

Parallel Model Reduction of Large-Scale Linear Descriptor Systems via Balanced Truncation*

Peter Benner¹, Enrique S. Quintana-Ortí², and
Gregorio Quintana-Ortí²

¹ Fakultät für Mathematik, Technische Universität Chemnitz,
D-09107 Chemnitz, Germany; benner@mathematik.tu-chemnitz.de.

Tel.: +371-531-8367. Fax: +371-531-2657.

² Depto. de Ingeniería y Ciencia de Computadores, Universidad Jaume I,
12.071-Castellón, Spain; {quintana,gquintan}@icc.uji.es.

Tel.: +34-964-728257. Fax: +34-964-728486.

Abstract. In this paper we investigate the use of parallel computing to deal with the high computational cost of numerical algorithms for model reduction of large linear descriptor systems. The state-space truncation methods considered here are composed of iterative schemes for the disk and sign functions which can be efficiently implemented on parallel architectures using the kernels from existing parallel linear algebra libraries.

Our experimental results on a cluster of Intel Pentium Xeon processors show the performance and scalability of the parallel algorithms.

Key words: Model reduction, balanced truncation, linear descriptor systems, parallel algorithms.

Topics: Numerical methods, cluster computing, large scale simulations in control.

1 Introduction

Model reduction is of fundamental importance in many modeling and control applications involving continuous-time linear descriptor systems. In generalized state-space form these systems are defined by

$$\begin{aligned} E\dot{x}(t) &= Ax(t) + Bu(t), & t > 0, & \quad x(0) = x^0, \\ y(t) &= Cx(t) + Du(t), & t \geq 0, \end{aligned} \quad (1)$$

where $E, A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$, $C \in \mathbb{R}^{p \times n}$, $D \in \mathbb{R}^{p \times m}$, $x^0 \in \mathbb{R}^n$ is the initial state of the system, and n is said to be the order (or degree) of the system. The

* P. Benner was supported by the DFG Research Center “Mathematics for key technologies” (FZT 86) in Berlin. E.S. Quintana-Ortí and G. Quintana-Ortí were supported by the CICYT project No. TIC2002-004400-C03-01.

transfer function matrix (TFM) associated with the system (1) is then given by $G(s) = C(sE - A)^{-1}B + D$. Hereafter, we assume that the matrix pencil $A - \lambda E$ is regular, i.e., $\exists \lambda \in \mathbb{C}$ with $\det(A - \lambda E) \neq 0$. We also assume that $A - \lambda E$ is stable, implying that the finite part of the spectrum of $A - \lambda E$ is contained in the open left half plane.

In the model reduction problem, we are interested in finding a reduced-order realization

$$\begin{aligned} \hat{E}\hat{x}(t) &= \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t), & t > 0 & \quad \hat{x}(0) = \hat{x}^0, \\ \hat{y}(t) &= \hat{C}\hat{x}(t) + \hat{D}\hat{u}(t), & t \geq 0, \end{aligned} \quad (2)$$

of order r , $r \ll n$, and associated TFM $\hat{G}(s) = \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D}$ which approximates $G(s)$.

Model reduction of large-scale linear descriptor systems arises in control of multibody (mechanical) systems, manipulation of fluid flow (e.g., (Navier)-Stokes equations), circuit simulation, VLSI chip design, in particular when modeling the interconnections via RLC networks, and simulation of MEMS and NEMS (micro- and nano-electro-mechanical systems), to name a few; see, e.g., [11, 12, 14, 26]. State-space dimensions n of order 10^2 to 10^4 are common in these applications.

There is no general technique for model reduction that can be considered as optimal in an overall sense. In this paper we focus on the so-called state-space truncation approach [2, 25] and, in particular, on balanced truncation (BT) of the system [24, 31, 36, 38]. BT methods belong to the family of absolute error methods, which try to minimize $\|\Delta_a\|_\infty = \|G - \hat{G}\|_\infty$. Here, $\|G\|_\infty$ denotes the \mathcal{L}_∞ - or \mathcal{H}_∞ -norm of a stable, rational matrix function which is defined for proper transfer functions as

$$\|G\|_\infty = \operatorname{ess\,sup}_{\omega \in \mathbb{R}, \omega \geq 0} \sigma_{\max}(G(j\omega)), \quad (3)$$

where $j := \sqrt{-1}$ and $\sigma_{\max}(M)$ is the largest singular value of the matrix M . Note that transfer functions of descriptor systems are in general not proper. BT and \mathcal{H}_∞ -norm considerations are therefore not always applicable directly. A way to circumvent these difficulties is presented in [5, 32, 33]. But BT methods for linear descriptor systems as proposed in [32, 33] with dense state matrix pencil $A - \lambda E$ present a computational cost of $\mathcal{O}(n^3)$ floating-point operations (flops). Large-scale applications, like those enumerated above, thus clearly benefit from using parallel computing techniques to obtain the reduced-order system. Parallel model reduction of standard linear systems ($E = I_n$) using the state-space approach has been investigated elsewhere; see, e.g., the review in [9].

