester Equations

Kågström-Westin Approach – Flop Count

Best Case – $\hat{n} = \hat{m}$ and $D_{k+1,k} = 0 \quad \forall k = 1 \ldots \hat{n} - 1$

One solve:

$$F^{(best)}(\hat{n}) = \hat{n}^3 + \sum_{i=1}^{\hat{n}} \sum_{j=1}^{\hat{n}} (11 + 4(\hat{n} - i) + 4(\hat{n} - j)) = 5\hat{n}^3 + 7\hat{n}^2$$

Inside Algorithm 1:

$$F^{(best)}_{overall}(n, N_B) = \frac{8}{3}n^3 + \frac{7}{2}n^2 + \frac{13}{2}N_B n^2 - \frac{1}{6}{N_B}^2 n + \frac{7}{2}N_B n - 4N^3$$

Inner Sylvester Equations

Kågström-Westin Approach – Flop Count

Best Case – $\hat{n} = \hat{m}$ and $D_{k+1,k} = 0 \quad \forall k = 1 \dots \hat{n} - 1$

One solve:

$$F^{(best)}(\hat{n}) = \hat{n}^3 + \sum_{i=1}^{\hat{n}} \sum_{j=1}^{\hat{n}} (11 + 4(\hat{n} - i) + 4(\hat{n} - j)) = 5\hat{n}^3 + 7\hat{n}^2$$

Inside Algorithm 1:

$$F^{(best)}_{overall}(n, N_B) = \frac{8}{3}n^3 + \frac{7}{2}n^2 + \frac{13}{2}N_B n^2 - \frac{1}{6}{N_B}^2 n + \frac{7}{2}N_B n - 4N^3$$

Worst Case – $\hat{n} = \hat{m}$ and $D_{2k-1,2k} \neq 0 \quad \forall k = 1 \dots \frac{\hat{n}}{2}$

One solve:

$$F^{(worst)}(\hat{n}) = \hat{n}^3 + \sum_{i=1}^{\frac{\hat{n}}{2}} \sum_{j=1}^{\frac{\hat{n}}{2}} \left(415 + 32(\frac{\hat{n}}{2} - i) + 32(\frac{\hat{n}}{2} - j)\right) = 5\hat{n}^3 + \frac{383}{4}\hat{n}^2$$

Inside Algorithm 1:

$$F^{(worst)}_{overall}(n, N_B) = \frac{8}{3}n^3 + \frac{383}{8}n^2 + \frac{13}{2}N_B n^2 + \frac{383}{8}N_B n - \frac{1}{6}{N_B}^2 n - 4{N_B}^3$$

Inner Sylvester Equations

Diagonal Blocks in the Outer Algorithm

All diagonal blocks result in

$$A_{ll}^T X_{ll} E_{ll} + E_{ll}^T X_{ll} A_{ll} = \hat{Y}_{ll},$$

which is a Generalized Lyapunov Equation again.

Inner Sylvester Equations

Diagonal Blocks in the Outer Algorithm

All diagonal blocks result in

$$A_{ll}^T X_{ll} E_{ll} + E_{ll}^T X_{ll} A_{ll} = \hat{Y}_{ll},$$

which is a Generalized Lyapunov Equation again.

We have to preserve the symmetry of X_{ll}.

Inner Sylvester Equations

Diagonal Blocks in the Outer Algorithm

All diagonal blocks result in

$$A_{ll}^T X_{ll} E_{ll} + E_{ll}^T X_{ll} A_{ll} = \hat{Y}_{ll},$$

which is a Generalized Lyapunov Equation again.

We have to preserve the symmetry of X_{ll}.

1. Copy the lower(upper) triangle of X_{ll} to the upper(lower) triangle.

Inner Sylvester Equations

Diagonal Blocks in the Outer Algorithm

All diagonal blocks result in

$$A_{ll}^T X_{ll} E_{ll} + E_{ll}^T X_{ll} A_{ll} = \hat{Y}_{ll},$$

which is a Generalized Lyapunov Equation again.

We have to preserve the symmetry of X_{ll}.

