

# An alternative way of solving large Lyapunov equations

André K. Eppler<sup>1,\*</sup> and Matthias Bollhöfer<sup>1,\*\*</sup>

<sup>1</sup> TU Braunschweig, Institut Computational Mathematics, Pockelsstrasse 14, 38116 Braunschweig

System reduction has become an essential tool in simulating large electrical circuits. Among such methods balanced truncation is in particular popular since it preserves properties like passivity of the underlying descriptor system. One of the main tasks therein consists of solving large scale Lyapunov equations efficiently. In this setting one is looking for a symmetric low rank solution. For this purpose we discuss Krylov-subspace methods such as the **General-Minimal-Residual** method. To speed up convergence the method is combined with the **Low Rank Cholesky-Factor-Alternating-Direct-Implicit-Iteration** as a structure preserving preconditioner.

Copyright line will be provided by the publisher

## 1 Introduction

The numerical simulation of large scale integrated circuits in VLSI design nowadays reaches systems with several hundred million elements. The underlying models require verification by many simulation runs based on different input signals. To do this efficiently one usually replaces the original system by a structure-preserving model of much smaller order which inherits the central properties of the underlying circuit like stability and passivity. To this end model order reduction (MOR) methods have become a key technology in constructing reduced order models, in particular the balanced truncation method [1] will be of interest in this paper. In the simplest case the original circuit can be described by an descriptor system of type

$$\begin{aligned} E\dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \quad \text{where } A, E \in \mathbb{R}^{n,n}, B \in \mathbb{R}^{n,m}, C \in \mathbb{R}^{p,n}, D \in \mathbb{R}^{p,m}$$

such that  $m, p \ll n$ . The reduced model replaces  $E, A, B, C, D$  by smaller matrices  $\tilde{E}, \tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$  such that for all matrices the dimension  $n$  is replaced by some  $l \ll n$ . To obtain a reduced model, balanced truncation computes a symmetric positive semidefinite solution of associated projected Lur'e equations (for special cases these are equivalent to projected Riccati equations). These in turn require solving a sequence of generalized Lyapunov equations of the form

$$\begin{aligned} AXE^T + EXA^T + P_l BB^T P_l^T &= 0, \text{ where } X = P_r X P_r^T, \\ A^T Y E + E^T Y A + P_r^T C^T C P_r &= 0, \text{ where } Y = P_l^T Y P_l. \end{aligned} \quad (1)$$

Here  $P_l, P_r$  refer to the left and right projection of the matrix pencil  $\lambda E - A$  to the subspace of finite eigenvalues. Assuming that the system is passive, we obtain symmetric and positive semidefinite solutions  $X, Y$  of (1). Moreover, in the case of MOR we are seeking for low rank solutions  $X = RR^T, Y = LL^T$ . Since solving (1) is a central task in balanced truncation, we will discuss structure-preserving numerical methods for their solution.

## 2 Low rank cholesky factor ADI

One of the most popular methods for solving (generalized) Lyapunov equations consists of the alternating direction implicit (ADI) method [2]. For simplicity we only consider the first equation in (1). The ADI method solves this system iteratively by computing for  $j = 1, 2, 3, \dots$

$$(E + \tau_j A)X_{j-\frac{1}{2}} = -P_l BB^T P_l^T - X_{j-1}(E - \tau_j A)^T, \quad X_j(E + \tau_j A)^T = -P_l BB^T P_l^T - (E - \tau_j A)X_{j-\frac{1}{2}}.$$

The low rank Cholesky factor ADI method directly computes  $X_j = R_j R_j^T$ , where

$$R_j = \left[ \sqrt{-2\text{Re}(\tau_j)} \{ (E + \tau_j A)^{-1} P_l B \}, \{ (E + \tau_j A)^{-1} (E - \bar{\tau}_j A) R_{j-1} \} \right]$$

It has been observed in [3] that interchanging the order of the shifts  $\tau_1, \tau_2, \dots$  allows reducing the complexity of computing  $R_j$  by one order of magnitude. This has lead to the low-rank Cholesky factor ADI method. For the convergence of the ADI method the choice of the shift parameters  $\tau_1, \tau_2, \dots$  is essential. For the simplest case optimal shift parameters are known [4], in general one has to work with heuristic parameters [5].

As for any other iterative method, the mapping  $X_{j-1} \rightarrow X_j$  induces an approximate inverse to the linear operator given by the generalized Lyapunov equation. This can be used for convergence acceleration within Krylov subspace methods.

