MAX-PLANCK-GESELLSCHAFT

Peter Benner   Patrick Kürschner   Jens Saak

Real versions of low-rank ADI methods with complex shifts

MAX–PLANCK–INSTITUT
FÜR DYNAMIK KOMPLEXER
TECHNISCHER SYSTEME
MAGDEBURG

Max Planck Institute Magdeburg Preprints

MPIMD/12-11   May 30, 2012

Impressum:

Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg

Publisher:
Max Planck Institute for Dynamics of Complex Technical Systems

Address:
Max Planck Institute for Dynamics of Complex Technical Systems
Sandtorstr. 1
39106 Magdeburg

www.mpi-magdeburg.mpg.de/preprints

Real versions of low-rank ADI methods with complex shifts

Peter Benner, Patrick Kürschner, and Jens Saak

I. Introduction

We consider the numerical solution of large-scale generalized Lyapunov equations

$$AXE^T + EXA^T = -BB^T \tag{1}$$

with $A,\ E \in \mathbb{R}^{n\times n}$ large and sparse, nonsingular, $\Lambda(A,E) \subset \mathbb{C}_-$ and $B \in \mathbb{R}^{n\times m}$, $m \ll n$. Due to the low rank of the right hand side, we are interested in approximating the solution of (1) via low-rank factors $Z \in \mathbb{R}^{n\times r}$, $r \ll n$ such that $ZZ^T \approx X \in \mathbb{R}^{n\times n}$. The generalized low-rank ADI method (G-LR-ADI) [7], [14], [1] is the method of our choice for this purpose. It constructs the following low-rank factors $Z = [V_1,\ldots,V_{j_{\max}}]$, where the V_j are obtained from the iteration

$$\begin{aligned} V_1 &= \sqrt{-2\operatorname{Re}(p_1)}(A+p_1E)^{-1}B,\\ V_j &= \sqrt{\tfrac{\operatorname{Re}(p_j)}{\operatorname{Re}(p_{j-1})}}\left[I-(p_j+\overline{p_{j-1}})(A+p_jE)^{-1}E\right]V_{j-1},\ j>1. \end{aligned}$$

We assume that the occurring linear systems involving $A + p_jE$ can be solved efficiently with sparse-direct [11], [9] or iterative methods [15]. The set of optimal shift parameters are related to a rational minmax problem which involves the complete spectrum $\Lambda(A,E)$. Since all eigenvalues will, in a large-scale setting, in general not be easily available, a cheap heuristic approach has been proposed in [14] which uses a small number of Ritz values as approximate eigenvalues to solve the minmax problem in an approximate sense.

Our main concern is the case when $\Lambda(A,E)$ contains complex eigenvalues. Hence, the shift parameters will be complex as well and G-LR-ADI will employ complex arithmetic and storage and produce complex low-rank factors. From a numerical and practical point of view this is undesirable. This technical note is concerned with G-LR-ADI versions that compute real low-rank factors by using only real, or only an absolutely necessary amount of complex arithmetic operations. Its main purpose is to give detailed implementations as pseudo code which have been left out in references regarding this issue [14], [13], [7], [6], [5]. All these methods require that complex shifts occur in pairs $p_j,\ p_{j+1} = \overline{p_j}$. This alone ensures that ZZ^H is a real matrix even if the low-rank factor is complex, provided that the method is not stopped in between a complex pair of shifts.

II. Completely or partially real variants of G-LR-ADI computing real low-rank factors

A. *Concatenating two complex into one real iteration steps via "squaring".*

For standard Lyapunov equation $(E = I_n)$ an approach is proposed in [14], [13], [7] that exploits the identity

$$(A \pm pI_n)(A \pm \overline{p}I_n) = A^2 + 2\operatorname{Re}(p)A + |p|^2I_n$$

to derive a reformulation of LR-ADI that avoids complex arithmetic completely at the price of squaring the matrix A. This might lead to numerical problem when solving the involved linear system with sparse-direct or iterative solvers. This variant is usually referred to as LR-ADI-R and a complete algorithm can be found in [7, Algorithm 4]. A derivation of the involved formulas is given in the appendix of [6].

For generalized Lyapunov equations we have, exemplary for the first two iterations and $p_1,\ p_2 = \overline{p_1}$:

$$\begin{aligned} V_2 &= \left[I-2\overline{p_{j-1}}(A+p_jE)^{-1}E\right]V_1\\ &= \sqrt{-2\operatorname{Re}(p_1)}(A+\overline{p_1}E)^{-1}(A-\overline{p_1}E)(A+p_1E)^{-1}B\\ &= \sqrt{-2\operatorname{Re}(p_1)}(E^{-1}A-\overline{p_1}I_n)(E^{-1}A+\overline{p_1}I_n)^{-1}(A+p_1E)^{-1}B\\ &= \sqrt{-2\operatorname{Re}(p_1)}(E^{-1}A-\overline{p_1}I_n)(AE^{-1}A+2\operatorname{Re}(p_1)A+|p_1|^2E)^{-1}B\\ &= -\overline{p_1}\sqrt{-2\operatorname{Re}(p_1)}\tilde{V}_1+\sqrt{-2\operatorname{Re}(p_1)}\tilde{V}_2,\\ \tilde{V}_1 &:= (AE^{-1}A+2\operatorname{Re}(p_1)A+|p_1|^2E)^{-1}B,\ \tilde{V}_2 := E^{-1}A\tilde{V}_1 \in \mathbb{R}^{n\times m}. \end{aligned}$$

Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstraße 1, 39106 Magdeburg, Germany, {benner, kuerschner, saak}@mpi-magdeburg.mpg.de, Phone: +49 391 6110 {450, 424}, Fax: +49 391 6110 453

The first $2m$ columns of the low-rank factor are then given by

$$Z_2 = [2\sqrt{-\operatorname{Re}(p_1)}|p_1|\tilde{V}_1,\ 2\sqrt{-\operatorname{Re}(p_1)}\tilde{V}_2].$$

The generalization of LR-ADI-R to (1) leads to a completely real G-LR-ADI (G-LR-ADI-R) version given in Algorithm 1. The obvious weak point of this approach is that forming the coefficient matrix for the linear system to be solved requires to build $AE^{-1}A$ which can, depending on the structure of E, easily become dense. Hence, except when E is an easy matrix (e.g., diagonal), this real formulation of G-LR-ADI is inefficient in a large-scale setting. Note also that for each complex pair, an additional solve with E is required which introduces even higher computation costs.

