Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Parallel Algorithms for Balanced Truncation of Large-Scale Unstable Systems

Sergio Barrachina, Peter Benner, Enrique S. Quintana-Ortí, and Gregorio Quintana-Ortí

***Abstract*— We discuss and compare two approaches for model reduction of large-scale unstable systems on parallel computers. The first method proceeds by computing the additive decomposition of the transfer function via a block diagonalization, followed by a reduction of the stable part of the system using techniques based on state-space truncation. The second method employs a representation of the controllability and observability Gramians of an unstable systems in terms of the Gramians of the stabilized system where the particular stabilization is obtained via the solution of dual algebraic Bernoulli equations. Based on these Gramians, balanced truncation is then applied in the usual manner. All core computational steps in these methods can be efficiently solved on parallel computers by means of diverse variants of the Newton iteration for the sign function. Numerical experiments on a cluster of Intel Xeon processors show the numerical and parallel performances of these methods.**

I. INTRODUCTION

Consider the transfer function matrix (TFM)

$$G(s) = C(sI - A)^{-1}B + D, \qquad (1)$$

with $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$, $D \in \mathbb{R}^{p\times m}$, and I_n the identity matrix of order n; and the associated, not necessarily minimal, realization of a linear time-invariant (LTI) system,

$$\begin{array}{rcll} \dot{x}(t) & = & Ax(t) + Bu(t), & t > 0, \quad x(0) = x^0, \\ y(t) & = & Cx(t) + Du(t), & t \geq 0, \end{array} \qquad (2)$$

with $x^0 \in \mathbb{R}^n$ being the initial state of the system. Here, the number of state variables n is also known as the *order* or the state-space dimension of the system. For simplicity we assume hereafter that the spectrum of A is dichotomic with respect to the imaginary axis, i.e., $Re(\lambda) \neq 0$ for all eigenvalues λ of A. The case with eigenvalues on the imaginary axis could be treated as well with the methods described in this paper by employing a spectral shift, but this would add some distracting technicalities.

Model reduction means to find an LTI system,

$$\begin{array}{rcll} \dot{\hat{x}}(t) & = & \hat{A}\hat{x}(t) + \hat{B}u(t), & t > 0, \\ \hat{y}(t) & = & \hat{C}\hat{x}(t) + \hat{D}u(t), & t \geq 0, \end{array} \qquad (3)$$

of reduced-order r, such that $r \ll n$, and the TFM

$$\hat{G}(s) = \hat{C}(sI - \hat{A})^{-1}\hat{B} + \hat{D}$$

"approximates" $G(s)$. The reduced-order system has the potential to replace the original one in many subsequent computations.

Model reduction of large-scale LTI systems is necessary in control of multibody (mechanical) systems, manipulation of fluid flow, circuit simulation, VLSI chip design, in particular when modeling the interconnect via RLC networks, simulation of MEMS and NEMS (micro- and nano-electro-mechanical systems), weather forecast, circuit simulation, VLSI design, and air quality simulation, among others. (See, e.g., [1], [2] and the references therein.) State-space dimensions n of order 10^2–10^4 are common in these applications. Numerically reliable methods for model reduction, as those described later, require $\mathcal{O}(n^3)$ flops (floating-point arithmetic operations) and storage for $\mathcal{O}(n^2)$ real numbers. Thus, there are important benefits to be obtained from the application of parallel computing techniques.

Traditional (absolute error) methods for model reduction based on state-space truncation or projected dynamics include balanced truncation (BT) methods, singular perturbation approximation (SPA) methods, and optimal Hankel-norm approximation (HNA) methods [1], [3]. None of these methods can be applied directly to *unstable systems* (i.e., TFMs with poles in the right half plane) though this type of systems occurs quite often, in particular if stabilization is the computational task to solve. Unstable systems also appear in controller reduction: controllers are often themselves unstable LTI systems — thus the task of controller reduction leads to model reduction of this class of systems [4]. Usually, unstable poles cannot be neglected when modeling the dynamics of the system, and therefore should be preserved in the reduced-order system in some sense.

Here we describe parallel implementations of two different approaches for model reduction of unstable systems. The first method requires an additive decomposition of the TFM and consists of two stages: first, $G(s)$ is decomposed as $G(s) = G^-(s) + G^+(s)$, where $G^-(s)$ is stable and $G^+(s)$ is unstable. Next, any appropriate model reduction method for stable systems (e.g., BT, SPA, or HNA) is applied to G^- in order to obtain a reduced-order system $\hat{G}^-$; the reduced-order system is then synthesized by

$$\hat{G}(s) = \hat{G}^-(s) + G^+(s). \qquad (4)$$

This approach is advantageous in controller reduction where it is needed to guarantee the stabilization property of the

S. Barrachina, E.S. Quintana-Ortí, and G. Quintana-Ortí were supported by the CICYT project No. TIC2002-004400-C03-01 and FEDER, and project No. P1B-2004-6 of the *Fundación Caixa-Castellón/Bancaixa and UJI*. P. Benner was supported DFG Sonderforschungsbereich 393 *Parallele Numerische Simulation für Physik und Kontinuumsmechanik* at TU Chemnitz.

