Zentrum für Technomathematik

Fachbereich 3 – Mathematik und Informatik

Partial Stabilization of Large-Scale Discrete-Time Linear Control Systems

Peter Benner Maribel Castillo
Enrique S. Quintana-Ortí

Report 01–04

Berichte aus der Technomathematik

Report 01–04 März 2001

Partial Stabilization of Large-Scale Discrete-Time Linear Control Systems*

Peter Benner† Maribel Castillo‡ Enrique S. Quintana-Ortí‡

March 15, 2001

Abstract

We propose a parallel algorithm for stabilizing large discrete-time linear control systems on a Beowulf cluster. Our algorithm first separates the Schur stable part of the linear control system using an inverse-free iteration for the matrix disc function, and then computes a stabilizing feedback matrix for the unstable part. This stage requires the numerical solution of a Stein equation. This linear matrix equation is solved using the sign function method after applying a Cayley transformation to the original equation.

The experimental results on a cluster composed of Intel P–II processors and a Myrinet interconnection network show the parallelism and scalability of our approach.

Keywords: Linear control systems, stabilization, Stein equation, invariant subspace, mathematical software.

1 Introduction

Consider a discrete time-invariant linear control system

$$x_{k+1} = Ax_k + Bu_k, \quad k = 0, 1, 2, \ldots, \tag{1}$$

where $x_0 = \hat{x}$ is given, $A \in \mathbb{R}^{n\times n}$ is the state matrix, and $B \in \mathbb{R}^{n\times m}$ is the input matrix. In case the spectrum (or set of eigenvalues) of the state matrix, denoted by $\Lambda(A)$, is contained in the open unit disc we say that A is *(Schur) stable* or *convergent* (in other words, $|\lambda| < 1$ for all $\lambda \in \Lambda(A)$). The stabilization problem consists in finding a feedback matrix $F \in \mathbb{R}^{m\times n}$ such that the input $u_k = Fx_k$, $k = 0, 1, 2, \ldots$, yields a stable closed-loop system

$$x_{k+1} = (A + BF)x_k, \quad k = 0, 1, 2, \ldots. \tag{2}$$

This problem has a solution if the matrix pair (A, B) is stabilizable, i.e., rank $([A - \lambda I_n, B]) = n$ for all λ with $|\lambda| \geq 1$ (hereafter, I_n denotes the identity matrix of order n) [20]. The stabilization problem arises in control problems such as, e.g., the computation of an initial approximate solution in Newton's method for solving discrete-time algebraic Riccati equations,

*Partially supported by the DAAD programme *Acciones Integradas Hispano-Alemanas*. Maribel Castillo and Enrique S. Quintana-Ortí were also supported by the Project GV99-59-1-14 of the *Generalidad Valenciana*.

†Zentrum für Technomathematik, Fachbereich 3/Mathematik und Informatik, Universität Bremen, D–28334 Bremen, Germany; benner@math.uni-bremen.de.

‡Departamento. de Ingeniería y Ciencia de los Computadores, Universidad Jaume I, 12.080–Castellón, Spain; {castillo,quintana}@inf.uji.es.

simple synthesis methods to design controllers, and many more [11, 12, 22, 29]. Large-scale problems occur whenever the linear system results from some sort of discretization of a partial differential equation or from delay systems. There, the number of states is often a couple of thousands.

The stabilization problem can in principal be solved as a pole assignment problem. Pole assignment methods compute a feedback matrix such that the state matrix of the closed-loop system (2) has a prespecified spectrum. In this sense, there is a high degree of freedom in the design of the stabilizing feedback matrix as the poles can be chosen to lie anywhere inside the unit circle. This approach however presents several drawbacks. First, the pole assignment problem is probably an intrinsically ill-conditioned problem for systems of order larger than 10 [3, 16, 23, 24]; second, how to choose the poles to improve the conditioning of the problem is still an open problem; and third, most of the pole assignment algorithms are based on QR algorithm-like procedures (see [3, 25, 31, 32] and the references therein) which are not well-suited for parallel computation. Therefore, they are too expensive for large control systems.

In this paper we follow an efficient approach for partial stabilization similar to those proposed in [7, 16, 29]. Our stabilization procedure is composed of two stages. We first use a spectral division technique, related with the matrix disc function [21, 4], to separate the stable and the unstable parts of the spectrum of the state matrix. This stage is composed of matrix algebra kernels such as QR factorizations, matrix products, etc., which are highly efficient on parallel distributed architectures [10].