1. Copy the lower(upper) triangle of X_{ll} to the upper(lower) triangle.
2. Symmetrize the block by

$$\frac{1}{2}\left(X_{ll} + X_{ll}^T\right) \rightarrow X_{ll}.$$

Inner Sylvester Equations

Diagonal Blocks in the Outer Algorithm

All diagonal blocks result in

$$A_{ll}^T X_{ll} E_{ll} + E_{ll}^T X_{ll} A_{ll} = \hat{Y}_{ll},$$

which is a Generalized Lyapunov Equation again.

We have to preserve the symmetry of X_{ll}.

1. Copy the lower(upper) triangle of X_{ll} to the upper(lower) triangle.
2. Symmetrize the block by
$$\frac{1}{2}\left(X_{ll} + X_{ll}^T\right) \to X_{ll}.$$
3. Solve using Algorithm 1 again with a smaller block size.
   - Reduces the flop count for those blocks to $\mathcal{O}(\frac{8}{3}N_B^3)$.
   - But we get a recursive scheme.

Numerical Results

Hardware:

- Intel® Xeon® X5650, 2×6 Cores, 2×24 GB DDR3 RAM

Software:

- Intel® Fortran Compiler 13
- Intel® MKL 11
- SLICOT 5.0

Numerical Results

Hardware:

- Intel® Xeon® X5650, 2×6 Cores, 2×24 GB DDR3 RAM

Software:

- Intel® Fortran Compiler 13
- Intel® MKL 11
- SLICOT 5.0

- Performance tests: random matrices via subsequent calls of `DLARNV`,
- Accuracy/Reliability:

$$A = (2^{-t} - 1)I_n + \operatorname{diag}(1, 2, \ldots, n) + U_n$$
$$E = I_n + 2^{-t}U_n,$$

where I_n is the identity and U_n is an $n \times n$ matrix with only unit entries above the diagonal and varying t.

Numerical Results

Runtime

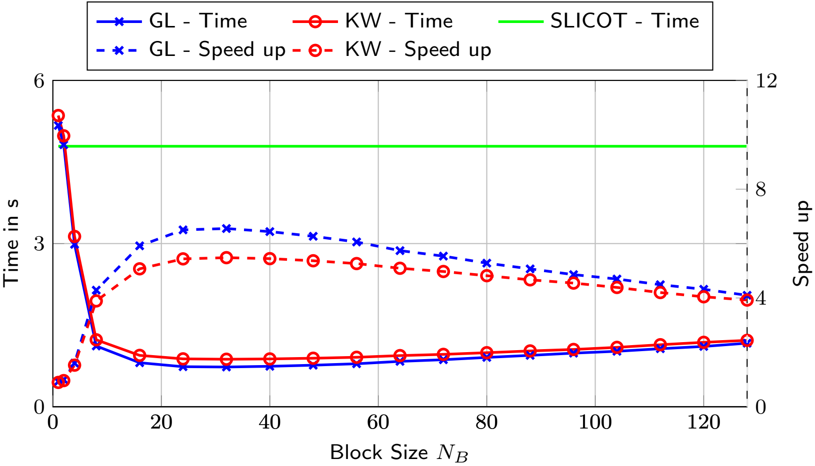


Figure: Runtime and speed up, $n = 1000$, single thread.
(GL = Gardiner and Laub, KW = Kågström and Westin)

Numerical Results

Runtime

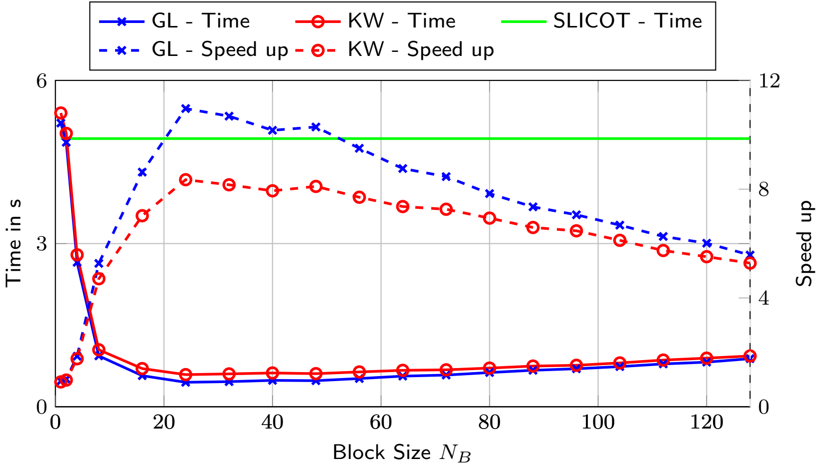


Figure: Runtime and speed up, $n = 1000$, 12 threads.
(GL = Gardiner and Laub, KW = Kågström and Westin)