S. Barrachina, E.S. Quintana-Ortí, and G. Quintana-Ortí are with the Departamento de Ingeniería y Ciencia de los Computadores, Universidad Jaume I, 12.071-Castellón, Spain. {barrachi,quintana,gquintan}@icc.uji.es

P. Benner is with the Fakultät für Mathematik, Technische Universität Chemnitz, D-09107 Chemnitz, Germany. benner@mathematik.tu-chemnitz.de

0-7803-9568-9/05/$20.00 ©2005 IEEE

controller. Of course, if the number of unstable poles is dominating, the potential for reducing the model is limited. However, in many applications, in particular those coming from semi-discretization of parabolic or hyperbolic partial differential equations, the numbers of unstable poles is very low compared to the number of stable poles. On the other hand, if the number of unstable poles is not as low as in these applications, one would certainly also like to reduce the unstable part of the system. Several approaches to generalize the concept of balanced truncation to unstable systems can be found in the literature. We will focus here on a method described in [5]; for a discussion of earlier attempts at extending balanced truncation to unstable systems we refer the reader also to [5]. The method we will use is based on a representation of the Gramians of the unstable system which is always well-defined whenever A has no eigenvalues on the imaginary axis. These Gramians are computable in a two step approach where, first, a particular stabilization of the system is performed. In the second step, the Gramians are computed as the controllability and observability Gramians of the stabilized system in the usual way, from the solutions of two dual Lyapunov equations. The stabilization procedure requires the solution of dual algebraic Bernoulli equations, i.e., homogeneous algebraic Riccati equations. Other methods for model reduction of unstable systems based on coprime factorization of the transfer function (see, e.g., [4] and the references therein) will not be discussed in this paper; work on comparing these approaches for large-scale systems is in progress.

The outline of the paper is as follows: In Sections II and III we describe the approaches for model reduction of unstable systems. Next, in Section IV, we review the basics of the sign function and how this matrix function provides a solution to all major computational steps involved in these methods. The iterative schemes for the matrix sign function are specially appealing in that they are easy to parallelize using existing parallel libraries for dense linear algebra. In particular, a few details of a parallelization using ScaLAPACK [6] and the contents of our parallel model reduction library PLiCMR are discussed in Section V. The examples in Section VI report the numerical accuracy and parallel efficiency of both approaches. Finally, some concluding remarks follow in Section VII.

Throughout the paper we use $\Lambda(M)$ to refer to the spectrum (set of eigenvalues) of a matrix M; $\mathbb{C}^-$, $\mathbb{C}^+$ denote, respectively, the open left and right half planes; and $\jmath = \sqrt{-1}$, so that $\jmath\mathbb{R}$ stands for the imaginary axis. Also, when the order is clear from the context, we drop the subindex denoting the order of the identity matrix, as in I.

II. MODEL REDUCTION VIA ADDITIVE DECOMPOSITION

In this section we describe the two stages (and major computations) which compose the first approach considered here for model reduction of unstable systems: additive decomposition of the TFM, followed by BT model reduction of the stable part.

A. Additive Decomposition of a TFM

Following [7], we perform an additive decomposition of $G(s)$ by computing a state-space transformation $T \in \mathbb{R}^{n\times n}$ such that

$$(\tilde{A},\tilde{B},\tilde{C},\tilde{D}) \;:=\; (T^{-1}AT, T^{-1}B, CT, D), \qquad (5)$$

and

$$\tilde{A} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix},\; \tilde{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},\; \tilde{C} = [\, C_1 \;\; C_2], \quad (6)$$

with $\Lambda(A_{11}) = \Lambda(A) \cap \mathbb{C}^-$, $A_{11} \in \mathbb{R}^{k\times k}$, $\Lambda(A_{22}) = \Lambda(A) \cap \mathbb{C}^+$, $A_{22} \in \mathbb{R}^{n-k\times n-k}$. Thus,

$$\begin{array}{rcl} G(s) & = & C(sI-A)^{-1}B + D = \tilde{C}(sI-\tilde{A})^{-1}\tilde{B} + \tilde{D} \\ & = & \big\{C_1(sI-A_{11})^{-1}B_1 + D\big\} \\ & & \quad + \big\{C_2(sI-A_{22})^{-1}B_2\big\} \\ & =: & G^-(s) + G^+(s) \end{array}$$

is a decomposition of $G(s)$ into a stable TFM, $G^-(s)$, and an unstable one, $G^+(s)$.