Although there exist several other approaches for model reduction, see, e.g., [2, 14, 15], those are specialized for certain problem classes and often lack properties like error bounds or preservation of stability, passivity, or phase information.

Our procedure for parallel model reduction of linear descriptor systems is composed of two major tasks: First, it is necessary to decouple (separate) the finite poles of the system from the infinite ones, which is equivalent to computing

an additive decomposition of the TFM. Then, in a second step, BT is applied to the part of the system associated with the finite poles, which requires the solution of two generalized Lyapunov equations.

The rest of the paper is structured as follows. In Section 2 we review specific algorithms for the two major tasks involved in model reduction of linear descriptor systems. In Section 3 we then introduce two numerical tools, namely the disk and sign functions, that are the basis for our model reduction algorithms. In particular, an inverse-free iteration for the disk function is employed for the separation of the poles of the system into the appropriate regions, while the generalized Lyapunov equations are solved by means of a modified Newton iteration for the sign function of a matrix pencil. A few details on the implementation and parallelization of the resulting algorithm are given in Section 4. Finally, the numerical and parallel performances of the algorithms are reported in Section 5, and some concluding remarks follow in Section 6.

2 Model Reduction of Descriptor Systems

In this section we briefly describe a model reduction method for linear descriptor systems proposed in [5]. It is based on the observation that the method of [32, 33] is mathematically equivalent to the following procedure. First, the TFM of the system is decomposed as

$$G(s) = G_0(s) + G_\infty(s),$$

where G_0 and G_∞ contain, respectively, the finite and infinite poles of the system. This is followed by the application of BT to approximate $G_0(s)$ by $\hat{G}_0(s)$. The reduced-order model is then given by

$$\hat{G}(s) = \hat{G}_0(s) + G_\infty(s).$$

In [34] it is also shown how to apply model reduction to the infinite part of the system. We will not consider this here any further, but the procedure described there should accompany the method discussed here.

2.1 Decoupling the infinite poles of a TFM

Consider the system defined by $(A - \lambda E, B, C, D)$. In order to compute the sought-after additive decomposition, we are interested in finding an equivalence transformation defined by a pair of nonsingular matrices $U, V \in \mathbb{R}^{n \times n}$, such that $A - \lambda E$ is “block-diagonalized” as

$$\hat{A} - \lambda \hat{E} := U(A - \lambda E)V^{-1} = \begin{bmatrix} A_0 & 0 \\ 0 & A_\infty \end{bmatrix} - \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix},$$

where $A_0 - \lambda E_0$ and $A_\infty - \lambda E_\infty$ contain, respectively, the finite and infinite eigenvalues of $A - \lambda E$.

If this equivalence transformation is now applied to the rest of the system

$$\hat{B} := UB =: \begin{bmatrix} B_0 \\ B_\infty \end{bmatrix}, \quad \hat{C} := CV^{-1} =: [C_0 \ C_\infty],$$

we obtain the desired additive decomposition of the TFM into

$$\begin{aligned} G(s) &= C(sE - A)^{-1}B + D \\ &= \hat{C}(s\hat{E} - \hat{A})^{-1}\hat{B} + \hat{D} \\ &= [C_0 \ C_\infty] \begin{bmatrix} (sE_0 - A_0)^{-1} & \\ & (sE_\infty - A_\infty)^{-1} \end{bmatrix} \begin{bmatrix} B_0 \\ B_\infty \end{bmatrix} + D \\ &= \underbrace{\{C_0(sE_0 - A_0)^{-1}B_0 + D\}}_{=:G_0(s)} + \underbrace{\{C_\infty(sE_\infty - A_\infty)^{-1}B_\infty\}}_{=:G_\infty(s)}. \end{aligned}$$

Thus, the problem of computing an additive decomposition has been redefined in terms of finding appropriate matrices U and V that block-diagonalize the system.

In order to compute these matrices we will proceed in two steps. First, we compute orthogonal matrices $Q, Z \in \mathbb{R}^{n \times n}$ that reduce $A - \lambda E$ to block-triangular form; i.e.,

$$Q^T(A - \lambda E)Z = \begin{bmatrix} A_0 & W_A \\ 0 & A_\infty \end{bmatrix} - \lambda \begin{bmatrix} E_0 & W_E \\ 0 & E_\infty \end{bmatrix}. \quad (4)$$

In subsection 3.1 we review a numerical tool: the matrix disk function [23], that allows the computation of the appropriate “block-triangularizing” transformation.