The second method of Lyapunov is then employed to stabilize the unstable part of the system [2]. This stage requires solving a Stein equation of the form

$$\tilde{A}X\tilde{A}^T - X - \tilde{Q} = 0, \tag{3}$$

where $\tilde{Q}$ is symmetric positive semidefinite, and X is the sought-after solution. Under the given assumptions (i.e., (A, B) is stabilizable), the Stein equation arising in this stage has an anti-stable coefficient matrix ($|\lambda| > 1$ for all $\lambda \in \Lambda(\tilde{A})$), and a unique symmetric positive semidefinite solution.

The Bartels-Stewart method is one of the most well-known and efficient algorithms for solving Stein equations of moderate dimension [5]. In this method, the coefficient matrix $\tilde{A}$ is first reduced to a condensed form by means of the QR algorithm [14]; then, in a second stage, the solution is obtained from the reduced equation by a backsubstitution procedure. However, the QR algorithm is known to present a lack of scalability [17, 18]; moreover, the parallelism in this algorithm is lower than that of the usual matrix algebra kernels (matrix factorizations, matrix products, etc.).

For large order Stein equations with stable coefficient matrix, Stein solvers based on the Smith iteration or the sign function are more appropriate due to their parallel efficiency [8]. In case the coefficient matrix of the Stein equation is anti-stable (this is the case we are facing here; see Section 3), the Smith iteration can not be used as it only converges for stable coefficient matrices and therefore we propose to employ the sign function method.

The paper is structured as follows. In Section 2 we describe an inverse-free iteration for the matrix disc function which can be used to divide the spectrum of a matrix along the unit circle. In Sections 3 and 4 we review, respectively, the stabilizing procedure via the second method of Lyapunov, and a solver for the Stein equation based on the sign function of a Cayley transformed matrix pair. Section 5 discusses a few implementation details of the complete algorithm for partial stabilization. Section 6 reports our numerical experiments on a cluster of personal computers based on Intel P–II processors connected via a high-speed system area network (Myrinet). Finally, in Section 7 we summarize our concluding remarks.

2 Spectral division with the matrix disc function

There exist several definitions of the matrix disc function (see, e.g., [6, 28]). For instance, let

$$Z = S \begin{bmatrix} J^0 & 0 \\ 0 & J^\infty \end{bmatrix} S^{-1}$$

be the Jordan decomposition of $Z \in \mathbb{R}^{n\times n}$ [14], where the Jordan blocks in $J^0 \in \mathbb{C}^{k\times k}$ and $J^\infty \in \mathbb{C}^{(n-k)\times(n-k)}$ contain, respectively, the eigenvalues of Z inside and outside the unit circle. In case Z has no eigenvalues on the unit circle, the *matrix disc function* is given by

$$\operatorname{disc}(Z) \ := \ S \begin{bmatrix} I_k & 0 \\ 0 & 0 \end{bmatrix} S^{-1}. \tag{4}$$

Note that $\operatorname{disc}(Z)$ is unique and independent of the order of the eigenvalues in the Jordan decomposition of Z [20].

In [21], Malyshev proposed an iterative procedure to divide the spectrum of a matrix pencil along the unit circle, which can also be used to compute the matrix disc function of the pencil [6]. In [4], Bai, Demmel and Gu refined this algorithm and provided a truly inverse-free procedure for spectral division. Specifically, when applied to a matrix Z with no eigenvalues on the unit circle the algorithm is based on the following iteration:

$$\begin{aligned} \begin{bmatrix} Y_k \\ -Z_k \end{bmatrix} &= \begin{bmatrix} U_{11} & U_{12} \\ U_{21} & U_{22} \end{bmatrix} \begin{bmatrix} R_k \\ 0 \end{bmatrix} \quad \text{(QR factorization)}, \\ Z_{k+1} &:= U_{12}^T Z_k, \quad Y_{k+1} := U_{22}^T Y_k, \end{aligned} \tag{5}$$

with $Z_0 := Z$ and $Y_0 := I_n$.

A practical stopping criterion for the iteration is to stop when $\|R_{k+1} - R_k\| < c\,\sqrt{\varepsilon}\,\|R_k\|$ for a suitable matrix norm $\|\cdot\|$, a small order constant c, and ε the machine precision. Once the stopping criterion is satisfied, two more iterations are carried out. Due to the quadratic convergence of the iteration [4] the maximum attainable accuracy is ensured.

