PSLICOT Routines for Model Reduction of Stable Large-Scale Systems *

Peter Benner[†] Enrique S. Quintana-Ortí[‡] Gregorio Quintana-Ortí[§]

December 11, 2000

Abstract

We describe model reduction routines developed for computer architectures with distributed memory. These routines form the basis for the model reduction software to be incorporated in PSLICOT, the extension of SLICOT for parallel computing. The routines considered here match the functionality of a subset of the model reduction routines in SLICOT. The algorithms implemented differ from those in SLICOT as the system Gramians are computed using solvers based on the sign function or squared Smith methods. These computations only require efficient implementations of basic linear algebra operations readily available, e.g., in BLAS, LAPACK, and ScaLAPACK. Hence, good performance of the resulting algorithms on parallel computers is to be expected. Our experimental results on a PC cluster show good performance of the parallel implementations.

1 Introduction

We consider the transfer function matrix (TFM) $G(\lambda) = C(\lambda I - A)^{-1}B + D$, and the associated stable, but not necessarily minimal, realization of a linear time-invariant (LTI) system,

$$\begin{array}{rcl} \delta x(t) & = & Ax(t) + Bu(t), \quad t > 0, \quad x(0) = x^0, \\ y(t) & = & Cx(t) + Du(t), \quad t \geq 0 \end{array} \tag{1}$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, and $D \in \mathbb{R}^{p\times m}$. The number of state variables n is said to be the order of the system. If $t \in \mathbb{R}$ and $\delta x(t) = \dot{x}(t)$, the system is a continuous-time system while (1) describes a discrete-time system if $t \in \mathbb{N}_0$ and $\delta x(t) = x(t+1)$ is the forward shift operator.

The aim is to find a reduced order LTI system,

$$\begin{array}{rcl} \delta x_r(t) & = & A_r x_r(t) + B_r u_r(t), \quad t > 0, \quad x_r(0) = x_r^0, \\ y_r(t) & = & C_r x_r(t) + D_r u_r(t), \quad t \geq 0, \end{array} \tag{2}$$

of order r, $r \ll n$, such that the TFM $G_r(\lambda) = C_r(\lambda I - A_r)^{-1}B_r + D_r$ approximates (1).

A large class of model reduction methods rely on similarity transformations, where given $Y = \left[T_l^T, L_l^T\right]^T \in \mathbb{R}^{n\times n}$ and $Y^{-1} = [T_r, L_r]$, with $T_l \in \mathbb{R}^{r\times n}$ and $T_r \in \mathbb{R}^{n\times r}$, the reduced model is defined as $A_r = T_l A T_r$, $B_r = T_l B$, $C_r = CT_r$, and $D_r = D$. The methods used here all belong to this class. Moreover, hereafter, we assume that the spectrum of A as denoted by $\Lambda(A)$ is contained in the open left half plane. This implies that the system (1) is stable, that is, all the poles of $G(s)$ have strictly negative real parts.

*Partially supported by the DAAD programme *Acciones Integradas Hispano-Alemanas*. Enrique S. Quintana-Ortí and Gregorio Quintana-Ortí's research is funded by the project GV-59-1-14 of the *Consellería de Educación y Cultura*.

[†]Zentrum für Technomathematik, Fachbereich 3 – Mathematik und Informatik, Universität Bremen, D–28334 Bremen, Germany. E-mail: benner@math.uni-bremen.de

[‡]Departamento de Informática, Universidad Jaime I, 12080 Castellón, Spain. E-mail quintana@inf.uji.es

[§]Departamento de Informática, Universidad Jaime I, 12080 Castellón, Spain. E-mail gquintan@inf.uji.es

There is no general technique for model reduction that can be considered as optimal in an overall sense since the reliability, performance and adequacy of the reduced system strongly depends on the system characteristics. Model reduction methods usually differ in the error measure they attempt to minimize. The methods considered here are balanced truncation (BT) methods [18, 20, 22, 23], singular perturbation approximation (SPA) methods [17], and Hankel-norm approximation (HNA) methods [12]. They all belong to the family of absolute error methods, which try to minimize $\|\Delta_a\| = \|G - G_r\|$. For BT and SPA, the considered system norm is the $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norm of a stable, rational matrix function, defined by.

$$\|G\|_\infty = \operatorname*{ess\,sup}_{\omega\in\mathbb{R}} \sigma_{\max}(G(\jmath\omega)), \tag{3}$$

where $\jmath := \sqrt{-1}$ and $\sigma_{\max}(M)$ is the largest singular value of the matrix M. In the HNA method, the error is measured in the *Hankel norm* given by $\|G\|_H := \sigma_1$, where σ_1 is the largest *Hankel singular value* of the system, see (6) below.