Then, in a second step, (4) is further reduced into the required block-diagonal form by solving the following generalized Sylvester equation

$$A_0Y + XA_\infty + W_A = 0, \quad E_0Y + XE_\infty + W_E = 0. \quad (5)$$

Thus,

$$\begin{aligned} \hat{A} - \lambda \hat{E} &:= \hat{U}(A - \lambda E)\hat{V}^{-1} \\ &:= \begin{bmatrix} I & X \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} A_0 & W_A \\ 0 & A_\infty \end{bmatrix} - \lambda \begin{bmatrix} E_0 & W_E \\ 0 & E_\infty \end{bmatrix} \right) \begin{bmatrix} I & Y \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} A_0 & 0 \\ 0 & A_\infty \end{bmatrix} - \lambda \begin{bmatrix} E_0 & 0 \\ 0 & E_\infty \end{bmatrix}, \end{aligned}$$

and $U = \hat{U}Q^T$, $V^{-1} = Z\hat{V}^{-1}$.

In case the system is of index 1, denoted by $\text{ind}(A, E) = 1$ (i.e., the infinite eigenvalues of $A - \lambda E$ are non-defective), we obtain that in (4) $E_\infty = 0$ and the Sylvester equation in (5) decouples into two linear systems

$$E_0Y = -W_E, \quad XA_\infty = -(W_A + A_0Y).$$

In case the index is greater than one, (5) can be solved using the generalized Schur or Hessenberg-Schur methods proposed in [13, 17]. However, the parallelization of both algorithms requires a parallel implementation of the QZ algorithm, an extremely difficult task (no parallel version exists yet!); besides, many applications with $\text{ind}(A, E) > 1$ have an equivalent formulation with $\text{ind}(A, E) = 1$. We therefore assume that the index is one hereafter. Note that this implies that the system has no impulsive modes.

2.2 Balanced truncation of linear descriptor systems

Consider now the realization $(A_0 - \lambda E_0, B_0, C_0, D_0)$ resulting from the deflation of the infinite part of $(A - \lambda E, B, C, D)$ using the procedure described in the previous section. In the second step we will proceed to apply BT to reduce this realization further, noting that E_0 is invertible and the deflated system is equivalent to the standard state-space system given by $(E_0^{-1}A_0, E_0^{-1}B_0, C_0, D_0)$.

BT methods are strongly related to the controllability Gramian W_c and the observability Gramian W_o of the system. In the continuous-time case, these Gramians are given by the solutions of two coupled *generalized Lyapunov equations*

$$\begin{aligned} A_0 W_c E_0^T + E_0 W_c A_0^T + B_0 B_0^T &= 0, \\ A_0^T \hat{W}_o E_0 + E_0^T \hat{W}_o A_0 + C_0^T C_0 &= 0, \quad W_o = E^T \hat{W}_o E. \end{aligned} \quad (6)$$

As $A - \lambda E$ was assumed to be stable, so is $A_0 - \lambda E_0$, and the Gramians W_c and W_o are positive semidefinite; therefore there exist factorizations $W_c = S^T S$ and $W_o = R^T R$. Matrices S and R are called the *Cholesky factors* of the Gramians (even if they are not Cholesky factors in a strict sense). Efficient parallel algorithms for the solution of (6) are discussed in subsection 3.2.

Consider now the singular value decomposition (SVD)

$$SR^T = [U_1 \ U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \quad (7)$$

where the matrices are partitioned at a given dimension r such that $\Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_r)$, $\Sigma_2 = \text{diag}(\sigma_{r+1}, \dots, \sigma_n)$, $\sigma_j \geq 0$ for all j , and $\sigma_r > \sigma_{r+1}$. Here, $\sigma_1, \dots, \sigma_n$ are the *Hankel singular values* of the system.

The so-called *square-root* (SR) BT algorithms determine the reduced-order model as

$$\hat{E}_0 := LE_0T, \quad \hat{A}_0 := LA_0T, \quad \hat{B}_0 := LB_0, \quad \hat{C}_0 := C_0T, \quad \hat{D}_0 := D_0, \quad (8)$$

where the truncation matrices L and T are given by

$$L = \Sigma_1^{-1/2} V_1^T R E_0^{-1} \quad \text{and} \quad T = S^T U_1 \Sigma_1^{-1/2}. \quad (9)$$

Note that $\hat{E}_0 = I_r$ and needs not be computed. Even though the reduction yields a standard state-space representation, it is still beneficial not to transform the generalized system to standard form by inverting E_0 directly as rounding

errors introduced by this operation are delayed as much as possible in the given procedure and therefore intermediate results are not corrupted by them.