Numerical Results

Accuracy

N_B	Relative Residual $t=0$	Relative Residual $t=30$	Relative Residual $t=40$	Relative Forward Error $t=0$	Relative Forward Error $t=30$	Relative Forward Error $t=40$
SLICOT						
1	0.00	0.00	1.07e-15	3.54e-17	4.18e-17	5.97e-14
Gardiner-Laub						
8	0.00	0.00	5.24e-16	0.00	0.00	3.00e-14
24	0.00	0.00	4.15e-16	0.00	0.00	2.23e-14
48	0.00	0.00	4.17e-16	0.00	0.00	2.07e-14
Kågström-Westin						
8	1.62e-16	1.56e-16	5.14e-16	2.15e-14	2.14e-14	3.03e-14
24	1.78e-16	1.73e-16	4.10e-16	2.71e-14	2.72e-14	3.15e-14
48	2.02e-16	2.03e-16	3.83e-16	8.17e-14	8.07e-14	8.82e-14

Table: Relative residual and relative forward error for the generalized Lyapunov equation, artificial example $n = 1000$.

Numerical Results

Symmetrization of the Diagonal Blocks

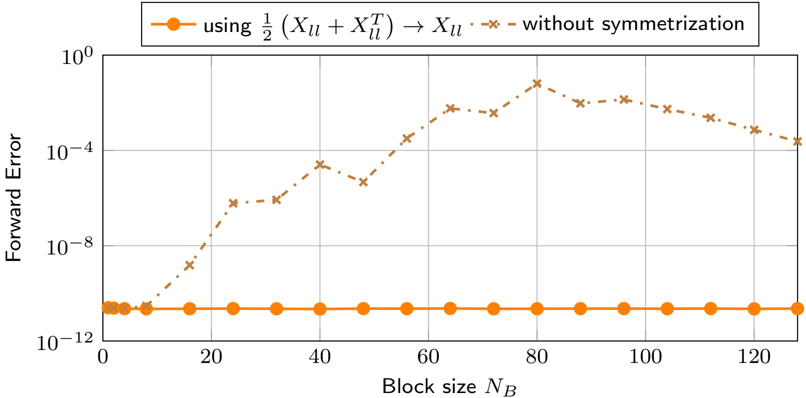


Figure: Forward error of the Gardiner-Laub approach with and without symmetrization the diagonal blocks.

Conclusions

Conclusions

- We rearranged the existing idea to level-3 BLAS algorithm.
- We improved two variants of existing solution techniques for the Generalized Sylvester Equations to fit the requirements of the Lyapunov solver.
- A speed-up of at least 6.5 (sequential) or 10.5 (parallel) is possible without loosing accuracy.

Conclusions

Conclusions

- We rearranged the existing idea to level-3 BLAS algorithm.
- We improved two variants of existing solution techniques for the Generalized Sylvester Equations to fit the requirements of the Lyapunov solver.
- A speed-up of at least 6.5 (sequential) or 10.5 (parallel) is possible without loosing accuracy.

Outlook

- Develop accelerator based variants (NVIDIA® CUDA, Intel® Xeon® Phi),
- Apply similar ideas to Hammarling's-Method.
- Accelerate the necessary QZ decomposition in the overall algorithm.

Conclusions

Conclusions

- We rearranged the existing idea to level-3 BLAS algorithm.
- We improved two variants of existing solution techniques for the Generalized Sylvester Equations to fit the requirements of the Lyapunov solver.
- A speed-up of at least 6.5 (sequential) or 10.5 (parallel) is possible without loosing accuracy.

Outlook

- Develop accelerator based variants (Nvidia® CUDA, Intel® Xeon® Phi),
- Apply similar ideas to Hammarling's-Method.
- Accelerate the necessary QZ decomposition in the overall algorithm.

Thank you for your attention!

Questions?