Algorithm 1: G-LR-ADI-R

Input : A, E and B as in (1) and shift parameters $\{p_1, \dots, p_{j_{\max}}\}$.
Output: $Z = Z_{j_{\max}} \in \mathbb{R}^{n \times t_{j_{\max}}}$, such that $ZZ^T \approx P$

for $j = 1, 2, \dots, j_{\max}$ **do**
  if $j = 1$ **then**
    if p_1 *is real* **then**
      $\tilde{V}_1 = (A + p_1 E)^{-1}B,\ Z_1 = \sqrt{-2\operatorname{Re}(p_1)}\tilde{V}_1;$
    else
      $\tilde{V}_1 := (AE^{-1}A + 2\operatorname{Re}(p_1)A + |p_1|^2E)^{-1}B,\ \tilde{V}_2 := E^{-1}A\tilde{V}_1;$
      $Z_2 = [2\sqrt{-\operatorname{Re}(p_1)}|p_1|\tilde{V}_1,\ 2\sqrt{-\operatorname{Re}(p_1)}\tilde{V}_2],\ j = j + 1;$
  else
    if p_j *is real* **then**
      if p_{j-1} *is real* **then**
        $\tilde{V}_j = \tilde{V}_{j-1} - (p_j + p_{j-1})(A + p_jE)^{-1}(E\tilde{V}_{j-1});$
      else
        $\tilde{V}_j = \tilde{V}_{j-1} - \operatorname{Re}(2p_{j-1} + p_j)\tilde{V}_{j-2} + (|p_{j-1}|^2 + 2p_j\operatorname{Re}(p_{j-1}) + p_j^2)(A + p_jE)^{-1}(E\tilde{V}_{j-2})$;
      $Z_j = [Z_{j-1},\ \sqrt{-2\operatorname{Re}(p_j)}\tilde{V}_j];$
    else
      if p_{j-1} *is real* **then**
        $\tilde{V}_j := (AE^{-1}A + 2\operatorname{Re}(p_j)A + |p_j|^2E)^{-1}(A\tilde{V}_{j-1} - p_{j-1}E\tilde{V}_{j-1})$
      else
        $\tilde{V}_j := \tilde{V}_{j-2} + (AE^{-1}A + 2\operatorname{Re}(p_1)A + |p_j|^2E)^{-1}((|p_{j-1}|^2 - |p_j|^2)\tilde{V}_{j-2} - 2\operatorname{Re}(p_j + p_{j-1})\tilde{V}_{j-1});$
      $\tilde{V}_{j+1} = E^{-1}A\tilde{V}_j;$
      $Z_{j+1} = [Z_{j-1},\ 2\sqrt{-\operatorname{Re}(p_j)}|p_j|\tilde{V}_j,\ 2\sqrt{-\operatorname{Re}(p_j)}\tilde{V}_{j+1}],\ j = j + 1;$

B. Exploiting the interconnection of ADI iterates with respect to complex shifts

Based on [6], an efficient modified LR-ADI method generating real low-rank factors but keeping the amount of complex arithmetic operations and storage at an absolutely necessary minimum can be derived. The key ingredient is that for a pair $p_j,\ p_{j+1} = \overline{p_j}$ of shifts, the $j+1$-th iterate is given by

$$V_{j+1} = \overline{V_j} - 2\tfrac{\operatorname{Re}(p_j)}{\operatorname{Im}(p_j)}\operatorname{Im}(V_j).$$

A version of G-LR-ADI utilizing this strategy is given in Algorithm 2. For the case $E = I_n$ we refer to [6, Algorithm 3]. There, at the cost of keeping the linear system and the corresponding iterate complex, the algorithm produces real low-rank factors. Advantages are that no expensive coefficient matrices are generated and that only one complex linear system has to be solved for each complex pair of shifts. The next approach aims at removing these remaining tasks which require complex arithmetic.

C. Avoiding complex arithmetic operations via equivalent real formulations.

Consider Step 3 of Algorithm 2 for $p_1 \in \mathbb{C}_-$. By splitting V_1 into its real and imaginary part, and rewriting the complex linear system into an equivalent real formulation [10], this step becomes

$$\begin{bmatrix} A + \operatorname{Re}(p_1)E & -\operatorname{Im}(p_1)E \\ \operatorname{Im}(p_1)E & A + \operatorname{Re}(p_1)E \end{bmatrix} \begin{bmatrix} \operatorname{Re}(V_1) \\ \operatorname{Im}(V_1) \end{bmatrix} = \begin{bmatrix} \sqrt{-2\operatorname{Re}(p_1)}B \\ 0 \end{bmatrix} \tag{2}$$

Algorithm 2: G-LR-ADI-pR

Input : A, E and B as in (1) and shift parameters $\{p_1, \dots, p_{j_{\max}}\}$.
Output: $Z = Z_{j_{\max}} \in \mathbb{R}^{n \times t_{j_{\max}}}$, such that $ZZ^T \approx P$

1 **for** $j = 1, 2, \dots, j_{\max}$ **do**
2   **if** $j = 1$ **then**
3   Solve $(A + p_1 E)V_1 = \sqrt{-2\operatorname{Re}(p_1)} B$ for V_1;
4   **else**
5   Solve $(A + p_j E)\tilde{V} = EV_{j-1}$ for $\tilde{V}$;
6     $V_j = \sqrt{\operatorname{Re}(p_j)/\operatorname{Re}(p_{j-1})}(V_{j-1} - (p_j + \overline{p_{j-1}})\tilde{V})$;
7   **if** $\operatorname{Im}(p_j) = 0$ **then**
8     $V_j = \operatorname{Re}(V_j)$;
9     $Z_j = [Z_{j-1},\ V_j]$;
10   **else**
11     $\beta = 2\frac{\operatorname{Re}(p_j)}{\operatorname{Im}(p_j)}$;
12     $V_{j+1} = \overline{V_j} + \beta \operatorname{Im}(V_j)$;
13     $Z_{j+1} = \left[Z_{j-1},\ \sqrt{2}\operatorname{Re}(V_j) + \frac{\beta}{\sqrt{2}}\operatorname{Im}(V_j),\ \sqrt{\frac{\beta^2}{2} + 2} \cdot \operatorname{Im}(V_j)\right]$;
14   Set $j = j + 1$;

and similarly, the complex linear system in Step 5 can be written as

$$\begin{bmatrix} A + \operatorname{Re}(p_j)E & -\operatorname{Im}(p_j)E \\ \operatorname{Im}(p_j)E & A + \operatorname{Re}(p_j)E \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} = \begin{bmatrix} E\operatorname{Re}(V_{j-1}) \\ E\operatorname{Im}(V_{j-1}) \end{bmatrix}. \tag{3}$$

For real and imaginary part of increment V_j this finally leads to [5]

$$\begin{bmatrix} \operatorname{Re}(V_j) \\ \operatorname{Im}(V_j) \end{bmatrix} = \sqrt{\frac{\operatorname{Re}(p_j)}{\operatorname{Re}(p_{j-1})}} \left(\begin{bmatrix} \operatorname{Re}(V_{j-1}) \\ \operatorname{Im}(V_{j-1}) \end{bmatrix} - \begin{bmatrix} (\operatorname{Re}(p_j)+\operatorname{Re}(p_{j-1}))I_m & (-\operatorname{Im}(p_j)+\operatorname{Im}(p_{j-1}))I_m \\ (\operatorname{Im}(p_j)-\operatorname{Im}(p_{j-1}))I_m & (\operatorname{Re}(p_j)+\operatorname{Re}(p_{j-1}))I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} \right). \tag{4}$$

In Algorithm 2 the following sequences of shift parameters are possible:

1) $p_j,\ p_{j-1}$ are both real,
2) $p_j \in \mathbb{C},\ p_{j-1} \in \mathbb{R}$,
3) $p_j \in \mathbb{C},\ p_{j-1} = \overline{p_{j-2}} \in \mathbb{C}$,
4) $p_j \in \mathbb{R},\ p_{j-1} = \overline{p_{j-2}} \in \mathbb{C}$.

All four cases lead to slightly different version of (4) and an the resulting novel completely real G-LR-ADI is given in Algorithm 3. In order to be in line with the notation in [5] we use the abbreviation G-LR-ADI-R2 for this version. Although the linear system to be solved are augmented to dimension $2n$, their coefficient matrix is still sparse and hence they are preferable over the ones in G-LR-ADI-R (Algorithm 1).