The *balancing-free square-root* (BFSR) BT algorithms often provide more accurate reduced-order models in the presence of rounding errors [38]. These algorithms share the first two stages (solving the coupled equations and computing the SVD of SR^T) with the SR methods, but differ in the procedure to obtain L and T . Specifically, the following two QR factorizations are computed,

$$S^T U_1 = [P_1 \ P_2] \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}, \quad E_0^{-T} R^T V_1 = [Q_1 \ Q_2] \begin{bmatrix} \bar{R} \\ 0 \end{bmatrix}, \quad (10)$$

where $P_1, Q_1 \in \mathbb{R}^{n \times r}$ have orthonormal columns, and $\hat{R}, \bar{R} \in \mathbb{R}^{r \times r}$ are upper triangular. The reduced-order system is then given by the projection matrices

$$L = (Q_1^T P_1)^{-1} Q_1^T, \quad T = P_1, \quad (11)$$

and (8).

Both SR and BFSR BT algorithms provide a realization of order r that satisfies the error bound [19]

$$\|\Delta_a\|_\infty = \|G_0 - \hat{G}_0\|_\infty \leq 2 \sum_{j=r+1}^n \sigma_j, \quad (12)$$

allowing thus an adaptive choice of the state-space dimension r of the reduced-order model. Also note that as $G - \hat{G} = G_0 - \hat{G}_0$ and $G - \hat{G}$ is proper, the error bound (12) applies also for the complete reduced-order model.

3 Numerical Tools

The model reduction method described in the previous section involves, as the two major computational problems, the computation of an equivalence transformation that divides the eigenspectrum of $A - \lambda E$, $\Lambda(A, E)$, into the appropriate parts, and the solution of two coupled generalized Lyapunov equations. We next review efficient iterative algorithms for both computations that yield high performance on parallel distributed memory computers.

3.1 The inverse-free iteration for the disk function

In [23], Malyshev proposed an iterative scheme for the spectral division of a matrix pencil. Given $F - \lambda G$, $F, G \in \mathbb{R}^{n \times n}$, with no eigenvalues on the unit circle, the technique described there provides orthogonal matrices $Q, Z \in \mathbb{R}^{n \times n}$ such that

$$Q^T (F - \lambda G) Z = \begin{bmatrix} F_{11} & F_{12} \\ 0 & F_{22} \end{bmatrix} - \lambda \begin{bmatrix} G_{11} & G_{12} \\ 0 & G_{22} \end{bmatrix}, \quad (13)$$

where $F_{11} - \lambda G_{11}$ and $F_{22} - \lambda G_{22}$ contain, respectively, the eigenvalues of $F - \lambda G$ inside and outside the unit circle. A careful use of Moëbius transformations

allows to separate the spectrum along other regions. In particular, the finite eigenvalues of $\Lambda(F, G)$ can be separated from the infinite ones by applying the method described next to the matrix pencil $\alpha F - \lambda G$, with $\alpha > \max\{|\lambda|, \lambda \in \Lambda(F, G) \setminus \{\infty\}\}$. However, it remains an open question how to estimate the value of α .

Malyshev's iteration was further refined and made truly inverse-free in [4], where it was redefined as follows:

$$\begin{aligned}
& F_0 \leftarrow F, G_0 \leftarrow G \\
& \text{for } j = 0, 1, 2, \dots \\
& \quad \text{Compute the QR decomposition} \\
& \quad \begin{bmatrix} F_j \\ -G_j \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} R_j \\ 0 \end{bmatrix} \\
& \quad F_{j+1} \leftarrow U_{12}^T F_j \\
& \quad G_{j+1} \leftarrow U_{22}^T G_j
\end{aligned}$$

As the convergence rate of the inverse-free iteration is ultimately quadratic, a suitable convergence criterion is to stop when

$$\|R_{j+1} - R_j\|_1 < \tau := c\varepsilon,$$

with c a certain threshold value and ε the machine precision. In practice, however, it is better to set $\tau = n\sqrt{\varepsilon}\|R_j\|_1$ and perform two more iterations once this criterion is satisfied. Due to the ultimate quadratic convergence of the iteration this ensures in most cases the maximum possible accuracy while avoiding convergence stagnation problems.