B. BT Model Reduction of a stable TFM

Let $(A,B,C,D) := (A_{11},B_1,C_1,D)$ be the realization associated with the stable TFM $G^-(s)$ in (4). BT methods are strongly related with the controllability Gramian W_c and the observability Gramian W_o of the LTI system, which are given by the solutions of the dual *Lyapunov equations*

$$\begin{array}{rcl} AW_c + W_cA^T + BB^T & = & 0, \\ A^TW_o + W_oA + C^TC & = & 0. \end{array} \qquad (7)$$

As A is stable, W_c and W_o are positive semidefinite, and therefore there exist factorizations $W_c = S^TS$ and $W_o = R^TR$.

Consider now the singular value decomposition (SVD)

$$SR^T = [U_1\; U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix}, \qquad (8)$$

where the matrices are partitioned at a given dimension r such that $\Sigma_1 = \operatorname{diag}(\sigma_1,\ldots,\sigma_r)$, $\Sigma_2 = \operatorname{diag}(\sigma_{r+1},\ldots,\sigma_n)$, $\sigma_j \geq \sigma_{j+1} \geq 0$ for all j, and $\sigma_r > \sigma_{r+1}$.

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

$$\begin{array}{rclcrcl} \hat{A} & = & T_lAT_r, & & \hat{B} & = & T_lB, \\ \hat{C} & = & CT_r, & & \hat{D} & = & D, \end{array} \qquad (9)$$

using the projection implied by the matrices

$$T_l = \Sigma_1^{-1/2}V_1^TR \quad \text{and} \quad T_r = S^TU_1\Sigma_1^{-1/2}. \qquad (10)$$

SR BT algorithms provide a realization $\hat{G}$ which satisfies the theoretical error bound

$$\|\Delta_a\|_\infty = \|G-\hat{G}\|_\infty \leq 2\sum_{j=r+1}^{n}\sigma_j, \qquad (11)$$

where $\|G\|_\infty$ denotes the $\mathcal{L}_\infty$- or $\mathcal{H}_\infty$-norm of a rational matrix-valued function. This allows an adaptive choice of the state-space dimension r of the reduced-order model once the HSVs are known.

III. MODEL REDUCTION VIA BALANCED REALIZATION

In [5], the following definition for the controllability and observability Gramians of unstable systems is given:

$$\begin{array}{rcl} \hat{W}_c & := & \int_{-\infty}^{\infty}(\jmath\omega - A)^{-1}BB^T(\jmath\omega - A)^{-H}\,d\omega, \\ \hat{W}_o & := & \int_{-\infty}^{\infty}(\jmath\omega - A)^{-H}C^TC(\jmath\omega - A)^{-1}\,d\omega, \end{array} \tag{12}$$

where M^H denotes the complex conjugate transpose of the matrix M. It is shown in [5] that this definition is consistent with the usual one in the stable case and that most of the properties of the Gramians known in the stable case carry over to the unstable case. For instance, the LTI system (3) is controllable and observable if and only if $\hat{W}_c$ and $\hat{W}_o$, respectively, are positive definite. Based on this fact a balanced realization and balanced truncated reduced-order model are computable in complete analogy to the stable case. Hence, if $\hat{W}_c$ and $\hat{W}_o$ are known, a model reduction procedure based on (8)–(10) can be applied to (3). It is also shown that the error bound (11) holds for the reduced-order model computed in this way.

In order to compute $\hat{W}_c, \hat{W}_o$, in a first step we need to solve the dual algebraic Bernoulli equations (ABEs)

$$A^TX + XA - XBB^TX \;=\; 0, \tag{13}$$

$$AY + YA^T - YC^TCY \;=\; 0 \tag{14}$$

for $X, Y \in \mathbb{R}^{n\times n}$. The solutions of these ABEs — if they exist — are usually not unique. Here we are interested in the "stabilizing" solutions X^- and Y^- such that $A_c := A - BB^TX^-$ and $A_o := A - Y^-C^TC$ are both stable.

Due to its nature as an algebraic Riccati equation (ARE), it is not surprising that the solution theory of the ABE can be derived from that of the ARE. Using Theorem 7.5.1 and Section 8.3 of [8], together with the observation that $X = 0$, $Y = 0$ are symmetric solutions of the ABEs (13)–(14), we obtain the following result.

Proposition 1 (Extremal ABE solutions): Consider the ABE (13) with (A, B) controllable. Then there exist symmetric solutions $X^+ \geq 0$, $X^- \leq 0$ of (13) with $X^- \leq X \leq X^+$ for all solutions X of (13).

Moreover, X^- is the unique solution satisfying $\Lambda\left(A - BB^TX^-\right) \subset \mathbb{C}^+ \cup \jmath\mathbb{R}$ and X^+ is the unique solution satisfying $\Lambda\left(A - BB^TX^+\right) \subset \mathbb{C}^- \cup \jmath\mathbb{R}$. If $\Lambda\left(A\right) \cap \jmath\mathbb{R} = \emptyset$, then X^- is the unique anti-stabilizing solution and X^+ is the unique stabilizing solution of the ABE.

An analogous result holds for the ABE (14) provided (A, C) is observable (i.e., (A^T, C^T) is controllable).