Remark 1. Note that, depending on existing symmetries of A, E and the choice of an iterative solver, choosing another equivalent linear system [10] might be appropriate. For instance, in the case when $A = A^T$ and $E = E^T$ indefinite, the coefficient matrices of the equivalent augmented real systems

$$\begin{bmatrix} A+\operatorname{Re}(p_j)E & \operatorname{Im}(p_j)E \\ \operatorname{Im}(p_j)E & -A-\operatorname{Re}(p_j)E \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ -\operatorname{Im}(\tilde{V}) \end{bmatrix} = \begin{bmatrix} E\operatorname{Re}(V_{j-1}) \\ E\operatorname{Im}(V_{j-1}) \end{bmatrix},$$

$$\begin{bmatrix} \operatorname{Im}(p_j)E & A+\operatorname{Re}(p_j)E \\ A+\operatorname{Re}(p_j)E & -\operatorname{Im}(p_j)E \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} = \begin{bmatrix} E\operatorname{Im}(V_{j-1}) \\ E\operatorname{Re}(V_{j-1}) \end{bmatrix}$$

are symmetric, allowing the use of short-recurrence methods, e.g., MINRES for its solution. This might be more economical compared to the augmented system in (3) which require nonsymmetric solvers. The drawback is that the spectrum of the above symmetric augmented matrices is symmetric with respect to the origin which can deteriorate the convergence of the applied Krylov subspace method. Also note that the original coefficient matrix $A + p_j E$ of the original complex system is in fact complex symmetric in this case, such that specialized iterative solvers can be applied, e.g., [12].

Remark 2. If the occurring linear systems are indeed solved by iterative Krylov subspace methods, preconditioning should be considered as well. Although this might be straightforward for the original complex linear systems with $A+p_j E$ in the original and partially real version of (G-)LR-ADI, e.g. using incomplete factorization, little is known how to construct a preconditioner for the linear systems arising in G-LR-ADI-R. For the augmented real linear systems in the novel completely real version G-LR-ADI-R2, some ideas for preconditioners can be found, e.g., in [10, Section 2].

Algorithm 3: G-LR-ADI-R2

Input : A, E and B as in (1) and shift parameters $\{p_1, \ldots, p_{j_{\max}}\}$.
Output: $Z = Z_{j_{\max}} \in \mathbb{R}^{n \times t_{j_{\max}}}$, such that $ZZ^T \approx P$

for $j = 1, 2, \ldots, j_{\max}$ **do**
  if $j = 1$ **then**
    if p_1 *is real* **then**
      $\operatorname{Re}(V_1) = \sqrt{-2\operatorname{Re}(p_1)}(A + p_1 E)^{-1} B$;
    else
      $\begin{bmatrix} \operatorname{Re}(V_1) \\ \operatorname{Im}(V_1) \end{bmatrix} = \begin{bmatrix} A+\operatorname{Re}(p_1)E & -\operatorname{Im}(p_1)E \\ \operatorname{Im}(p_1)E & A+\operatorname{Re}(p_1)E \end{bmatrix}^{-1} \begin{bmatrix} \sqrt{-2\operatorname{Re}(p_1)}B \\ 0 \end{bmatrix}$;
  else
    if p_j *is real* **then**
      if p_{j-1} *is real* **then**
        $\tilde{V} = (A + p_j E)^{-1}(EV_{j-1})$;
        $\operatorname{Re}(V_j) = \sqrt{p_j/p_{j-1}}(\operatorname{Re}(V_{j-1}) - (p_j + p_{j-1})\tilde{V})$;
      else
        $[\operatorname{Re}(\tilde{V}), \operatorname{Im}(\tilde{V})] = (A + p_j E)^{-1}[E\operatorname{Re}(V_{j-1}), E\operatorname{Im}(V_{j-1})]$;
        $\operatorname{Re}(V_j) = \sqrt{\frac{p_j}{\operatorname{Re}(p_{j-2})}}\Big(\operatorname{Re}(V_{j-1}) - (p_j + \operatorname{Re}(p_{j-2}))\operatorname{Re}(\tilde{V}) + \operatorname{Im}(p_{j-2})\operatorname{Im}(\tilde{V})\Big)$;
    else
      if p_{j-1} *is real* **then**
        $\begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} = \begin{bmatrix} A+\operatorname{Re}(p_j)E & -\operatorname{Im}(p_j)E \\ \operatorname{Im}(p_j)E & A+\operatorname{Re}(p_j)E \end{bmatrix}^{-1} \begin{bmatrix} E\operatorname{Re}(V_{j-1}) \\ 0 \end{bmatrix}$;
        $\begin{bmatrix} \operatorname{Re}(V_j) \\ \operatorname{Im}(V_j) \end{bmatrix} = \sqrt{\frac{\operatorname{Re}(p_j)}{p_{j-1}}}\left(\begin{bmatrix} \operatorname{Re}(V_{j-1}) \\ 0 \end{bmatrix} - \begin{bmatrix} (\operatorname{Re}(p_j)+p_{j-1})I_m & -\operatorname{Im}(p_j)I_m \\ \operatorname{Im}(p_j)I_m & (\operatorname{Re}(p_j)+p_{j-1})I_m \end{bmatrix}\begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix}\right)$;
      else
        $\begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} = \begin{bmatrix} A+\operatorname{Re}(p_j)E & -\operatorname{Im}(p_j)E \\ \operatorname{Im}(p_j)E & A+\operatorname{Re}(p_j)E \end{bmatrix}^{-1} \begin{bmatrix} E\operatorname{Re}(V_{j-1}) \\ E\operatorname{Im}(V_{j-1}) \end{bmatrix}$;
        $\begin{bmatrix} \operatorname{Re}(V_j) \\ \operatorname{Im}(V_j) \end{bmatrix} = \sqrt{\frac{\operatorname{Re}(p_j)}{\operatorname{Re}(p_{j-2})}}\left(\begin{bmatrix} \operatorname{Re}(V_{j-1}) \\ \operatorname{Im}(V_{j-1}) \end{bmatrix} - \begin{bmatrix} (\operatorname{Re}(p_j)+\operatorname{Re}(p_{j-2}))I_m & -(\operatorname{Im}(p_j)+\operatorname{Im}(p_{j-2}))I_m \\ (\operatorname{Im}(p_j)+\operatorname{Im}(p_{j-2}))I_m & (\operatorname{Re}(p_j)+\operatorname{Re}(p_{j-2}))I_m \end{bmatrix}\begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix}\right)$;
  if p_j *is real* **then**
    $Z_j = [Z_{j-1}, \ \operatorname{Re}(V_j)]$;
  else
    $\beta = 2\frac{\operatorname{Re}(p_j)}{\operatorname{Im}(p_j)}$;
    $[\operatorname{Re}(V_{j+1}), \ \operatorname{Im}(V_{j+1})] = [\operatorname{Re}(V_j) + \beta\operatorname{Im}(V_j), \ -\operatorname{Im}(V_j)]$;
    $Z_{j+1} = \left[Z_{j-1}, \ \sqrt{2}\operatorname{Re}(V_j) + \frac{\beta}{\sqrt{2}}\operatorname{Im}(V_j), \ \sqrt{\frac{\beta^2}{2} + 2}\cdot\operatorname{Im}(V_j)\right]$;
    Set $j = j + 1$;