The iteration can be used both for the computation of the matrix disk function and for spectral division. For instance, let $F_\infty := \lim_{j \rightarrow \infty} F_j$ and $G_\infty := \lim_{j \rightarrow \infty} G_j$, then the disk function of the matrix pencil is defined as

$$\text{disk}(F, G) = (F_\infty + G_\infty)^{-1}(F_\infty - \lambda G_\infty).$$

The procedure for spectral division was further improved in [35], where an inverse-free subspace extraction scheme was described that provides both Q and Z in (13) from a single inverse-free iteration, saving thus half of the computational cost. Specifically, this technique requires first the computation of the rank-revealing QR (RRQR) factorization [20]

$$F_\infty^T = \bar{Z}\bar{R}\bar{\Pi}, \quad r = \text{rank}(F_\infty) = \text{rank}(\bar{R}).$$

Now, let $\bar{Z} = [Z_1, Z_2]$, $Z_1 \in \mathbb{R}^{n \times (n-r)}$, $Z_2 \in \mathbb{R}^{n \times r}$. The sought-after orthogonal matrices are then given by $Z = [Z_2, Z_1]$ and the orthogonal matrix in the RRQR factorization

$$[AZ_2, BZ_2] = QR\Pi.$$

The parallelization of the inverse-free iteration has been previously reported in [3, 27].

3.2 Solving generalized Lyapunov equations via the sign function

Since its introduction in [30], the sign function has proved useful in a variety of numerical linear algebra problems which include, among others, spectral division, solution of algebraic Riccati equations, and solution of linear matrix equations; see, e.g., [7, 16, 30].

We next describe a special variant of the Newton iteration for the matrix sign function that can be used for the solution of the generalized Lyapunov equations in (6). Specifically, in our method the equations are solved for the Cholesky factors of the solutions allowing great computational savings in case they are applied to the type of equations arising in large-scale non-minimal systems. For a detailed description of this technique, see, e.g., [8].

In brief, the proposed variant of the Newton iteration for the sign function is defined as follows:

$$\begin{aligned}
& A_0 \leftarrow A, S_0 \leftarrow B, R_0 \leftarrow C \\
& \text{for } j = 0, 1, 2, \dots \\
& \quad c_j = (\det(A_j) / \det(E_j))^{1/n} \\
& \quad A_{j+1} \leftarrow \frac{1}{\sqrt{2c_j}} (A_j + c_j^2 E A_j^{-1} E) \\
& \quad \text{Compute the RRQR decompositions} \\
& \quad \quad \frac{1}{\sqrt{2c_j}} [S_j \quad c_j E A_j^{-1} S_j]^T = Q_s \begin{bmatrix} R_s \\ 0 \end{bmatrix} \Pi_s \\
& \quad \quad \frac{1}{\sqrt{2c_j}} \begin{bmatrix} R_j \\ c_j R_j A_j^{-1} E \end{bmatrix} = Q_r \begin{bmatrix} R_r \\ 0 \end{bmatrix} \Pi_r \\
& \quad R_{j+1} \leftarrow (R_r \Pi_r)^T \\
& \quad S_{j+1} \leftarrow R_s \Pi_s
\end{aligned}$$

At convergence, the iteration provides factors S , R of controllability and observability Gramians of the system as follows:

$$S := \frac{1}{\sqrt{2}} \lim_{j \rightarrow \infty} S_j E^{-T}, \quad R := \frac{1}{\sqrt{2}} \lim_{j \rightarrow \infty} R_j.$$

A convergence criterion can easily be derived from

$$A_\infty := \lim_{j \rightarrow \infty} A_j = E.$$

Thus, following the same ideas as in the inverse-free iteration, in order to avoid convergence stagnation problems we can test whether $\|A_j - E\|_1 < n\sqrt{\varepsilon}\|A_j\|_1$ and perform two more iterations once this criterion is satisfied.

For further details on the numerical accuracy, converge rate, and (parallel) implementation of this iteration, see [7, 6].

4 Implementation and Parallelization Details

The iterative schemes that we have described in the previous sections are basically composed of traditional matrix computations such as matrix factorizations,

solution of triangular linear systems, matrix inversion, and matrix products. All these operations can be efficiently performed employing parallel linear algebra libraries for distributed memory computers [10, 37]. The use of these libraries enhances the reliability and improves the portability of the model reduction algorithms. The performance will depend on the efficiencies of the underlying serial and parallel computational linear algebra kernels and the communication routines.

Here we employ the parallel kernels in the ScaLAPACK library [10]. This is a freely available library that implements parallel versions of many of the routines in LAPACK [1], using the message-passing paradigm. ScaLAPACK employs the PBLAS (a parallel version of the serial BLAS) for computation and BLACS for communication. The BLACS can be ported to any (serial and) parallel architecture with an implementation of the MPI (our case) or the PVM libraries [18, 22].

The RRQR factorizations in our codes are computed by means of the traditional QR factorization with column pivoting [20]. In order to improve the performance of our parallel model reduction routines we have designed and implemented two specialized parallel kernels that outperform parallel kernels in ScaLAPACK with an analogous purpose: the QR factorization with partial pivoting is computed in our codes using a parallel BLAS-3 version instead of the traditional BLAS-2 approach [29]. Also, our matrix inversion routine is based on a Gauss-Jordan elimination procedure [28] that presents a better load balance on distributed memory architectures than traditional inversion algorithms based on Gaussian elimination (LU factorization).