It can then be shown (see [5]) that with the stabilizing solutions of the ABEs (13) and (14), the Gramians in (12) are given by the solutions of the dual stable Lyapunov equations

$$\begin{array}{rcl} A_c\hat{W}_c + \hat{W}_cA_c^T + BB^T & = & 0, \\ A_o^T\hat{W}_o + \hat{W}_oA_o + C^TC & = & 0. \end{array} \tag{15}$$

Here, A_c and A_o are again stable and therefore there exist $\hat{S}$ and $\hat{R}$ such that $\hat{W}_c = \hat{S}^T\hat{S}$ and $\hat{W}_o = \hat{R}^T\hat{R}$.

Summarizing, a computational procedure for balanced truncation of unstable systems requires first to solve the ABEs (13) and (14), then to compute $\hat{S}, \hat{R}$ via (15) so that the reduced-order model is then obtained using (8), (9), and (10) with S and R replaced by $\hat{S}$ and $\hat{R}$.

Remark 1: The Gramians $\hat{W}_c, \hat{W}_o$ in (12) can also be computed in an alternative way using the additive decomposition of G described in Section II and then computing the Gramians of the stable systems given by (A_{11}, B_1, C_1, D) and $(-A_{22}, B_2, C_2, 0)$ separately [5]. This is computationally more complex than the approach based on the ABEs and we therefore do not pursue this any further.

IV. THE MATRIX SIGN FUNCTION AND APPLICATIONS

In this section we briefly summarize the sign function and the classical Newton iteration for its computation. We also describe how to employ the sign function in order to decompose a TFM, and we give a survey of specialized variants of the Newton iteration which can be employed to solve certain matrix equations.

A. Background on the Sign Function

Consider a matrix $H \in \mathbb{R}^{s\times s}$ with $\Lambda\left(H\right) \cap \jmath\mathbb{R} = \emptyset$, and let $H = S\left[\begin{smallmatrix} J^- & 0 \\ 0 & J^+ \end{smallmatrix}\right]S^{-1}$ be its Jordan decomposition, where the Jordan blocks in $J^- \in \mathbb{R}^{t\times t}$ and $J^+ \in \mathbb{R}^{(s-t)\times(s-t)}$ contain, respectively, the eigenvalues of H in the open left and right half of the complex plane. The *matrix sign function* of H is defined as $\operatorname{sign}\left(H\right) := S\left[\begin{smallmatrix} -I_t & 0 \\ 0 & I_{s-t} \end{smallmatrix}\right]S^{-1}$. For an overview of many definitions of the sign function, see [9].

Applying Newton's root-finding iteration to $H^2 = I_s$, where the starting point is chosen as H, we obtain the Newton iteration for the matrix sign function:

$$H_0 \leftarrow H, \quad H_{j+1} \leftarrow \frac{1}{2}(H_j + H_j^{-1}), \quad j = 0, 1, 2, \ldots. \tag{16}$$

Under the given assumptions, the sequence $\{H_j\}_{j=0}^{\infty}$ converges to $\operatorname{sign}\left(H\right) = \lim_{j\to\infty} H_j$ [10] with a locally quadratic convergence rate. As the initial convergence may be slow, acceleration techniques are often used; e.g., *determinantal scaling*:

$$H_j \leftarrow c_jH_j, \quad c_j = |\det\left(H_j\right)|^{-\frac{1}{s}}.$$

For a discussion of several scaling strategies, see [9].

We next illustrate the use of the sign function in order to compute a pair of similarity transformations defined by $Q \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{n\times n}$ which, respectively, separate the stable/anti-stable eigenspectrum of A, and block diagonalize the resulting matrix. As a result, $T := QV$ is the state-space transformation to be used in (5) for the additive decomposition of (A, B, C, D).

B. Spectral division of matrices

The sign function is an efficient numerical tool for spectral division. Specifically, in order to separate the eigenspectrum of A along the imaginary axis, we only need to compute a rank-revealing QR (RRQR) factorization of $I_n - \operatorname{sign}\left(A\right)$ as

$$I_n - \operatorname{sign}\left(A\right) = QRP, \quad R = \left[\begin{array}{cc} R_{11} & R_{12} \\ 0 & 0 \end{array}\right],$$

where P is a permutation matrix, $\mathrm{rank}\,(I - \mathrm{sign}\,(A)) = \mathrm{rank}\,(R) = k$, and $R_{11} \in \mathbb{R}^{k \times k}$ is upper triangular. (Note that k is also the number of stable eigenvalues in A.) The similarity transformation defined by Q then satisfies that

$$\bar{A} \;:=\; Q^T A Q \;=\; \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}, \tag{17}$$

where $\Lambda\,(A_{11}) = \Lambda\,(A) \cap \mathbb{C}^-$, $A_{11} \in \mathbb{R}^{k \times k}$, $\Lambda\,(A_{22}) = \Lambda\,(A) \cap \mathbb{C}^+$, $A_{22} \in \mathbb{R}^{n-k \times n-k}$; in other words, Q separates the stable and the anti-stable parts of $\Lambda\,(A)$.