Remark 3. Note that (3) is equivalent to the generalized Sylvester equation

$$AY + EYC = EF \tag{5}$$

with

$$Y := [\operatorname{Re}(\tilde{V}), \ \operatorname{Im}(\tilde{V})] \in \mathbb{R}^{n\times 2m}, \ C := \begin{bmatrix} \operatorname{Re}(p_j)I_m & \operatorname{Im}(p_j)I_m \\ -\operatorname{Im}(p_j)I_m & \operatorname{Re}(p_j)I_m \end{bmatrix} \in \mathbb{R}^{2m\times 2m}, \ F := [\operatorname{Re}(V_{j-1}), \ \operatorname{Im}(V_{j-1})] \in \mathbb{R}^{n\times 2m}.$$

Reordering (3) such that the solution vector is of the form

$$\begin{bmatrix} \operatorname{Re}(\tilde{V}_{1,:}) \\ \operatorname{Im}(\tilde{V}_{1,:}) \\ \vdots \\ \operatorname{Re}(\tilde{V}_{n,:}) \\ \operatorname{Im}(\tilde{V}_{n,:}) \end{bmatrix}$$

will introduce appropriately reordered matrices $\hat{Y}$, $\hat{F}$ in (5) and $C = \begin{bmatrix} \operatorname{Re}(p_j) & \operatorname{Im}(p_j) \\ -\operatorname{Im}(p_j) & \operatorname{Re}(p_j) \end{bmatrix} \otimes I_m$ becomes block-diagonal:

$$\hat{C} := I_m \otimes \begin{bmatrix} \operatorname{Re}(p_j) & \operatorname{Im}(p_j) \\ -\operatorname{Im}(p_j) & \operatorname{Re}(p_j) \end{bmatrix}.$$

Since each diagonal block in $\hat{C}$ is of size 2×2, the reordered generalized Sylvester equation can be solved by sequence of m generalized Sylvester equations:

$$A[\operatorname{Re}(\tilde{V}_{:,k}), \operatorname{Im}(\tilde{V}_{:,k})] + E[\operatorname{Re}(\tilde{V}_{:,k}), \operatorname{Im}(\tilde{V}_{:,k})] \left[\begin{smallmatrix} \operatorname{Re}(p_j) & \operatorname{Im}(p_j) \\ -\operatorname{Im}(p_j) & \operatorname{Re}(p_j) \end{smallmatrix}\right] = E[\operatorname{Re}(V_{j-1}), \operatorname{Im}(V_{j-1})][e_k, e_k], \;\; k = 1, \dots, m.$$

Similarly structured equations arise in a solution method [3] for sparse-dense Sylvester equation $MXG + NXH = W$ with $M, H \in \mathbb{R}^{n \times n}$ large and sparse, $G, H \in \mathbb{R}^{r \times r}$ dense and $r \ll n$ when a generalized real Schur decomposition of (G, H) is used [3].

D. Relation between both completely real versions

Now we investigate how the connection of the iterates $\tilde{V}_j$ of G-LR-ADI-R (Algorithm 1) with $\operatorname{Re}(V_j),\ \operatorname{Im}(V_j)$ of G-LR-ADI-R2 (Algorithm 3). Consider exemplary the first two G-LR-ADI iteration for $p_1,\ p_2 = \overline{p_1} \in \mathbb{C}_-$. With $\xi := \sqrt{-2\operatorname{Re}(p_1)}$, we have in G-LR-ADI-R2 that

$$\left[\begin{smallmatrix} \operatorname{Re}(V_1) \\ \operatorname{Im}(V_1) \end{smallmatrix}\right] = \left[\begin{smallmatrix} A+\operatorname{Re}(p_1)E & -\operatorname{Im}(p_1)E \\ \operatorname{Im}(p_1)E & A+\operatorname{Re}(p_1)E \end{smallmatrix}\right]^{-1} \left[\begin{smallmatrix} \xi B \\ 0 \end{smallmatrix}\right]$$

which leads to

$$\operatorname{Re}(V_1) = \tfrac{-1}{\operatorname{Im}(p_1)}(E^{-1}A + \operatorname{Re}(p_1)I_n)\operatorname{Im}(V_1) = \xi\operatorname{Re}(p_1)\tilde{V}_1 + \xi\tilde{V}_2, \tag{6}$$

$$\operatorname{Im}(V_1) = -\operatorname{Im}(p_1)\xi(AE^{-1}A + 2\operatorname{Re}(p_1)A + |p_1|^2E)^{-1}B = -\operatorname{Im}(p_1)\xi\tilde{V}_1, \tag{7}$$

where

$$\tilde{V}_1 := (AE^{-1}A + 2\operatorname{Re}(p_1)A + |p_1|^2E)^{-1}B, \quad \tilde{V}_2 := E^{-1}A\tilde{V}_1$$

as in G-LR-ADI-R, such that $\operatorname{Re}(V_1),\ \operatorname{Im}(V_1)$ are just linear combinations of $\tilde{V}_1,\ \tilde{V}_2$. Now G-LR-ADI-R2 proceeds by constructing the second iterate as

$$\left[\begin{smallmatrix} \operatorname{Re}(V_2) \\ \operatorname{Im}(V_2) \end{smallmatrix}\right] = \left[\begin{smallmatrix} \operatorname{Re}(V_1)+\beta\operatorname{Im}(V_1) \\ -\operatorname{Im}(V_1) \end{smallmatrix}\right] = \left[\begin{smallmatrix} -\xi\operatorname{Re}(p_1)\tilde{V}_1+\xi\tilde{V}_2 \\ \operatorname{Im}(p_1)\xi\tilde{V}_1 \end{smallmatrix}\right]$$

and the low-rank factor of these first two iterations is given by

$$\begin{aligned}
[Z_1,\ Z_2]^{R2} = [V_1,\ V_2] &= \left[\sqrt{2}\operatorname{Re}(V_1) + \tfrac{2\operatorname{Re}(p_1)}{\sqrt{2}\operatorname{Im}(p_1)}\operatorname{Im}(V_1),\ -\sqrt{2\tfrac{\operatorname{Re}(p_1)^2}{\operatorname{Im}(p_1)^2} + 2}\xi\operatorname{Im}(V_1)\right] \\
&= \left[\sqrt{2}\xi\tilde{V}_2,\ -\sqrt{2}|p_1|\tilde{V}_1\right] = \left[2\sqrt{-\operatorname{Re}(p_1)}\xi\tilde{V}_2,\ 2\sqrt{-\operatorname{Re}(p_1)}|p_1|\tilde{V}_1\right] \\
&= [Z_1,\ Z_2]^R \left[\begin{smallmatrix} 0 & -I_m \\ I_m & 0 \end{smallmatrix}\right],
\end{aligned}$$

revealing that the R and R2 variant add new blocks with respect to a pair of complex shifts in reversed order to the low-rank factor Z. Since $\left[\begin{smallmatrix} 0 & -I_m \\ I_m & 0 \end{smallmatrix}\right]$ is orthogonal, this does obviously not lead to any difference in the computed low-rank solution in the end. It is possible to show that similar relations as (6), (7) hold also regarding $\operatorname{Re}(V_j),\ \operatorname{Im}(V_j)$ and $\tilde{V}_j,\ \tilde{V}_{j+1}$ of the consecutive iterations.