5 Experimental Results

All the experiments presented in this section were performed on a cluster of 16 nodes using IEEE double-precision floating-point arithmetic ($\varepsilon \approx 2.2204 \times 10^{-16}$). Each node consists of an Intel Pentium Xeon processor at 2.4 GHz with 1 GByte of RAM. We employ a BLAS library specially tuned for this processor that achieves around 3.0 Gflops (millions of flops per second) for the matrix product (routine DGEMM) [21]. The nodes are connected via a *Myrinet* multistage network and the MPI communication library is specially developed and tuned for this network. The performance of the interconnection network was measured by a simple loop-back message transfer resulting in a latency of 18 μ sec. and a bandwidth of 1.4 Gbit/sec. We made use of the LAPACK, BLAS, and ScaLAPACK libraries whenever possible.

As in our experiments both SR and BFSR BT algorithms obtained closely similar results we only report data for the first algorithm.

5.1 Numerical aspects

In order to evaluate the numerical behavior of our model reduction algorithms we employed a linear descriptor system with $n = 500$ states, $m = 5$ inputs, and

$p=10$ outputs. This system was generated so that the ratio between finite/infinite poles was $470/30$ and $\text{ind}(A, E) = 1$.

Our first experiment is designed to evaluate the accuracy and performance of the inverse-free iteration as a spectral division tool. For this purpose compare the residuals (backward errors for spectral division) of $\|[A_{21}, E_{21}]\|_F$ and $\|E_{22}\|_F$ in (4) obtained both with the inverse-free iteration and the QZ algorithm. The latter is proved to be a numerically stable algorithm [20] and therefore should provide a measure of how accurate our inverse-free approach is. The results in Table 1 demonstrate that, at least for this example, the inverse-free iteration provides orthogonal bases Q and Z that are as “accurate” as those resulting from the QZ algorithm.

	Disk function	QZ alg.
$\ [A_{21}, E_{21}]\ _F$	4.2×10^{-13}	1.0×10^{-12}
$\ E_{22}\ _F$	1.1×10^{-15}	1.1×10^{-15}

Table 1. Accuracy of the bases for the deflating subspaces.

The convergence of the inverse-free iteration for block-triangularization of this example is shown in the left plot in Fig. 1.

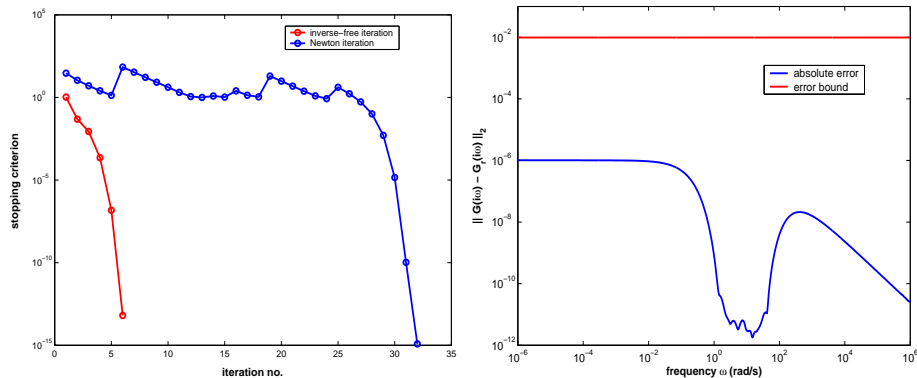


Fig. 1. Convergence rate for the iterative schemes (left plot) and Bode plot (magnitude) of error system (right plot).

We then applied BT to the realization associated with the finite part of $A(A, E)$ to obtain a reduced-order realization so that

$$\|G_0 - \hat{G}_0\|_\infty \leq 0.01.$$

As a result, the order of the system was reduced from $n=470$ to $r=60$ and

$$\|\Delta_a\|_\infty = \|G_0 - \hat{G}_0\|_\infty \leq 2 \sum_{j=61}^{470} \sigma_j = 9.796 \times 10^{-3}.$$

In the left-hand plot in Figure 1 we also report the convergence rate of our Newton iteration for the solution of the coupled generalized Lyapunov equations. We report the absolute error between the original and the reduced-order TFM in the right-hand plot of this figure. The results show that the reduced-order model is very accurate and the error bound (12) is fairly pessimistic in this example.

5.2 Parallel performance

In this subsection we report the performance of the parallel routine for the decoupling of the infinite poles of the TFM, `pggdec`, and the reduction of a linear descriptor system with no infinite poles, `pgab09ax`. In order to mimic a real case, we employ a stable linear descriptor system of order n with $m = n/10$ inputs, $p = n/10$ outputs, $n/10$ infinite poles, and $\text{ind}(A, E) = 1$.