C. Block diagonalization

Consider now the block triangular matrix $\bar{A}$ from (17). The goal here is to find a similarity transformation that sets A_{12} to zero. Thus, we need to find V such that

$$\begin{aligned} \hat{A} \;&:=\; V^{-1}\bar{A}V = V^{-1}Q^T AQV \\ &=\; \begin{bmatrix} I_k & -U \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix} \begin{bmatrix} I_k & U \\ 0 & I_{n-k} \end{bmatrix} \\ &=\; \begin{bmatrix} A_{11} & 0 \\ 0 & A_{22} \end{bmatrix}. \end{aligned}$$

A little manipulation shows that $U \in \mathbb{R}^{k \times n-k}$ is the solution of the *Sylvester equation*

$$A_{11}U - UA_{22} + A_{12} = 0. \tag{18}$$

Also, as $\Lambda\,(A_{11}) \cap \Lambda\,(A_{22}) = \emptyset$, equation (18) has a unique solution [11] which can be computed with the method described in the following paragraph.

D. Solving Sylvester and Lyapunov Equations

The Sylvester equation (18) can be solved using a sign function-based procedure. Given that A_{11} and $-A_{22}$ are stable matrices, this iterative procedure, already derived in [10], proceeds as follows:

$$\begin{array}{ll} E_0 \leftarrow A_{11}, & E_{j+1} \leftarrow \frac{1}{2}\left(E_j + E_j^{-1}\right), \\ F_0 \leftarrow A_{22}, & F_{j+1} \leftarrow \frac{1}{2}\left(F_j + F_j^{-1}\right), \\ W_0 \leftarrow A_{12}, & W_{j+1} \leftarrow \frac{1}{2}\left(W_j + E_j^{-1}W_jF_j^{-1}\right), \\ & \qquad\qquad j = 0, 1, 2, \ldots. \end{array} \tag{19}$$

At convergence, $\lim_{j\to\infty} E_j = -I_k$, $\lim_{j\to\infty} F_j = I_{n-k}$, and the solution $U = \frac{1}{2}\lim_{j\to\infty} W_j$.

For an efficient implementation of this iteration on modern computer architectures and numerical experiments reporting efficiency and accuracy, see [12].

The Lyapunov equations in (7) and (15) are special cases of the Sylvester equation. Therefore, these equations can be solved using a simplified variant of the iterative scheme in (19). Now, consider, e.g., the first Lyapunov equation in (7) and its factorized solution $W_c = S^T S$. In model reduction, S is often a numerically rank deficient matrix. In such a case, an efficient variant of the Newton iteration proposed in [13] delivers a full-rank factor S_f, so that $S_f S_f^T \approx S^T S = W_c$. Specifically, this variant is formulated as follows:

$$\begin{array}{ll} A_0 \leftarrow A, & A_{j+1} \leftarrow \frac{1}{\sqrt{2}}\left(A_j + A_j^{-1}\right), \\ S_0 \leftarrow B, & S_{j+1} \leftarrow R_s \Pi_s, \\ & \qquad\qquad j = 0, 1, 2, \ldots, \end{array} \tag{20}$$

where R_s and Π_s are obtained at each iteration from the RRQR decomposition:

$$\frac{1}{\sqrt{2}}\left[\, S_j, \;\; A_j^{-1}S_j \,\right]^T = Q_s \begin{bmatrix} R_s \\ 0 \end{bmatrix} \Pi_s.$$

On convergence, $S_f = \frac{1}{\sqrt{2}}\lim_{j\to\infty} S_j$ can reliably substitute S. This iteration is in general cheaper to compute than the one for the Cholesky factor. Besides, the use of a full-rank factor allows significant computational savings in the subsequent computations (8)–(10) of model reduction.

E. Solving ABEs with the Sign Function

ABEs of the form (13)–(14) can be solved using the following coupled iteration

$$\begin{array}{ll} A_0 \leftarrow A, & A_{j+1} \leftarrow \frac{1}{2}\left(A_j + A_j^{-1}\right), \\ F_0 \leftarrow BB^T, & F_{j+1} \leftarrow \frac{1}{2}\left(F_j + A_j^{-1}F_jA_j^{-T}\right), \\ G_0 \leftarrow C^TC, & G_{j+1} \leftarrow \frac{1}{2}\left(G_j + A_j^{-T}G_jA_j^{-1}\right), \\ & \qquad\qquad j = 0, 1, 2, \ldots. \end{array} \tag{21}$$

The solutions are then obtained from the full-rank linear least-squares problems (see [14], [10], [15])

$$\begin{aligned} \begin{bmatrix} F_\infty \\ I_n - A_\infty^T \end{bmatrix} X \;&=\; \begin{bmatrix} A_\infty + I_n \\ 0_n \end{bmatrix}, \\ \begin{bmatrix} G_\infty \\ I_n - A_\infty \end{bmatrix} Y \;&=\; \begin{bmatrix} A_\infty^T + I_n \\ 0_n \end{bmatrix}, \end{aligned}$$

where $M_\infty = \lim_{j\to\infty} M_j$, $M \in \{A, F, G\}$.