\* André K. Eppler E-mail: a.eppler@tu-braunschweig.de, Phone: +49 531 391 7542, Fax: +49 531 391 8206

\*\* Matthias Bollhöfer E-mail: m.bollhoefer@tu-braunschweig.de, Phone: +49 531 391 7536, Fax: +49 531 391 8206

### 3 Krylov subspace methods and Galerkin projection

For the solution of generalized Lyapunov we discuss the use of Krylov subspace methods such as GMRES [6]. Krylov subspace methods in general consist of matrix vector multiplications as well as linear combinations. It is easy to see that a linear combination of symmetric low rank matrices leads again to a symmetric matrix of lower rank. Furthermore the “matrix-vector” multiplication by the Lyapunov operator applied to  $VZV^T$

$$AVZV^T E^T + EVZV^T A^T = [AVEV] \begin{bmatrix} 0 & Z \\ Z & 0 \end{bmatrix} [AVEV]^T,$$

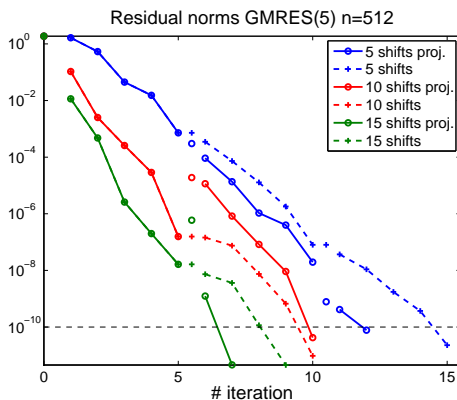
which can be rewritten as a symmetric matrix of lower rank as well. As for the linear combination of matrices, the “matrix-vector” product can be truncated by method like the rank-revealing QR decomposition [7] to reduce the rank of  $[AVEV]$ . One problem for Krylov subspace methods in general is the use of preconditioning methods. E.g., diagonal preconditioning can be implemented fairly easy but in the case of Lyapunov equations it would immediately destroy the low rank structure. To preserve the structure of the iterates in Krylov subspace methods we therefore propose to use the LRCF-ADI method for preconditioning. Suppose that the GMRES method computes Arnoldi vectors  $\mathcal{V}_j = V_j Z_j V_j^T$  represented as sym. low rank matrices, then LR-CFADI preconditioning is applied to  $V_j$  and returns  $L_j$  for  $\mathcal{W}_j = L_j D_j L_j^T$  for  $j = 1 \dots m$ .

Another concept for solving Lyapunov equations by Galerkin projection was proposed in [8]. Here we can use this idea for building a solution different from GMRES. For this we calculate a projection basis from the two sequences of Arnoldi vectors. We compute a rank-revealing QR decomposition of each sequence  $[L_1 \dots L_m] = Q_L R_L \Pi_L^T$ ,  $\text{rank} R_L = r_L$ ,  $[V_1 \dots V_m] = Q_V R_V \Pi_V^T$ ,  $\text{rank} R_V = r_V$ . Suppose  $r = \max\{r_L, r_V\}$ , then the GMRES solution  $X_m$  can be represented

as  $X_m = X_0 + Q_L S Q_L^T$ , where  $S = R_L \Pi_L \begin{pmatrix} s_1 D_1 & & \\ & \ddots & \\ & & s_m D_m \end{pmatrix} \Pi^T R_L^T$  and  $s_1 \dots s_m$  are chosen to minimize the asso-

ciated Hessenberg problem. Now we project the Lyapunov equation with  $Q_V$  and get a  $r$ -dimensional Lyapunov equation for  $S$ :  $Q_V^T \mathcal{R}_0 Q_V + A_Q S E_Q^T + E_Q S A_Q = 0$  where  $E_Q = Q_V^T E Q_L$ ,  $A_Q = Q_V^T A Q_L$  and  $\mathcal{R}_0$  initial residual. This can be solved by standard techniques [9] and the  $X_m$  can be computed from that.

### 4 Numerical experiment



In the left figure one can see the convergence history of the residual norms in the GMRES algorithm for different numbers of shift parameters. After 5 steps of GMRES a restart is taken. The solid lines represent when Galerkin projection was done while for the dashed lines the GMRES solution was formed. Applying the described Galerkin projection does not yield a minimizing property as for GMRES but it was observed that the projection often reduced the number of iterations needed in GMRES. On the other hand solving an additional Lyapunov equation is time consuming.

**Acknowledgements** This work is supported by the German Federal Ministry of Education and Research (BMBF) grant no. 03BOPAE4.

### References

- [1] V. Mehrmann and T. Stykel, in: Balanced Truncation Model Reduction for Large-Scale Systems in Descriptor Form, edited by P. Benner, V. Mehrmann, and D. Sorensen, Lecture Notes in Computational Science and Engineering Vol. 45 (, Berlin, Heidelberg, 2005), pp. 83–115.
- [2] T. Stykel, Electron. Trans. Numer. Anal. **30**, 187–202 (2008).
- [3] J. R. Li and J. White, SIAM J. Matrix Anal. Appl. **24**(1), 260–280 (2002).
- [4] E. L. Wachspress, Appl. Math. Lett. **1**(1), 87–90 (1988).
- [5] T. Penzl, SIAM J. Sci. Comput. **21**(4), 1401–1418 (2000).
- [6] Y. Saad and M. H. Schultz, SIAM J. Sci. Stat. Comput. **7**(3) (1986).
- [7] G. H. Golub and C. F. Van Loan, Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition), 3rd edition (The Johns Hopkins University Press, October 1996).
- [8] I. M. Jaimoukha and E. M. Kasenally, SIAM J. Numer. Anal. **31**(1), 227–251 (1994).
- [9] R. H. Bartels and G. W. Stewart, Commun. ACM **15**(9), 820–826 (1972).