Applied Numerical Mathematics 109 (2016) 85–95

Contents lists available at ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

ADI iteration for Lyapunov equations: A tangential approach and adaptive shift selection

Thomas Wolf*, Heiko K.F. Panzer, Boris Lohmann

Institute of Automatic Control, Technische Universität München, Boltzmannstr. 15, D-85748 Garching, Germany

ARTICLE INFO

Article history:
Received 24 March 2014
Received in revised form 12 June 2016
Accepted 27 June 2016
Available online 1 July 2016

Keywords:
Lyapunov equation
Large-scale systems
ADI iteration

ABSTRACT

A new version of the alternating directions implicit (ADI) iteration for the solution of large-scale Lyapunov equations is introduced. It generalizes the hitherto existing iteration, by incorporating tangential directions in the way they are already available for rational Krylov subspaces. Additionally, first strategies to adaptively select shifts and tangential directions in each iteration are presented. Numerical examples emphasize the potential of the new results.

1. Introduction

We consider the generalized algebraic Lyapunov equation

$$\boldsymbol{A}\boldsymbol{P}\boldsymbol{E}^T + \boldsymbol{E}\boldsymbol{P}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T = \boldsymbol{0}, \tag{1}$$

with $\boldsymbol{A}, \boldsymbol{E}, \boldsymbol{P} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{n\times m}$. The order n is assumed to be large and m small, $m \ll n$. Furthermore, the matrix $\boldsymbol{E}$ has full rank, $\det(\boldsymbol{E}) \neq 0$. The standard formulation of (1) results from setting $\boldsymbol{E}$ to identity. The solution $\boldsymbol{P}$ of the Lyapunov equation (1) represents the so-called controllability Gramian of a linear time-invariant dynamical system. It is used e.g. in model reduction by *truncated balanced realization* (TBR), see [1] for details.

Direct (also called dense) numerical solvers of (1) are available in matrix computation software [15]. However, for high-dimensional problems with large n, they become inappropriate due to high memory requirements and execution time. A remedy is to employ iterative methods that can exploit the sparsity of the matrices $\boldsymbol{A}$, $\boldsymbol{E}$ and $\boldsymbol{B}$. These methods typically compute a low-rank factor $\boldsymbol{Z}$, having less columns than rows. The approximation $\widehat{\boldsymbol{P}} \approx \boldsymbol{P}$ is then given by the low-rank formulation $\widehat{\boldsymbol{P}} := \boldsymbol{Z}\boldsymbol{Z}^H$. This reduces memory requirements, because only the low-rank factor $\boldsymbol{Z}$ has to be stored, instead of the full n-by-n matrix $\widehat{\boldsymbol{P}}$.

Here we focus on the *alternating directions implicit* (ADI) iteration in its low-rank formulation [12,13]. It successively accumulates the low-rank factor $\boldsymbol{Z}$ by a block of m columns. This, however, can be a drawback if $m > 1$ and a high number of iterations is necessary for sufficient approximation: a low-rank factor $\boldsymbol{Z}$, growing by m columns per iteration, might become too large for reasonable processing.

In this work we propose a *tangential* generalization of the ADI iteration, for which the low-rank factor $\boldsymbol{Z}$ grows by only one column per iteration—irrespective of m. The hitherto existing ADI iteration based on blocks is included as a special

* Corresponding author.
E-mail addresses: thomas.wolf@tum.de (T. Wolf), panzer@tum.de (H.K.F. Panzer), lohmann@tum.de (B. Lohmann).

http://dx.doi.org/10.1016/j.apnum.2016.06.006

case. The tangential ADI iteration is most appropriately performed in combination with an adaptive shift selection. We also address this by suggesting a first procedure that determines new shifts and tangential directions on the fly.

In Section 2 relevant preliminaries are reviewed. A reformulation of the ADI iteration is presented in Section 3, from which the tangential version is derived in Section 4. The adaptive shift selection is addressed in Section 5 and numerical examples are given in Section 6.

Throughout the paper, we use the following notation: $\boldsymbol{I}_k$ denotes the $k \times k$ identity matrix. Complex conjugation of a matrix $\boldsymbol{X}$ is denoted by $\overline{\boldsymbol{X}}$; $\boldsymbol{X}^T$ means transposition and $\boldsymbol{X}^H = \overline{\boldsymbol{X}}^T$ is transposition with complex conjugation. The real and imaginary part of a complex $\boldsymbol{X}$ are given by $\boldsymbol{X} = \mathrm{Re}(\boldsymbol{X}) + \imath\,\mathrm{Im}(\boldsymbol{X})$, where $\imath$ is the imaginary unit. The open right half of the complex plane is denoted by $\mathbb{C}_+$ and $\Lambda(\boldsymbol{A})$ denotes the spectrum of a quadratic matrix $\boldsymbol{A}$.

2. Preliminaries

We review the low-rank ADI iteration for solving (1) in the following.