V. IMPLEMENTATION DETAILS

The methods for model reduction basically require matrix operations such as the solution of linear systems and matrix equations (Sylvester, Lyapunov and ABE), and the computation of matrix products and matrix decompositions (QR, SVD, etc.). The variants of the Newton iteration described earlier only require operations such as matrix products and matrix inversion. All these operations are basic dense matrix algebra kernels parallelized in ScaLAPACK and PBLAS. Using these libraries, we have developed parallel model reduction routines, integrated into the PLiCMR library (`http://spine.act.uji.es/~plicmr`).

The kernels in the PLiCMR library allow model reduction of stable and unstable large-scale systems, with state-space order up to $\mathcal{O}(10^4)$, on parallel distributed-memory platforms. Currently, the library includes the following computational routines for model reduction of stable systems:

Absolute error methods:

`pab09ax` (BT), `pab09bx` (SPA), `pab09cx` (HNA).

Relative error methods:

`pab09hx` (Balanced Stochastic Truncation),

`pab09px` (Balanced Positive Real Truncation).

For unstable systems, the library contains two more computational routines:

- Additive decomposition: `pab09ex`. For model reduction apply any of the abovementioned routines to the stable part of the system.
- Balanced truncation: `pab09fx`. This implements the method described in Section III.

VI. Numerical Experiments

All the experiments presented in this section were performed on a cluster of $n_p = 16$ nodes using IEEE double-precision floating-point arithmetic ($\varepsilon \approx$ 2.2204e$-$16). Each node consists of an Intel Pentium Xeon processor@2.4 GHz with 1 GByte of RAM. We employ a BLAS library specially tuned for this processor that achieves around 3,800 Mflops (millions of flops per second) for the matrix product (routine DGEMM) [16]. 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 μsec. and a bandwidth of 1.4 Gbit/sec.

In the experiments we compare additive decomposition followed by the SR BT method (implemented using pab09ex+pab09ax) with the method described in Section III (using pab09fx). All the Newton-type iterative schemes were accelerated using appropriate variants of determinantal scaling.

In the evaluation we employ an example of order n with $m = p = n/10$ inputs and outputs, and 95% stable poles; $k = n \cdot 0.95$. The matrix A is generated as $A = U^T \text{diag}(-k, \ldots, -1, 1, \ldots, n-k)U$, where U is an $n \times n$ random orthogonal matrix. Matrices B, C, and D of the appropriate dimensions are generated following a random uniform distribution. The system is reduced to $r = n/10$.

A. Numerical Accuracy

We first evaluate the numerical behaviour of the methods. There are several measures that can help to asses the quality of the methods. Among these, we use the following relative residuals of the equations that are solved in the methods:

- Sylvester equation:

$$\mathcal{R}_S = \frac{\|A_{11}U - UA_{22} + A_{12}\|_F}{(\|A_{11}\|_F + \|A_{22}\|_F)\|U\|_F + \|A_{12}\|_F}.$$

- Lyapunov equations:

$$\begin{aligned}\mathcal{R}_{L_c} &= \frac{\|AW_c + W_cA^T + BB^T\|_F}{2(\|A\|_F\|W_c\|_F) + \|BB^T\|_F},\\ \mathcal{R}_{L_o} &= \frac{\|A^TW_o + W_oA + C^TC\|_F}{2(\|A\|_F\|W_o\|_F) + \|C^TC\|_F}.\end{aligned}$$

- ABEs:

$$\begin{aligned}\mathcal{R}_{B_c} &= \frac{\|A^TX + XA - XBB^TX\|_F}{2(\|A\|_F\|X\|_F) + \|X\|_F^2\|BB^T\|_F},\\ \mathcal{R}_{B_o} &= \frac{\|AY + YA^T - YC^TCY\|_F}{2(\|A\|_F\|Y\|_F) + \|Y\|_F^2\|C^TC\|_F}.\end{aligned}$$

Tables I and II report, among other data, these residuals. All relative residuals are below the order of the machine precision and the convergence of the Newton-type iterations is fast: between 8 and 12 iterations were required in all cases. For the approach based on additive decomposition, the low decoupling residual shows that the sign function experienced no trouble to separate the stable/unstable parts of this example. For the second approach, based on prior stabilization of the system, all eigenvalues of the closed-loop matrices $A - BB^TX$ and $A - YC^TC$ are stable, as shown by the largest real parts of the closed-loop poles.