III. Matrices with special block structure

Now let

$$A = \begin{bmatrix} -K & 0 \\ 0 & M \end{bmatrix}, \; E = \begin{bmatrix} D & M \\ M & 0 \end{bmatrix}, \; B = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \tag{8}$$

with $M, D, K \in \mathbb{R}^{n \times n}$, $B_1 \in \mathbb{R}^{n \times m}$ such that $\Lambda(A,\ E) \subset \mathbb{C}_-$ which is, e.g., the case when M, D, K are positive definite. One important application, where the above matrix structure arised, is the treatment of second order dynamical systems

$$M\ddot{x}(t) + D\dot{x}(t) + Kx(t) = B_1u(t).$$

The matrices A, E, B correspond then to the equivalent first order dynamical system

$$E\dot{z}(t) = Az(t) + Bu(t), \; z(t) := [x^T(t),\ \dot{x}^T(t)]^T.$$

As shown, for instance in [2], [4], [8] the G-LR-ADI iteration can be rewritten such that it works with the original blocks M, D, K, B_1 instead of forming the larger $2n \times 2n$ matrices (8) explicitly. The key idea there is to partition the iterates V_j into an upper and lower block:

$$V_j = \begin{bmatrix} V_j^{(1)} \\ V_j^{(2)} \end{bmatrix}, \; V_j^{(k)} \in \mathbb{R}^{n \times m}, \; k \in \{1, 2\}.$$

Then, solving $(A+p_jE)\tilde{V}=EV_{j-1}$ for $\tilde{V}$ is equivalent to

$$
\begin{aligned}
(p_j^2M-p_jD+K)\tilde{V}^{(1)} &= (p_jM-D)V_{j-1}^{(1)}-MV_{j-1}^{(2)}\\
\tilde{V}^{(2)} &= V_{j-1}^{(1)}-p_j\tilde{V}^{(1)}.
\end{aligned}
$$

A derivation can be found in [16], [8]. The final algorithm is frequently being referred to as second order LR-ADI (SO-LR-ADI). It is possible to use the squaring approach of G-LR-ADI-R (Algorithm 1) and exploit the block structure of the involved matrices. This completely real SO-LR-ADI (SO-LR-ADI-R) is illustrated in Algorithm 4. It suffers from the same problems as G-LR-ADI-R since the coefficient matrix of the linear system w.r.t. a pair of complex shifts is of the form

$$-2\,\mathrm{Re}\,(p_j)K+|p_j|^2D-(K-|p_j|^2M)(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(K-|p_j|^2M)$$

and hence possibly dense, and additional linear systems with $-D+2\,\mathrm{Re}\,(p_j)M$ occur.

Algorithm 4: SO-LR-ADI-R

Input : M, D, K, B_1 as in (8) and shift parameters $\{p_1,\ldots,p_{j_{\max}}\}$.
Output: $Z=Z_{j_{\max}}\in\mathbb{R}^{n\times t_{j_{\max}}}$, such that $ZZ^T\approx P$

for $j=1,2,\ldots,j_{\max}$ **do**
  if $j=1$ **then**
    if p_1 *is real* **then**
      $\tilde{V}_1^{(1)}=(p_1^2M-p_1D+K)^{-1}(-B_1)$, $\tilde{V}_1^{(2)}=-p_1\,\mathrm{Re}\,(\tilde{V}_1^{(1)})$, $Z_1=\sqrt{-2\,\mathrm{Re}\,(p_1)}\tilde{V}_1$;
    else
      $\tilde{V}_1^{(1)}=\left(-2\,\mathrm{Re}\,(p_1)K+|p_1|^2D-(K-|p_1|^2M)(-D+2\,\mathrm{Re}\,(p_1)M)^{-1}(K-|p_1|^2M)\right)^{-1}B_1$;
      $\tilde{V}_1^{(2)}=(-D+2\,\mathrm{Re}\,(p_1)M)^{-1}(K\tilde{V}_1^{(1)}-|p_1|^2M\tilde{V}_1^{(1)})$;
      $\tilde{V}_2^{(1)}=\tilde{V}_1^{(2)}$, $\tilde{V}_2^{(2)}=M^{-1}(K\tilde{V}_1^{(1)}+D\tilde{V}_1^{(2)})$;
      $Z_2=[2\sqrt{-\,\mathrm{Re}\,(p_1)}|p_1|\tilde{V}_1,\ 2\sqrt{-\,\mathrm{Re}\,(p_1)}\tilde{V}_2]$, $j=2$;
  else
    if p_j *is real* **then**
      if p_{j-1} *is real* **then**
        $\tilde{V}_j^{(1)}=(p_j^2M-p_jD+K)^{-1}(p_jM\tilde{V}_{j-1}^{(1)}-D\tilde{V}_{j-1}^{(1)}-M\tilde{V}_{j-1}^{(2)})$;
        $\tilde{V}_j^{(2)}=\tilde{V}_{j-1}^{(1)}-p_j\tilde{V}_j^{(1)}$;
        $\tilde{V}_j=\tilde{V}_{j-1}-(p_j+p_{j-1})\tilde{V}_j$;
      else
        $\tilde{V}_j^{(1)}=(p_j^2M-p_jD+K)^{-1}(p_jM\tilde{V}_{j-2}^{(1)}-D\tilde{V}_{j-2}^{(1)}-M\tilde{V}_{j-2}^{(2)})$;
        $\tilde{V}_j^{(2)}=\tilde{V}_{j-2}^{(1)}-p_j\tilde{V}_j^{(1)}$;
        $\tilde{V}_j=\tilde{V}_{j-1}-\mathrm{Re}\,(2p_{j-1}+p_j)\tilde{V}_{j-2}+(|p_{j-1}|^2+2p_j\,\mathrm{Re}\,(p_{j-1})+p_j^2)\tilde{V}_j$;
      $Z_j=[Z_{j-1},\ \sqrt{-2\,\mathrm{Re}\,(p_j)}\tilde{V}_j]$;
    else
      if p_{j-1} *is real* **then**
        $W:=-K\tilde{V}_{j-1}^{(1)}-p_{j-1}(D\tilde{V}_{j-1}^{(1)}+M\tilde{V}_{j-1}^{(2)})+(K-|p_j|^2M)(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(M\tilde{V}_{j-1}^{(2)}-p_{j-1}M\tilde{V}_{j-1}^{(1)})$;
        $\tilde{V}_j^{(1)}=\left(-2\,\mathrm{Re}\,(p_j)K+|p_j|^2D-(K-|p_j|^2M)(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(K-|p_j|^2M)\right)^{-1}W$;
        $\tilde{V}_j^{(2)}=(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(M\tilde{V}_{j-1}^{(2)}-p_{j-1}M\tilde{V}_{j-1}^{(1)}+K\tilde{V}_j^{(1)}-|p_j|^2M\tilde{V}_j^{(1)})$;
      else
        $W:=(|p_j|^2-|p_{j-1}|^2)\begin{bmatrix}D\tilde{V}_{j-2}^{(1)}+M\tilde{V}_{j-2}^{(2)}\\ M\tilde{V}_{j-2}^{(1)}\end{bmatrix}-2\,\mathrm{Re}\,(p_j+p_{j-1})\begin{bmatrix}D\tilde{V}_{j-1}^{(1)}+M\tilde{V}_{j-1}^{(2)}\\ M\tilde{V}_{j-1}^{(1)}\end{bmatrix}$;
        $\tilde{V}_j^{(1)}=\left(-2\,\mathrm{Re}\,(p_j)K+|p_j|^2D-(K-|p_j|^2M)(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(K-|p_j|^2M)\right)^{-1}$
          $\times(W^{(1)}+(K-|p_j|^2M)(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}W^{(2)})$;
        $\tilde{V}_j^{(2)}=(-D+2\,\mathrm{Re}\,(p_j)M)^{-1}(W^{(2)}+K\tilde{V}^{(1)}-|p_j|^2M\tilde{V}_j^{(1)})$;
        $\tilde{V}_j=\tilde{V}_{j-1}+\tilde{V}_j$;
      $\tilde{V}_{j+1}^{(1)}=\tilde{V}_j^{(2)}$, $\tilde{V}_{j+1}^{(2)}=-M^{-1}(K\tilde{V}_j^{(1)}+D\tilde{V}_j^{(2)})$;
      $Z_{j+1}=[2\sqrt{-\,\mathrm{Re}\,(p_j)}|p_j|\tilde{V}_j,\ 2\sqrt{-\,\mathrm{Re}\,(p_j)}\tilde{V}_{j+1}]$, $j=j+1$;

SO-LR-ADI can also be equipped with the realification strategy of Algorithm 2 as it is shown in [4, Algorithm 1].