2.1. The ADI iteration

For a given set $\mathcal{S} = \{s_1, s_2, \dots, s_k\}$ of complex valued shifts $s_i \in \mathbb{C}_+$, the blocks $\boldsymbol{Z}_i \in \mathbb{C}^{n\times m}$ of the low-rank factor $\boldsymbol{Z} = [\boldsymbol{Z}_1,\ \boldsymbol{Z}_2,\ \dots,\ \boldsymbol{Z}_k]$ are given by the following ADI iteration for $i = 2, \dots, k$, [12]:

$$\begin{aligned} \boldsymbol{Z}_1 &= \sqrt{2\,\mathrm{Re}(s_1)}\,(\boldsymbol{A} - s_1\boldsymbol{E})^{-1}\,\boldsymbol{B}, \\ \boldsymbol{Z}_i &= \sqrt{\frac{\mathrm{Re}(s_i)}{\mathrm{Re}(s_{i-1})}}\left[\boldsymbol{I} + (s_i + \overline{s}_{i-1})\,(\boldsymbol{A} - s_i\boldsymbol{E})^{-1}\,\boldsymbol{E}\right]\boldsymbol{Z}_{i-1}. \end{aligned} \tag{2}$$

As $\boldsymbol{A}$, $\boldsymbol{E}$ and $\boldsymbol{B}$ are real, we assume the set $\mathcal{S}$ to be closed under conjugation, causing also $\widehat{\boldsymbol{P}} = \boldsymbol{Z}\boldsymbol{Z}^H$ to be real. Typically, the set $\mathcal{S}$ is chosen a priori and reused periodically until convergence occurs [12,14]. The residual $\boldsymbol{R} \in \mathbb{R}^{n\times n}$ is the remainder in the original Lyapunov equation (1), after substituting $\boldsymbol{P}$ by $\widehat{\boldsymbol{P}}$:

$$\boldsymbol{R} := \boldsymbol{A}\widehat{\boldsymbol{P}}\boldsymbol{E}^T + \boldsymbol{E}\widehat{\boldsymbol{P}}\boldsymbol{A}^T + \boldsymbol{B}\boldsymbol{B}^T. \tag{3}$$

In [4] and [16,17] it was independently shown, that the residual can be factorized as $\boldsymbol{R} = \boldsymbol{B}_\perp\boldsymbol{B}_\perp^T$—with real $\boldsymbol{B}_\perp \in \mathbb{R}^{n\times m}$, if the set $\mathcal{S}$ is closed under conjugation. One way to compute $\boldsymbol{B}_\perp$ is

$$\boldsymbol{B}_\perp = \boldsymbol{B} + \boldsymbol{E}\boldsymbol{Z}\boldsymbol{L}^T, \quad \text{with} \tag{4}$$

$$\boldsymbol{L} := [\,\sqrt{2\,\mathrm{Re}(s_1)}\boldsymbol{I}_m,\ \dots,\ \sqrt{2\,\mathrm{Re}(s_k)}\boldsymbol{I}_m\,]. \tag{5}$$

The notation $\boldsymbol{B}_\perp$ stems from the fact, that a specific projection can be associated to the ADI iteration: $\boldsymbol{B}_\perp$ is the residual after projecting $\boldsymbol{B}$, and it fulfills the Petrov–Galerkin condition, see [17] for details. The low-rank formulation $\boldsymbol{R} = \boldsymbol{B}_\perp\boldsymbol{B}_\perp^T$ allows to compute the induced matrix 2-norm of the residual $\|\boldsymbol{R}\|_2$ by the maximum eigenvalue of the m-by-m matrix $\boldsymbol{B}_\perp^T\boldsymbol{B}_\perp$: $\|\boldsymbol{R}\|_2 = \max\Lambda(\boldsymbol{B}_\perp^T\boldsymbol{B}_\perp)$. This can be done with low numerical effort, which is why one often uses $\|\boldsymbol{R}\|_2$ as a convergence criterion in the ADI iteration.

2.2. The problem

The low-rank factor $\boldsymbol{Z}$ of the ADI iteration (2) gains a new block of m columns in each step. For problems that require many iterations for convergence, i.e. until $\|\boldsymbol{R}\|_2$ is small enough, the final $\boldsymbol{Z}$ might become too large for reasonable processing. It would be desirable, that for every shift s_i only one column, or as many columns as absolutely necessary, are added to the low-rank factor $\boldsymbol{Z}$. An ad hoc solution would be to substitute $\boldsymbol{B}$ in (2) by $\boldsymbol{B}\boldsymbol{b}$, where $\boldsymbol{b} \in \mathbb{C}^m$ is referred to as a *tangential direction* in the following. However, as every block/column $\boldsymbol{Z}_i$ in the iteration (2) originates from its predecessor $\boldsymbol{Z}_{i-1}$, this tangential direction is fixed for all time, and the left over directions in $\boldsymbol{B}$ would be completely neglected. Obviously, this generally cannot result in a good approximation.

Our first contribution will be a re-formulation of the ADI iteration (2). This re-formulation is more natural, as it incorporates the low-rank factor $\boldsymbol{B}_\perp$ of the residual, and it is essential for our second contribution: we show, how each shift s_i can be equipped with an individual tangential direction $\boldsymbol{b}_i$. We will denote this variation with *tangential (low-rank) ADI iteration* (T-LR-ADI) in the following. As it is advisable to employ T-LR-ADI with adaptively selected shifts s_i and tangential directions $\boldsymbol{b}_i$, a first procedure of this kind is our third contribution. Because the results require the recently established connection of the ADI iteration to Krylov subspace projection methods [16,17], we briefly review this in the following subsections.