Convergence sign (A)	$\|A_{12} - A_{11}\|_F/\|A\|_F$=1.08e$-$17
Decoupling residual	$\|A_{2,1}\|_F/\|A\|_F$=2.71e$-$15
Convergence Sylvester	$\max(\|E_8 + I_k\|_F/\sqrt{k},$ $\|F_8 + I_{n-k}\|_F/\sqrt{n-k})$=6.35e$-$18
Residual Sylvester	$\mathcal{R}_S$ = 1.11e$-$16
Convergence Lyapunov	$\|A_9 + I_k\|_F/\sqrt{k}$=2.67e$-$18
Residual Lyapunov	$\mathcal{R}_{L_c}$ = 9.94e$-$17
Residual Lyapunov	$\mathcal{R}_{L_o}$ = 8.64e$-$17

TABLE I

ACCURACY OF THE INVOLVED COMPUTATIONS FOR MODEL REDUCTION BASED ON SECTION II; RANDOM MODEL OF ORDER n=500.

Convergence ABE	$\|A_{11} - A_{10}\|_F/\|A\|_F = 4.14e - 15$
Residual ABE	$\mathcal{R}_{B_c} = 9.72e - 18$
Residual ABE	$\mathcal{R}_{B_o} = 3.56e - 18$
Stability	$\max(Re(\Lambda(A - BB^TX))) = -1.00$ $\max(Re(\Lambda(A - YC^TC))) = -1.00$
Convergence Lyapunov	$\|A_{11} + I_n\|_F/\sqrt{n} = 5.22e - 22$
Residual Lyapunov	$\mathcal{R}_{L_c} = 3.80e - 17$
Convergence Lyapunov	$\|A_{11} + I_n\|_F/\sqrt{n} = 2.70e - 22$
Residual Lyapunov	$\mathcal{R}_{L_o} = 4.60e - 17$

TABLE II

ACCURACY OF THE INVOLVED COMPUTATIONS FOR MODEL REDUCTION BASED ON SECTION III; RANDOM MODEL OF ORDER n=500.

As a measure of the quality of the reduced-order systems, we also compare the frequency responses of the TFM of the original system and those of the reduced-order realizations. In order to do so, the TFMs are evaluated at frequencies jw, with w composed of 500 samples logarithmically distributed in the interval [1.0e$-$5,1.0e+5].

Figure 1 reports in the left-hand side plot the frequency response of the original system and those of the reduced-order realizations computed by both methods. The plot in the right-hand side of this figure shows the absolute error in the frequency response, measured by $\|G(j\omega) - \hat{G}(j\omega)\|_2 = \sigma_{\max}(G(j\omega) - \hat{G}(j\omega))$. In this second plot, the error bound of the solid line corresponds to the theoretical error bound for the SR BT method applied to the stable part of the system. Both figures show that the dynamics of the original system is closely reproduced by the reduced-order realizations.

B. Performance

A second major criterion to appreciate the efficacy of the methods is the amount of time required for the reduction of the system. In this subsection we evaluate the model reduction algorithm using a system of order $n = 1,200$.

Figure 2 reports the execution time of both approaches. A first observation from the figure is that the algorithm based on additive decomposition, pab09ex+pab09ax, requires half the time required by algorithm pab09fx. For example, reducing the system using a single node via algorithms pab09ex+pab09ax and pab09fx requires, respectively, about 4.5 and 9.5 minutes. Parallel execution on 8 nodes

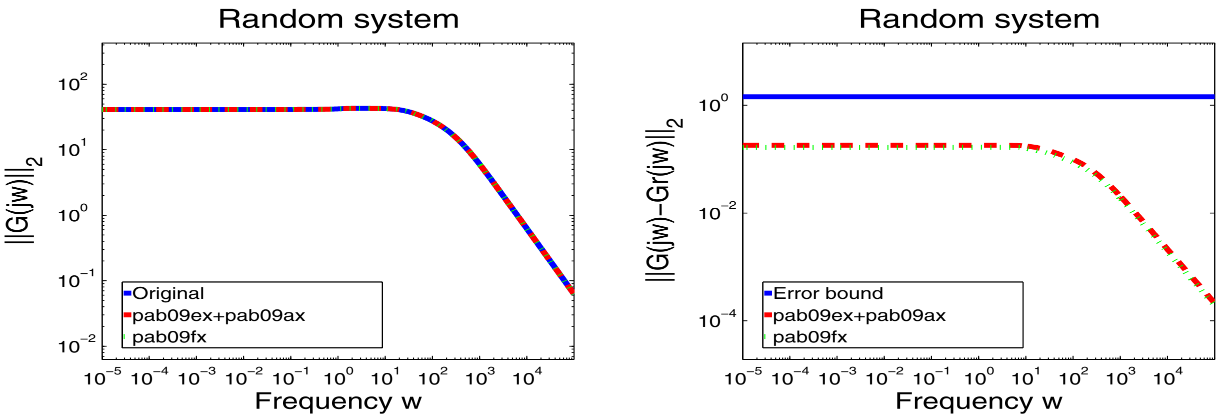

Fig. 1. Frequency response (left) and frequency response error (right); Random model of order $n = 500$.