All model reduction methods considered here are strongly related to the controllability Gramian W_c and the observability Gramian W_o of the system (1). For continuous-time systems the Gramians are given by the solutions of two "coupled" (as they share the same coefficient matrix A) *Lyapunov equations*

$$AW_c + W_cA^T + BB^T = 0, \qquad A^TW_o + W_oA + C^TC = 0, \tag{4}$$

while in the discrete-time case, the Gramians satisfy the *Stein equations* (or discrete Lyapunov equations)

$$AW_cA^T - W_c + BB^T = 0, \qquad A^TW_oA - W_o + C^TC = 0. \tag{5}$$

The Hankel singular values of the systems are given by the square-roots of the eigenvalues of W_cW_o, i.e.,

$$\Lambda\left(W_cW_o\right) = \{\sigma_1^2, \dots, \sigma_n^2\}, \qquad \sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0. \tag{6}$$

As A is assumed to be stable, W_c and W_o are positive semidefinite and therefore can be factored as $W_c = S^TS$ and $W_o = R^TR$. The factors S and R are called the *Cholesky factors* of the Gramians. These factors are usually computed using Hammarling's method [14], yielding $S, R \in \mathbb{R}^{n\times n}$ upper triangular. Numerically reliable model reduction methods use the Cholesky factors rather than the Gramians themselves [25]. This approach is implemented in the SLICOT subroutines `AB09AD`, `AB09BD`, and `AB09CD` for BT, SPA, and HNA model reduction, respectively, [24]. On current desktop computers, these routines can be applied to systems up to order $n \approx 1000$.

The goal of the subroutines proposed here is to enable users to apply the same model reduction techniques to systems of larger dimensions. This is appealing as the methods mentioned above yield computable error bounds for the approximation error while techniques for large sparse systems based on Padé approximations or moment matching do not provide sufficient information to bound the errors; see, e.g., [11, 13]. The way to achieve this goal is to employ parallel computing techniques. Using communication libraries like PVM or MPI and basing our routines on the numerical linear algebra kernels provided in ScaLAPACK [9], it is possible to solve large-scale problems on clusters of PCs or workstations. That is, relatively cheap infrastructure can be used to employ the implemented routines rather than having to rely on expensive parallel architectures as, for example, the IBM SP– or CRAY T3- platforms.

The major computational step in the model reduction techniques considered here is the solution of the Lyapunov or Stein equations in (4) or (5). The algorithms in SLICOT that compute the required Cholesky factors of the Gramians are based on the QR algorithm. The implementation of this algorithm on parallel distributed memory architectures has been reported several times as a notoriously difficult task. The algorithm itself is not scalable [15], and the parallelization results [15] are far from those of usual matrix factorizations [9]. Therefore we employ a technique based on the matrix sign function. It is well-known that sign function-based computations are easy to parallelize and yield very efficient code; see, e.g., [1]. The solution of Lyapunov equations using the sign function method was first described in [19]. Efficient algorithms for solving a pair of

Lyapunov equations as in (4) are reported in [2, 16]. The counterpart of the sign function method for solving Stein equations as in (5) is the squared Smith method [21]. The application of this method to a pair of Stein equations as in (5) is described in [7, 8].

Besides parallelization issues, sign function-based Lyapunov solvers and Stein solvers using the squared Smith method also have some other advantages in this context. As large-scale systems tend to be (numerically) non-minimal, i.e., W_c and/or W_o are (numerically) singular, the Cholesky factors are not full-rank matrices. Their (numerical) rank is often much lower than n. Using an iteration that computes the factors of the *full-rank factorizations* $W_c = \hat{S}^T\hat{S}$ and $\hat{W}_o = \hat{R}^T\hat{R}$, i.e., $\hat{S} \in \mathbb{R}^{\mathrm{rank}(W_c)\times n}$, $\hat{R} \in \mathbb{R}^{\mathrm{rank}(W_o)\times n}$ can save a significant amount of computational cost and workspace when solving the Lyapunov equations and particularly during the subsequent computations for computing the reduced-order model; see [5, 6, 8] for details. This approach also has some advantages regarding numerical robustness when determining the McMillan degree and a minimal realization of an LTI system.