Algorithm 5: SO-LR-ADI-R2

Input : $M,\ D,\ K$ and B_1 as in (8) and shift parameters $\{p_1, \ldots, p_{j_{\max}}\}$.
Output: $Z = Z_{j_{\max}} \in \mathbb{R}^{n \times t_{j_{\max}}}$, such that $ZZ^T \approx P$

for $j = 1, 2, \ldots, j_{\max}$ **do**

if $j = 1$ **then**

if p_1 *is real* **then**

$\operatorname{Re}(V_1^{(1)}) = \sqrt{-2\operatorname{Re}(p_1)}(p_1^2 M - p_1 D + K)^{-1}(-B_1)$; $\operatorname{Re}(V_1^{(2)}) = -p_1 \operatorname{Re}(V_1^{(1)})$;

else

$$\begin{bmatrix} \operatorname{Re}(V_1^{(1)}) \\ \operatorname{Im}(V_1^{(1)}) \end{bmatrix} = \sqrt{-2\operatorname{Re}(p_1)} \begin{bmatrix} \gamma(p_1)M - \operatorname{Re}(p_1)D + K & -2\delta(p_1)M + \operatorname{Im}(p_1)D \\ 2\delta(p_1)M - \operatorname{Im}(p_1)D & \gamma(p_1)M - \operatorname{Re}(p_1)D + K \end{bmatrix}^{-1} \begin{bmatrix} (\operatorname{Re}(p_1)M - D)B_1 - M\operatorname{Re}(p_1)B_1^{(2)} \\ M\operatorname{Im}(p_1)B_1 \end{bmatrix};$$

$$\begin{bmatrix} \operatorname{Re}(V_1^{(2)}) \\ \operatorname{Im}(V_1^{(2)}) \end{bmatrix} = \begin{bmatrix} B_1 \\ 0 \end{bmatrix} - \begin{bmatrix} \operatorname{Re}(p_1)I_m & -\operatorname{Im}(p_1)I_m \\ \operatorname{Im}(p_1)I_m & \operatorname{Re}(p_1)I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(V_1^{(1)}) \\ \operatorname{Im}(V_1^{(1)}) \end{bmatrix};$$

else

if p_j *is real* **then**

if p_{j-1} *is real* **then**

$\operatorname{Re}(\tilde{V}^{(1)}) = \sqrt{-2\operatorname{Re}(p_j)}(p_j^2 M - p_j D + K)^{-1}((p_j M - D)\operatorname{Re}(V_{j-1}^{(1)}) - M\operatorname{Re}(V_{j-1}^{(2)}))$;

$\operatorname{Re}(\tilde{V}^{(2)}) = \operatorname{Re}(V_{j-1}^{(1)}) - p_j \operatorname{Re}(\tilde{V}^{(1)})$;

$\operatorname{Re}(V_j) = \sqrt{p_j / p_{j-1}}(\operatorname{Re}(V_{j-1}) - (p_j + p_{j-1})\tilde{V})$;

else

$[\operatorname{Re}(\tilde{V}^{(1)}), \operatorname{Im}(\tilde{V}^{(1)})] =$

$(p_j^2 M - p_j D + K)^{-1}\left[(p_j M - D)[\operatorname{Re}(V_{j-1}^{(1)}), \operatorname{Im}(V_{j-1}^{(1)})] - M[\operatorname{Re}(V_{j-1}^{(2)}), \operatorname{Im}(V_{j-1}^{(2)})]\right]$;

$[\operatorname{Re}(\tilde{V}^{(2)}), \operatorname{Im}(\tilde{V}^{(2)})] = [\operatorname{Re}(V_{j-1}^{(1)}), \operatorname{Im}(V_{j-1}^{(1)})] - p_1[\operatorname{Re}(V_{j-1}^{(2)}), \operatorname{Im}(V_{j-1}^{(2)})]$;

$\operatorname{Re}(V_j) = \sqrt{\frac{p_j}{\operatorname{Re}(p_{j-2})}}\left(\operatorname{Re}(V_{j-1}) - (p_j + \operatorname{Re}(p_{j-2}))\operatorname{Re}(\tilde{V}) + \operatorname{Im}(p_{j-2})\operatorname{Im}(\tilde{V})\right)$;

else

if p_{j-1} *is real* **then**

$$\begin{bmatrix} \operatorname{Re}(\tilde{V}^{(1)}) \\ \operatorname{Im}(\tilde{V}^{(1)}) \end{bmatrix} = \begin{bmatrix} \gamma(p_j)M - \operatorname{Re}(p_j)D + K & -2\delta(p_j)M + \operatorname{Im}(p_j)D \\ 2\delta(p_j)M - \operatorname{Im}(p_j)D & \gamma(p_j)M - \operatorname{Re}(p_j)D + K \end{bmatrix}^{-1} \begin{bmatrix} (\operatorname{Re}(p_j)M - D)\operatorname{Re}(V_{j-1}^{(1)}) - M\operatorname{Re}(V_{j-1}^{(2)}) \\ \operatorname{Im}(p_j)M\operatorname{Re}(V_{j-1}^{(1)}) \end{bmatrix};$$

$$\begin{bmatrix} \operatorname{Re}(\tilde{V}^{(2)}) \\ \operatorname{Im}(\tilde{V}^{(2)}) \end{bmatrix} = \begin{bmatrix} \operatorname{Re}(V_{j-1}^{(1)}) \\ 0 \end{bmatrix} - \begin{bmatrix} \operatorname{Re}(p_j)I_m & -\operatorname{Im}(p_j)I_m \\ \operatorname{Im}(p_j)I_m & \operatorname{Re}(p_j)I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}^{(1)}) \\ \operatorname{Im}(\tilde{V}^{(1)}) \end{bmatrix};$$

$$\begin{bmatrix} \operatorname{Re}(V_j) \\ \operatorname{Im}(V_j) \end{bmatrix} = \sqrt{\frac{\operatorname{Re}(p_j)}{p_{j-1}}} \left(\begin{bmatrix} \operatorname{Re}(V_{j-1}) \\ 0 \end{bmatrix} - \begin{bmatrix} (\operatorname{Re}(p_j) + p_{j-1})I_m & -\operatorname{Im}(p_j)I_m \\ \operatorname{Im}(p_j)I_m & (\operatorname{Re}(p_j) + p_{j-1})I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} \right);$$

else

$$\begin{bmatrix} \operatorname{Re}(\tilde{V}^{(1)}) \\ \operatorname{Im}(\tilde{V}^{(1)}) \end{bmatrix} =$$