Our first experiment reports the execution time of the parallel routines on a system of order $n = 1800$. Here, once the infinite poles have been decoupled, routine `pgab09ax` is applied on a system of order $n - n/10 = 1620$. These are about the largest sizes we could evaluate on a single node of our cluster, considering the number of data matrices involved, the amount of workspace necessary for computations, and the size of the RAM per node. The left-hand plot in Figure 2 reports the execution time of the parallel routine using $n_p=1, 2, 4, 6, 8,$ and 10 nodes. The execution of the parallel algorithm on a single node is likely to require a higher time than that of a serial implementation of the algorithm (using, e.g., LAPACK and BLAS); however, at least for such large scale problems, we expect this overhead to be negligible compared to the overall execution time.

The figure shows reasonable speed-ups when a reduced number of processors is employed. Thus, when $n_p=4$, speed-ups of 3.25 and 4.25 are obtained for routines `pdggdec` and `pgab09ax`, respectively. The super speed-up in the latter case is due to a better use of the memory in the parallel algorithm. However, as expected the efficiency decreases as n_p gets larger (as the system dimension is fixed, the problem size per node is reduced) so that using more than a few processors does not achieve a significant reduction in the execution time for such a small problem.

We next evaluate the scalability of the parallel routine when the problem size per node is constant. For that purpose, we fix the problem dimensions to $n/\sqrt{n_p} = 1800$ for routine `pggdec`, and $n/\sqrt{n_p} = 1620$ for routine `pgab09ax`. The number of inputs/outputs in both cases is $m = p = n/10$. The right-hand plot in Figure 2 shows the GigaFlop rate per node of the parallel routine. These results demonstrate the scalability of our parallel kernels, as there is only a minor decrease in the performance of the algorithms when n_p is increased while the problem dimension per node remains fixed.

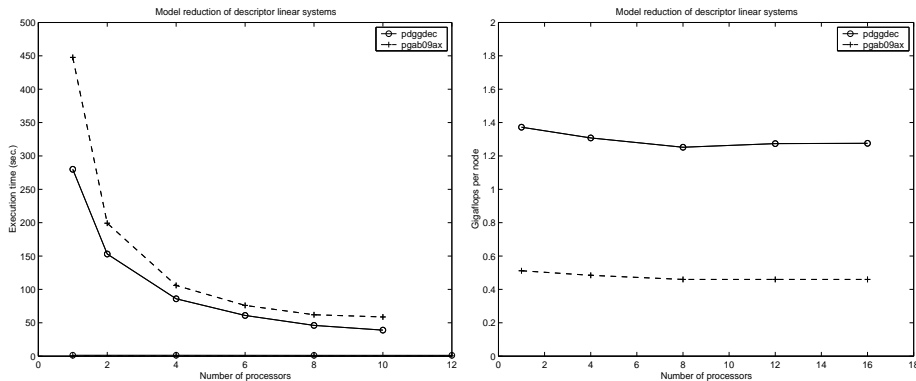


Fig. 2. Performance of the parallel routines for model reduction of linear descriptor systems.

6 Concluding Remarks

We have described the design and use of parallel algorithms for model reduction of large-scale linear descriptor systems. The spectral projection methods considered here are composed of highly efficient iterative schemes for the disk and sign functions and allow one to obtain low-order approximations of very large dense systems.

Experimental results report a high performance and scalability of the parallel algorithms on a cluster of Intel Pentium Xeon processors.

References

1. E. Anderson, Z. Bai, C. Bischof, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, Philadelphia, PA, third edition, 1999.
2. A.C. Antoulas. *Lectures on the Approximation of Large-Scale Dynamical Systems*. SIAM Publications, Philadelphia, PA, to appear.
3. 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.
4. Z. Bai, J. Demmel, and M. Gu. An inverse free parallel spectral divide and conquer algorithm for nonsymmetric eigenproblems. *Numer. Math.*, 76(3):279–308, 1997.
5. P. Benner. Spectral projection methods for model reduction of descriptor systems. Preprint, DFG Research Center "Mathematics for key technologies" (FZT 86), Berlin, Germany, 2003.
6. P. Benner, J.M. Claver, and E.S. Quintana-Ortí. Efficient solution of coupled Lyapunov equations via matrix sign function iteration. In A. Dourado and et al., editors, *Proc. 3rd Portuguese Conf. on Automatic Control CONTROL'98*, Coimbra, pages 205–210, 1998.