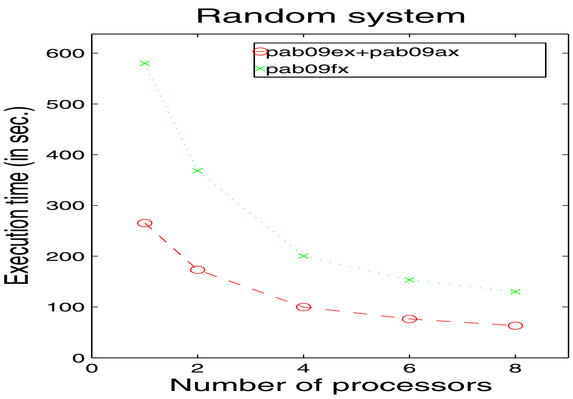

Fig. 2. Execution time; Random model of order $n = 1,200$.

reduces these times to scarcely a little bit more than 1 and 2 minutes, respectively. On the other hand, pab09fx has the potential to compute smaller reduced-order models than pab09ex+pab09ax if the number of unstable poles of the LTI system is large. Moreover, there is more parallelism in algorithm pab09fx which, e.g., achieves speed-ups of 1.57 and 4.46 for 2 and 8 nodes, respectively. For this same number of nodes, algorithm pab09ex+pab09ax delivers speed-ups of 1.53 and 4.19. The speed-ups in both algorithms are not remarkably high as the order of the largest system that can be reduced using a single processor considered here is $n = 1,200$. A larger system would surely provide higher speed-ups.

VII. CONCLUSIONS

We have evaluated two efficient parallel model reduction algorithms for unstable systems. All major computational problems that appear in the algorithms can be solved using Newton-type iterative schemes for the sign function. Experimental results with a random unstable system report both approaches as numerically reliable.

These model reduction algorithms are particularly simple to parallelize and perform efficiently on distributed-memory platforms. The scalability of the algorithms is ensured by that of the underlying iterative schemes. Thus, large-scale unstable systems can be reduced by employing a proportional amount of computational resources.

REFERENCES

[1] A. Antoulas, *Lectures on the Approximation of Large-Scale Dynamical Systems*. Philadelphia, PA: SIAM Publications, 2005.

[2] P. Benner, V. Mehrmann, and D. Sorensen, Eds., *Dimension Reduction of Large-Scale Systems*, ser. Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin/Heidelberg, Germany, 2005, vol. 45.

[3] G. Obinata and B. Anderson, *Model Reduction for Control System Design*, London, UK: Springer-Verlag, 2001.

[4] A. Varga, “Model reduction software in the SLICOT library,” in *Applied and Computational Control, Signals, and Circuits*, ser. The Kluwer International Series in Engineering and Computer Science, B. Datta, Ed. Boston, MA: Kluwer Academic Publishers, 2001, vol. 629, pp. 239–282.

[5] K. Zhou, G. Salomon, and E. Wu, “Balanced realization and model reduction for unstable systems,” *Int. J. Robust Nonlinear Control*, vol. 9, no. 3, pp. 183–198, 1999.

[6] L. 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. Whaley, *ScaLAPACK Users’ Guide*, SIAM, Philadelphia, PA, 1997.

[7] M. Safonov, E. Jonckheere, M. Verma, and D. Limebeer, “Synthesis of positive real multivariable feedback systems,” *Internat. J. Control*, vol. 45, no. 3, pp. 817–842, 1987.

[8] P. Lancaster and L. Rodman, *The Algebraic Riccati Equation*. Oxford: Oxford University Press, 1995.

[9] C. Kenney and A. Laub, “The matrix sign function,” *IEEE Trans. Automat. Control*, vol. 40, no. 8, pp. 1330–1348, 1995.

[10] J. Roberts, “Linear model reduction and solution of the algebraic Riccati equation by use of the sign function,” *Internat. J. Control*, vol. 32, pp. 677–687, 1980, (Reprint of Tech. Rep. No. TR-13, CUED/B-Control, Cambridge University, Engineering Dept., 1971).

[11] P. Lancaster and M. Tismenetsky, *The Theory of Matrices*, 2nd ed. Orlando: Academic Press, 1985.

[12] P. Benner, E. Quintana-Ortí, and G. Quintana-Ortí, “Solving stable Sylvester equations via rational iterative schemes,” *J. Sci. Comput.*, to appear.

[13] ——, “State-space truncation methods for parallel model reduction of large-scale systems,” *Parallel Comput.*, vol. 29, pp. 1701–1722, 2003.

[14] R. Byers, “Solving the algebraic Riccati equation with the matrix sign function,” *Linear Algebra Appl.*, vol. 85, pp. 267–279, 1987.

[15] V. Sima, *Algorithms for Linear-Quadratic Optimization*. New York, NY: Marcel Dekker, Inc., 1996.

[16] K. Goto and R. van de Geijn, “On reducing TLB misses in matrix multiplication,” Department of Computer Sciences, The University of Texas at Austin, http://www.cs.utexas.edu/users/flame, FLAME Working Note 9, 2002.