$$\begin{bmatrix} \gamma(p_j)M - \operatorname{Re}(p_j)D + K & -2\delta(p_j)M + \operatorname{Im}(p_j)D \\ 2\delta(p_j)M - \operatorname{Im}(p_j)D & \gamma(p_j)M - \operatorname{Re}(p_j)D + K \end{bmatrix}^{-1} \begin{bmatrix} (\operatorname{Re}(p_j)M - D)\operatorname{Re}(V_{j-1}^{(1)}) - \operatorname{Im}(p_j)M\operatorname{Im}(V_{j-1}^{(1)}) - M\operatorname{Re}(V_{j-1}^{(2)}) \\ (\operatorname{Re}(p_j)M - D)\operatorname{Im}(V_{j-1}^{(1)}) + \operatorname{Im}(p_j)M\operatorname{Re}(V_{j-1}^{(1)}) - M\operatorname{Im}(V_{j-1}^{(2)}) \end{bmatrix};$$

$$\begin{bmatrix} \operatorname{Re}(\tilde{V}^{(2)}) \\ \operatorname{Im}(\tilde{V}^{(2)}) \end{bmatrix} = \begin{bmatrix} \operatorname{Re}(V_{j-1}^{(1)}) \\ \operatorname{Im}(V_{j-1}^{(1)}) \end{bmatrix} - \begin{bmatrix} \operatorname{Re}(p_j)I_m & -\operatorname{Im}(p_j)I_m \\ \operatorname{Im}(p_j)I_m & \operatorname{Re}(p_j)I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}^{(1)}) \\ \operatorname{Im}(\tilde{V}^{(1)}) \end{bmatrix};$$

$$\begin{bmatrix} \operatorname{Re}(V_j) \\ \operatorname{Im}(V_j) \end{bmatrix} = \sqrt{\frac{\operatorname{Re}(p_j)}{\operatorname{Re}(p_{j-2})}} \left(\begin{bmatrix} \operatorname{Re}(V_{j-1}) \\ \operatorname{Im}(V_{j-1}) \end{bmatrix} - \begin{bmatrix} (\operatorname{Re}(p_j) + \operatorname{Re}(p_{j-2}))I_m & -(\operatorname{Im}(p_j) + \operatorname{Im}(p_{j-2}))I_m \\ (\operatorname{Im}(p_j) + \operatorname{Im}(p_{j-2}))I_m & (\operatorname{Re}(p_j) + \operatorname{Re}(p_{j-2}))I_m \end{bmatrix} \begin{bmatrix} \operatorname{Re}(\tilde{V}) \\ \operatorname{Im}(\tilde{V}) \end{bmatrix} \right);$$

if p_j *is real* **then**

$Z_j = [Z_{j-1},\ \operatorname{Re}(V_j)]$;

else

$\beta = 2\frac{\operatorname{Re}(p_j)}{\operatorname{Im}(p_j)}$;

$[\operatorname{Re}(V_{j+1}),\ \operatorname{Im}(V_{j+1})] = [\operatorname{Re}(V_j) + \beta\operatorname{Im}(V_j),\ -\operatorname{Im}(V_j)]$;

$Z_{j+1} = \left[Z_{j-1},\ \sqrt{2}\operatorname{Re}(V_j) + \frac{\beta}{\sqrt{2}}\operatorname{Im}(V_j),\ \sqrt{\frac{\beta^2}{2} + 2} \cdot \operatorname{Im}(V_j)\right]$;

Set $j = j + 1$;

An analogue to our novel completely real G-LR-ADI (Algorithm 3) is also possible and given below, where we used the quantities

$$\gamma(z) := \mathrm{Re}\,(z)^2 - \mathrm{Im}\,(z)^2,\ \ \delta(z) := \mathrm{Re}\,(z)\,\mathrm{Im}\,(z),\ \ z \in \mathbb{C}.$$

to rewrite the equivalent real linear systems.

As in G-LR-ADI-R2, the occurring linear systems of dimension $2n \times 2n$ in Algorithm 5 are much more appealing as $n \times n$ systems in Algorithm 4.

Remark 4. As in remark 1, if an iterative solver is used for the linear systems the augmented linear system in SO-LR-ADI-R2 should be chosen appropriately. For instance, if M, D, K are symmetric, then the coefficient matrices of the equivalent real linear systems

$$\begin{bmatrix} \gamma(p_j)M - \mathrm{Re}\,(p_j)D + K & 2\delta(p_j)M - \mathrm{Im}\,(p_j)D \\ 2\delta(p_j)M - \mathrm{Im}\,(p_j)D & -\gamma(p_j)M + \mathrm{Re}\,(p_j)D - K \end{bmatrix} \begin{bmatrix} \mathrm{Re}\,(\tilde{V}^{(1)}) \\ -\,\mathrm{Im}\,(\tilde{V}^{(1)}) \end{bmatrix} = \begin{bmatrix} (\mathrm{Re}\,(p_j)M - D)\,\mathrm{Re}\,(V_{j-1}^{(1)}) - \mathrm{Im}\,(p_j)M\,\mathrm{Im}\,(V_{j-1}^{(1)}) - M\,\mathrm{Re}\,(V_{j-1}^{(2)}) \\ (\mathrm{Re}\,(p_j)M - D)\,\mathrm{Im}\,(V_{j-1}^{(1)}) + \mathrm{Im}\,(p_j)M\,\mathrm{Re}\,(V_{j-1}^{(1)}) - M\,\mathrm{Im}\,(V_{j-1}^{(2)}) \end{bmatrix},$$

$$\begin{bmatrix} 2\delta(p_j)M - \mathrm{Im}\,(p_j)D & \gamma(p_j)M - \mathrm{Re}\,(p_j)D + K \\ \gamma(p_j)M - \mathrm{Re}\,(p_j)D + K & -2\delta(p_j)M + \mathrm{Im}\,(p_j)D \end{bmatrix} \begin{bmatrix} \mathrm{Re}\,(\tilde{V}^{(1)}) \\ \mathrm{Im}\,(\tilde{V}^{(1)}) \end{bmatrix} = \begin{bmatrix} (\mathrm{Re}\,(p_j)M - D)\,\mathrm{Im}\,(V_{j-1}^{(1)}) + \mathrm{Im}\,(p_j)M\,\mathrm{Re}\,(V_{j-1}^{(1)}) - M\,\mathrm{Im}\,(V_{j-1}^{(2)}) \\ (\mathrm{Re}\,(p_j)M - D)\,\mathrm{Re}\,(V_{j-1}^{(1)}) - \mathrm{Im}\,(p_j)M\,\mathrm{Im}\,(V_{j-1}^{(1)}) - M\,\mathrm{Re}\,(V_{j-1}^{(2)}) \end{bmatrix}$$

are symmetric, too. Note also that the original complex coefficient matrix $p_j^2 M - p_j D + K$ is complex symmetric in this case.