In the next section we list the routines implemented for use in PSLICOT [10]. In Section 3 we report some numerical results obtained on a Linux PC cluster.

2 Model Reduction Subroutines for PSLICOT

PSLICOT is an extension of the *Subroutine Library in Control Theory* [3], SLICOT, for parallel computations. It uses the ScaLAPACK [9] programming and parallelization model. It is assumed that a communication library as MPI or PVM is installed. On top of this, the Basic Linear Algebra Communication Subroutines (BLACS) are used. Parallel numerical linear algebra computations are performed using parallel PB-BLAS and ScaLAPACK routines as far as possible while local computations on one node employ BLAS and LAPACK. The PSLICOT implementation and documentation standard is described in [10].

The functionality of the routines to be implemented in PSLICOT should duplicate that of corresponding SLICOT routines. PSLICOT routines are developed only for areas in which parallel computations are considered useful, i.e., where large-scale problems need to be addressed. Model reduction is one of these areas. Although the functionality of SLICOT routines is duplicated, the implementation of the corresponding subroutine needs not be duplicated in general. Often, parallel computations involve implementation aspects that make algorithms attractive which may not be the best choice for serial computations. The selected model reduction algorithms are an example for this as outlined in the introduction. The solution of the Lyapunov or Stein equations (4), (5) is based on the sign function method or the squared Smith iteration rather than the Bartels-Stewart or Hammarling's method employed in the corresponding SLICOT routines [25].

The SLICOT routines are organized such that they can be applied to either continuous- or discrete-time systems. Moreover, for `AB09AD` and `AB09BD`, there is an option to compute a balanced reduced-order model using the Cholesky factors of the Gramians or to avoid the balancing of the reduced-order model. The first approach is referred to as the SR (square-root) method while the second option is called the BFSR (balancing-free) square-root method. (The term square-root method in this context comes from the fact that first descriptions of BT methods used the positive (semi-)definite square-roots of the Gramians rather than the Cholesky factors.) The routines implemented for PSLICOT duplicate this functionality, i.e., they can be applied to either continuous- or discrete-time systems and the SR or BFSR method can be selected.

The routines listed below are based on implementations for the *Parallel Library in Control*[1] (PLiC) [4]. In this library, ScaLAPACK based subroutines for analysis and design of linear systems are collected. Besides model reduction routines, it contains subroutines for the numerical solution of linear and quadratic matrix equations, partial stabilization, computation of staircase forms and special matrix factorizations.

The following model reduction subroutines, taken from the PLiC, have been standardized and documented according to [10] for inclusion into PSLICOT. All routines listed can be used for both, continuous- and discrete time systems.

[1] see `http://anna.act.uji.es/psc.html`

PAB09AX computes reduced (or minimal) order balanced models using either the SR or BFSR BT method.

PAB09BX computes reduced order models using either the BFSR or SR SPA method.

PAB09CX computes reduced order models using the optimal HNA method based on a minimal realization of the system computed with the SR version of `PAB09AX`.

The following PLiC subroutines are called from the model reduction routines.

PDCECLNC solves coupled stable Lyapunov equations for full-rank factors using a sign function-based method.

PDCEDLSC solves coupled stable Stein equations for full-rank factors using the squared Smith iteration or the sign function method.

PDGESINE computes the sign function of a matrix using the Newton iteration.

PDGECSNE solves a stable Sylvester equation using the sign function method.

PDGEPSVD computes the singular value decomposition of a matrix product.

PDGESSBA balances the system defined by (A, B, C, D).

PDTRRN estimates the rank of a triangular matrix using an incremental condition estimator.

3 Performance Results

The parallel routines described in the previous section have been tested on a variety of examples. The accuracy of the reduced-order model is reported to be comparable to that of the corresponding SLICOT routines in [5, 6, 8]. Here, we show only some results regarding the parallel performance of some of the routines listed in Section 2.