7. P. Benner and E.S. Quintana-Ortí. Solving stable generalized Lyapunov equations with the matrix sign function. *Numer. Algorithms*, 20(1):75–100, 1999.
8. P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Balanced truncation model reduction of large-scale dense systems on parallel computers. *Math. Comput. Model. Dyn. Syst.*, 6(4):383–405, 2000.
9. P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. State-space truncation methods for parallel model reduction of large-scale systems. *Parallel Comput.*, 2004. To appear.
10. 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.
11. J. Cheng, G. Ianculescu, C.S. Kenney, A.J. Laub, and P. M. Papadopoulos. Control-structure interaction for space station solar dynamic power module. *IEEE Control Systems*, pages 4–13, 1992.
12. P.Y. Chu, B. Wie, B. Gretz, and C. Plescia. Approach to large space structure control system design using traditional tools. *AIAA J. Guidance, Control, and Dynamics*, 13:874–880, 1990.
13. M.A. Epton. Methods for the solution of $AXD - BXC = E$ and its application in the numerical solution of implicit ordinary differential equations. *BIT*, 20:341–345, 1980.
14. L. Fortuna, G. Nummari, and A. Gallo. *Model Order Reduction Techniques with Applications in Electrical Engineering*. Springer-Verlag, 1992.
15. R. Freund. Reduced-order modeling techniques based on Krylov subspaces and their use in circuit simulation. In B.N. Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1, chapter 9, pages 435–498. Birkhäuser, Boston, MA, 1999.
16. J.D. Gardiner and A.J. Laub. A generalization of the matrix-sign-function solution for algebraic Riccati equations. *Internat. J. Control*, 44:823–832, 1986.
17. J.D. Gardiner, M.R. Wette, A.J. Laub, J.J. Amato, and C.B. Moler. Algorithm 705: A Fortran-77 software package for solving the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Software*, 18:232–238, 1992.
18. A. Geist, A. Beguelin, J. Dongarra, W. Jiang, B. Manckek, and V. Sunderam. *PVM: Parallel Virtual Machine - A Users Guide and Tutorial for Network Parallel Computing*. MIT Press, Cambridge, MA, 1994.
19. K. Glover. All optimal Hankel-norm approximations of linear multivariable systems and their L^∞ norms. *Internat. J. Control*, 39:1115–1193, 1984.
20. G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, third edition, 1996.
21. K. Goto and R. van de Geijn. On reducing TLB misses in matrix multiplication. FLAME Working Note 9, Department of Computer Sciences, The University of Texas at Austin, <http://www.cs.utexas.edu/users/flame>, 2002.
22. W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, Cambridge, MA, 1994.
23. A.N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.
24. B.C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Trans. Automat. Control*, AC-26:17–32, 1981.
25. G. Obinata and B.D.O. Anderson. *Model Reduction for Control System Design*. Communications and Control Engineering Series. Springer-Verlag, London, UK, 2001.

26. C.R. Paul. *Analysis of Multiconductor Transmission Lines*. Wiley-Interscience, Singapur, 1994.
27. E.S. Quintana-Ortí. *Algoritmos Paralelos Para Resolver Ecuaciones Matriciales de Riccati en Problemas de Control*. PhD thesis, Universidad Politécnica de Valencia, 1996.
28. E.S. Quintana-Ortí, G. Quintana-Ortí, X. Sun, and R.A. van de Geijn. A note on parallel matrix inversion. *SIAM J. Sci. Comput.*, 22:1762–1771, 2001.
29. G. Quintana-Ortí, X. Sun, and C.H. Bischof. A BLAS-3 version of the QR factorization with column pivoting. *SIAM J. Sci. Comput.*, 19:1486–1494, 1998.
30. 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).
31. M.G. Safonov and R.Y. Chiang. A Schur method for balanced-truncation model reduction. *IEEE Trans. Automat. Control*, AC-34:729–733, 1989.
32. T. Stykel. Model reduction of descriptor systems. Technical Report 720-2001, Institut fr Mathematik, TU Berlin, D-10263 Berlin, Germany, 2001.
33. T. Stykel. *Analysis and Numerical Solution of Generalized Lyapunov Equations*. Dissertation, TU Berlin, 2002.
34. T. Stykel. Balanced truncation model reduction for semidiscretized stokes equation. Technical Report 04-2003, Institut fr Mathematik, TU Berlin, D-10263 Berlin, Germany, 2003.
35. X. Sun and E.S. Quintana-Ortí. Spectral division methods for block generalized Schur decompositions. PRISM Working Note #32, 1996. Available from <http://www-c.mcs.anl.gov/Projects/PRISM>; To appear in *Math. Comp.*
36. 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.
37. R.A. van de Geijn. *Using PLAPACK: Parallel Linear Algebra Package*. MIT Press, Cambridge, MA, 1997.
38. A. Varga. Efficient minimal realization procedure based on balancing. In *Prepr. of the IMACS Symp. on Modelling and Control of Technological Systems*, volume 2, pages 42–47, 1991.