IV. Numerical Experiments

We briefly compare the discussed real versions of LR-ADI with MATLAB®. All linear solves were carried out with the backslash-command and the ADI iteration was terminated when

$$\|AZZ^TE^T + EZZ^TA^T + BB^T\|_2/\|BB^T\|_2 \ \leq\ \varepsilon,\ \varepsilon \ll 1,$$

where the spectral norm of the Lyapunov residual was computed efficiently using a Lanczos process. We use the following test examples without any concern whether the considered Lyapunov equation has a realistic meaning. For each example, the right hand side factor B is a random vector of appropriate length. Moreover, $J = J_{\mathbb{R}} + 2J_{\mathbb{C}}$ heuristic shifts, consisting of $J_{\mathbb{R}}$ real and $J_{\mathbb{C}}$ pairs of complex conjugate numbers, were used and computed following the approach in [14]. There, k_+ and k_- Ritz values with respect to $E^{-1}A$ and $A^{-1}E$, respectively, are constructed with an Arnoldi process in order to solve the minmax problem in an approximate sense.

Example 1. We take the matrix A with $n = 42249$ which represent a finite element model of an ocean circulation problem [18].

Example 2. For a generalized Lyapunov equation and G-LR-ADI, the coefficient matrix A is obtained from a finite difference discretization of a three-dimensional convection diffusion reaction equation on the domain $[0,1]^3$ similar to [7, Example 2], and E is chosen as the discrete Laplacian. Using 19 equidistant grid points in each spatial direction leads to $n = 6859$.

Example 3. For testing the variants of SO-LR-ADI we take matrices of the form (8) with $M, D, K \in \mathbb{R}^{n\times n}$ s.p.d, where is constructed as $D = 0.2 \cdot M + 0.05 \cdot K$ and $\nu = 5$ is added to the first, n-th, and $2n+1$-th diagonal entry of D. These matrices define a second order dynamical system which describes the equations of motion of a triple mass-spring-damper oscillator chain [17]. With 4000 masses in each chain this leads to $n = 12001$.

TABLE I
Parameters, required iterations j_{ITER}, and computation times in seconds.

Example	Parameters			j_{iter}	CPU time (s)			
	$k_+,\ k_-$	$J,\ (J_{\mathbb{R}},\ J_{\mathbb{C}})$	ε		basic	-R	-pR	-R2
1, LR-ADI	40, 30	30, (18, 6)	10^{-10}	86	239.130	248.488	65.392	88.943
2, G-LR-ADI	50, 30	30, (0, 30)	10^{-10}	30	13.281	389.444	7.224	13.284
3, SO-LR-ADI	80, 80	50, (14, 17)	10^{-8}	171	41.238	2416.603	17.188	20.895

The required number of iteration and computing times are given in Table I. The older completely real version obviously have the largest computational effort leading to much higher CPU times, especially for both generalized examples. If the matrix A defining a standard Lyapunov equation has very easy structure, e.g. if it is multi-diagonal with a low-bandwidth, then it can happen that the LR-ADI-R outperforms the other approaches since the real linear system involving A^2 might be easier to solve than the complex or equivalent real ones of the other approaches. The extremely high CPU time for Example 3 of SO-LR-ADI clearly shows the inferiority of the traditional approach even for this artificial test example, where the matrices $M,\ D,\ K$ are at most tridiagonal. In other similar experiments using examples from real industrial applications, G-LR-ADI and SO-LR-ADI did not finish in a reasonable amount of time. In some cases they even could not be carried out because of memory limitations. Thus, if no complex arithmetic is available, the completely real approach [5] based on an equivalent real formulation of the involved linear system is clearly superior. However, it is also apparent that the partially real version [6] is the fastest one for all examples and hence we suggest to choose those if complex linear systems can be dealt with efficiently.

V. Conclusions

We reviewed real formulations of the LR-ADI method which enable the computation of real low-rank solution factors of large-scale generalized Lyapunov equations in the presence of complex shift parameters. They employ only real or only an absolutely necessary amount of complex arithmetic operations and storage. Detailed implementations of the different approaches for important special cases of generalized Lyapunov equation are given. Moreover, relations between an older and a novel completely real approach were established. Numerical examples show the superiority of the novel approach in the generalized case if no complex arithmetic is possible. If complex linear system can be solved efficiently, a partially real approach based on [6] should be preferred.

References

[1] P. Benner, *Solving large-scale control problems*, IEEE Control Systems Magazine, 14 (2004), pp. 44–59. 1

[2] P. Benner, P. Eberhard, P. Kürschner, and C. Nowakowski, *Model reduction of an elastic crankshaft for elastic multibody simulations*, Tech. Rep. MPIMD/12-07, Max Planck Institute Magdeburg Preprints, 2012. 5

[3] P. Benner, M. Köhler, and J. Saak, *Sparse-Dense Sylvester equations in $\mathcal{H}_2$-model order reduction*, Tech. Rep. MPIMD/11-11, Max Planck Institute Magdeburg Preprints, December 2011. Submitted to Journal of Computational and Applied Mathematics. 5

[4] P. Benner, P. Kürschner, and J. Saak, *Improved second-order balanced truncation for symmetric systems*, Tech. Rep. MPIMD/11-13, Max Planck Institute Magdeburg Preprints, December 2011. 5, 6

[5] ———, *Avoiding complex arithmetic in the low-rank adi method efficiently*, PAMM, (2012). submitted. 1, 3, 9

[6] ———, *Efficient handling of complex shift parameters in the low-rank Cholesky factor ADI method*, Numerical Algorithms, (2012), pp. 1–27. 10.1007/s11075-012-9569-7, also available as preprint http://www.mpi-magdeburg.mpg.de/preprints/2011/08/. 1, 2, 9

[7] P. Benner, J.-R. Li, and T. Penzl, *Numerical solution of large Lyapunov equations, Riccati equations, and linear-quadratic control problems*, Numer. Lin. Alg. Appl., 15 (2008), pp. 755–777. 1, 8

[8] P. Benner and J. Saak, *Efficient Balancing based MOR for Large Scale Second Order Systems*, Math. Comput. Model. Dyn. Sys., 17 (2011), pp. 123–143. DOI:10.1080/13873954.2010.540822. 5, 6

[9] T. A. Davis, *Direct Methods for Sparse Linear Systems (Fundamentals of Algorithms 2)*, SIAM, Philadelphia, PA, USA, 2006. 1

[10] D. Day and M. A. Heroux, *Solving complex-valued linear systems via equivalent real formulations*, SIAM J. Sci. Comput., 23 (2001), pp. 480–498. 2, 3

[11] I. Duff, A. Erisman, and J. Reid, *Direct methods for sparse matrices*, Clarendon Press, Oxford, UK, 1989. 1

[12] R. W. Freund, *Conjugate gradient-type methods for linear systems with complex symmetric coefficient matrices*, SIAM Sci. Stat. Comp., 13 (1992), pp. 425–448. 3

[13] J.-R. Li and J. White, *Low rank solution of Lyapunov equations*, SIAM J. Matrix Anal. Appl., 24 (2002), pp. 260–280. 1

[14] T. Penzl, *A cyclic low rank Smith method for large sparse Lyapunov equations*, SIAM J. Sci. Comput., 21 (2000), pp. 1401–1418. 1, 8

[15] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, PA, USA, 2003. 1

[16] J. Saak, *Efficient Numerical Solution of Large Scale Algebraic Matrix Equations in PDE Control and Model Order Reduction*, PhD thesis, TU Chemnitz, July 2009. Available from http://nbn-resolving.de/urn:nbn:de:bsz:ch1-200901642. 6

[17] N. Truhar and K. Veselić, *Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix*, Syst. Control Lett., 56 (2007), pp. 493–503. 8

[18] M. van Gijzen, C. Vreugdenhil, and H. Oksuzoglu, *The finite element discretization for stream-function problems on multiply connected domains*, Journal of Computational Physics, 140 (1998), pp. 30 – 46. 8