The figures in 1 show the execution times of SLICOT routines (results on 1 processor) and corresponding PSLICOT routines (results on $n_p \geq 2$ processors). In both examples, we generated random systems with prescribed McMillan degree $t = 50$. The order of the systems was $n = 1000$ with $m = p = 100$ inputs and outputs. All the Cholesky factors have rank equal to 100. In the experiments we computed minimal realizations of the system, i.e., a reduced-order model of order $r = 50$ was computed.

Both figures show a high efficiency and speed-up for the parallel algorithms for small numbers of processors. We even have superspeed-ups caused by the better efficiency of the sign function-based or squared Smith iterations compared to the algorithms based on the QR algorithm used in the SLICOT routines. For a larger number of processors, the execution times do not drop significantly. This is due to the small size of the problem in this case. Better performance is achieved for larger problems, but these problems can not be solved using the SLICOT routines on 1 processor due to the lack of sufficient memory.

Further numerical examples, performance results, and more details on the implementation and parallelization of the routines can be found in [5, 6, 8].

4 Concluding Remarks

We have described subroutines for model reduction based on balanced truncation, singular perturbation approximation and optimal Hankel norm approximation. The subroutines are intended to be used on parallel computers with distributed memory architectures. They allow to apply model reduction techniques with computable error bounds to stable systems of order up to $\mathcal{O}(10^5)$ depending on the available memory in the parallel computer or cluster. For the future it is planned to implement algorithms for relative error model reduction employing similar techniques and to use the currently available PSLICOT routines for model reduction of unstable systems using an additive decomposition of the system.

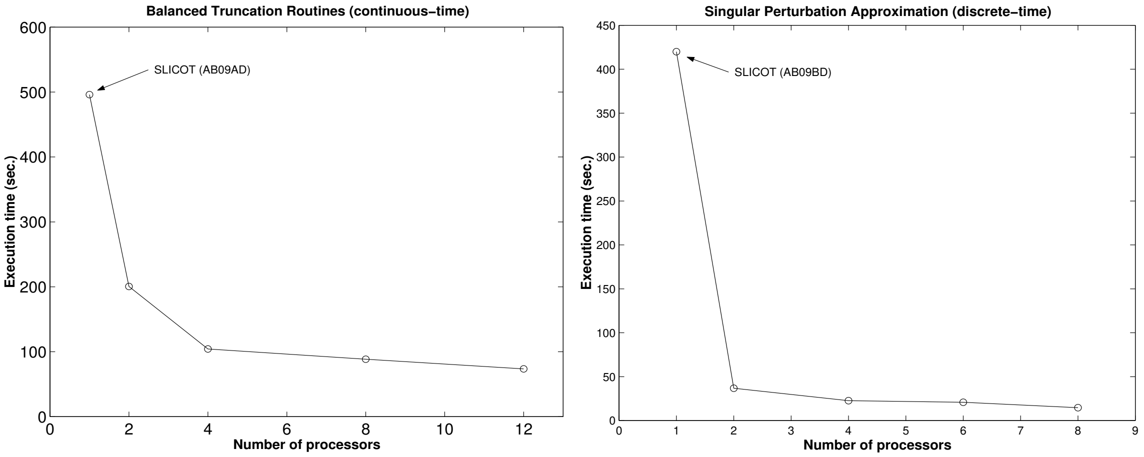

Figure 1: Execution time vs. number of nodes of the SLICOT/PSLICOT BT and SPA algorithms (SR versions)

References

[1] Z. Bai, J. Demmel, J. Dongarra, A. Petitet, H. Robinson, and K. Stanley. The spectral decomposition of nonsymmetric matrices on distributed memory parallel computers. *SIAM J. Sci. Comput.*, 18:1446–1461, 1997.

[2] P. Benner, J.M. Claver, and E.S. Quintana-Ortí. Efficient solution of coupled Lyapunov equations via matrix sign function iteration. In A. Dourado et al., (ed.), *Proc.* 3$^{\text{rd}}$ *Portuguese Conf. on Automatic Control CONTROLO'98,* Coimbra, pp. 205–210, 1998.

[3] P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT - a subroutine library in systems and control theory. In B.N. Datta, (ed.), *Applied and Computational Control, Signals, and Circuits*, vol. 1, ch. 10, pp. 499–539. Birkhäuser, Boston, MA, 1999.

[4] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. A portable subroutine library for solving linear control problems on distributed memory computers. In G. Cooperman, E. Jessen, and G.O. Michler (ed.), *Workshop on Distributed High Performance Computing and Gigabit Wide Area Networks, Essen (Germany), September 1998*, vol. 249 of *Lecture Notes in Control and Information Sciences*, pp. 61–88. Springer-Verlag, Berlin/Heidelberg, 1999.

[5] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Balanced truncation model reduction of large-scale dense systems on parallel computers. To appear in *Math. Comp. Model. Dyn. Syst.*, Vol. 6, No. 4, 2000.

[6] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Singular perturbation approximation of large, dense linear systems. In *Proc. 2000 IEEE Intl. Symp. CACSD, Anchorage, Alaska, USA, September 25–27, 2000*, pp. 255–260. Omnipress, Madison, WI, 2000.

[7] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Efficient numerical model reduction methods for discrete-time systems. In I. Troch and F. Breitenecker, (ed.), *Proc. IMACS Symp. Math. Modelling, 3rd MATHMOD Vienna,* Vol. 1, number 15 in ARGESIM Reports, pp. 277–280. Vienna (Austria), February 2000.

[8] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Parallel algorithms for model reduction of discrete-time systems. Berichte aus der Technomathematik, Report 00–18[2], FB3 – Mathematik und Informatik, Universität Bremen, 28334 Bremen (Germany), December 2000.

[2]Available from `http://www.math.uni-bremen.de/zetem/berichte.html`.

[9] L.S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R.C. Whaley. *ScaLAPACK Users' Guide*. SIAM, Philadelphia, PA, 1997.

[10] I. Blanquer, D. Guerrero, V. Hernandez, E. Quintana-Ortí, and P. Ruiz. Parallel-SLICOT implementation and documentation standards. SLICOT Working Note 1998–1[2], The Working Group on Software (WGS), September 1998.

[11] R. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In B.N. Datta, (ed.), *Applied and Computational Control, Signals, and Circuits*, vol. 1, ch. 9, pp. 435–498. Birkhäuser, Boston, MA, 1999.

[12] K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *Internat. J. Control*, 39:1115–1193, 1984.

[13] E.J. Grimme, D.C. Sorensen, and P. Van Dooren. Model reduction of state space systems via an implicitly restarted Lanczos method. *Numer. Algorithms*, 12:1–31, 1996.

[14] S.J. Hammarling. Numerical solution of the stable, non-negative definite Lyapunov equation. *IMA J. Numer. Anal.*, 2:303–323, 1982.

[15] G. Henry and R. van de Geijn. Parallelizing the QR algorithm for the unsymmetric algebraic eigenvalue problem: myths and reality. *SIAM J. Sci. Comput.*, 17:870–883, 1997.

[16] W. Lang and U. Lezius. Numerical realization of the balanced reduction of a control problem. In H. Neunzert, (ed.), *Progress in Industrial Mathematics at ECMI94*, pp. 504–512. John Wiley & Sons Ltd and B.G. Teubner, New York and Leipzig, 1996.

[17] Y. Liu and B. Anderson. Controller reduction via stable factorization and balancing. *Internat. J. Control*, 44:507–531, 1986.

[18] B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26:17–32, 1981.

[19] J.D. Roberts. Linear model reduction and solution of the algebraic Riccati equation by use of the sign function. *Internat. J. Control*, 32:677–687, 1980. (Reprint of Technical Report No. TR-13, CUED/B-Control, Cambridge University, Engineering Department, 1971).

[20] M.G. Safonov and R.Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, AC–34:729–733, 1989.

[21] R.A. Smith. Matrix equation $XA + BX = C$. *SIAM J. Appl. Math.*, 16(1):198–201, 1968.

[22] M.S. Tombs and I. Postlethwaite. Truncated balanced realization of a stable non-minimal state-space system. *Internat. J. Control*, 46(4):1319–1330, 1987.

[23] A. Varga. Efficient minimal realization procedure based on balancing. In *Prepr. of the IMACS Symp. on Modelling and Control of Technological Systems*, vol. 2, pp. 42–47, 1991.

[24] A. Varga. Task II.A.1 – selection of model reduction routines. SLICOT Working Note 1998–2[2], The Working Group on Software (WGS), June 1998.

[25] A. Varga. Model reduction routines for SLICOT. NICONET Report 1999–8[3], The Working Group on Software (WGS), June 1999.

[3]Available from `http://www.win.tue.nl/niconet/NIC2/reports.html`.