The matrix pair at convergence, (Z_s, Y_s), can be used to compute a basis for the stable invariant subspace of Z as follows. First, compute a rank-revealing QR (RRQR) factorization [14]

$$Y_s = \bar{U}\bar{R}\Pi,$$

where $\bar{U}$ is orthogonal, Π is a permutation matrix, and $\bar{R}$ is upper triangular, with $\operatorname{rank}(\bar{R}) = k$. Next, factorize

$$\bar{U}^T(Z_s + Y_s) = RU \quad \text{(RQ factorization)},$$

where U is orthogonal, and R is upper triangular. The matrix U divides the spectrum of Z as

$$U^T Z U = \begin{bmatrix} Z_{11} & Z_{12} \\ 0 & Z_{22} \end{bmatrix}, \qquad Z_{11} \in \mathbb{R}^{k\times k},\ Z_{22} \in \mathbb{R}^{(n-k)\times(n-k)}. \tag{6}$$

Here, $\Lambda(Z_{11})$ contains the eigenvalues of Z inside the unit circle and $\Lambda(Z_{22})$ contains the eigenvalues outside the unit circle. The first k columns of U are associated with the stable part of the spectrum of Z and form an orthonormal basis for the stable invariant subspace of this matrix.

The spectral division using the inverse-free iteration for the matrix disc function is only composed of matrix kernels like matrix multiplication and (rank-revealing) QR factorizations which can be efficiently implemented on parallel distributed architectures. Portable computational kernels for this matrix operations are provided in parallel linear algebra libraries as, e.g., ScaLAPACK and PLAPACK [10, 30].

The matrix disc function is related with the inverse-free iteration (5) as shown in [6]:

$$\begin{aligned} \mathrm{disc}\,(Z) &= \lim_{k\to\infty}(Z_k + Y_k)^{-1}Y_k, \\ I - \mathrm{disc}\,(Z) &= \lim_{k\to\infty}(Z_k + Y_k)^{-1}Z_k. \end{aligned}$$

3 Stabilizing discrete-time linear control systems

The second method of Lyapunov is a simple numerical procedure for stabilizing linear control systems. Specifically, in the discrete-time case, the method relies on the following result [2].

Theorem 3.1 *Let the pair (A, B) be stabilizable. Then,*

$$F = -B^T(X + BB^T)^+A$$

is a stabilizing feedback matrix, where X is the solution of the Stein equation

$$AXA^T - \alpha^2X - 2BB^T = 0, \quad 0 < \alpha < \min(1, \min\{|\lambda|; \lambda \in \Lambda(A)\}),$$

and Z^+ denotes the Moore-Penrose inverse of a matrix Z [14].

Consider we have to stabilize a system defined by the matrix pair (A, B), and assume U divides the spectrum of A as in (6). Applying this transformation to the matrix B, we obtain

$$U^TB = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix},$$

where $B_1 \in \mathbb{R}^{k\times n}$ and $B_2 \in \mathbb{R}^{(n-k)\times n}$; the stabilization problem is then reduced to finding a feedback matrix $F_2 \in \mathbb{R}^{m\times(n-k)}$ that stabilizes the matrix pair (A_{22}, B_2).

Note that the matrix A_{22} is anti-stable ($\Lambda(A_{22})$ lies outside the unit circle) and therefore we can simply set $\alpha = 1$ in the procedure described in Theorem 3.1 to stabilize (A_{22}, B_2). In order to solve the anti-stable Stein equation associated with (A_{22}, B_2), we propose to use the sign function method as described in the next section.

4 Solving Stein equations with the sign function method

The Stein equation is related with the Lyapunov equation of the form

$$\hat{A}X + X\hat{A}^T + \hat{Q} = 0, \tag{7}$$

via the *Cayley transformation*. Specifically, if we apply the transformation

$$c(A) = (A - I_n)^{-1}(A + I_n) \tag{8}$$

to A from (3), then the Stein equation is equivalent to the Lyapunov equation (7) with $\hat{A} = c(A)$ and $\hat{Q} = 2(A - I_n)^{-1}Q(A - I_n)^{-T}$. In other words, the Stein equation and the Lyapunov equation resulting from the Cayley transformation share the same solution.

The Stein equation can also be related with the generalized Lyapunov equation $\bar{A}X\bar{E}^T + \bar{E}X\bar{A}^T + \bar{Q} = 0$, with $\bar{A} = (A + I_n)$, $\bar{E} = (E - I_n)$, and $\bar{Q} = 2Q$ [8]. No explicit inversions are required for this transformation. The cost of solving this generalized Lyapunov equation is, in general, higher than that of solving the standard equation in (7), so we do not explicitly investigate this approach here as we focus on the parallelization aspect of the problem. Note, however, that in case $c(A)$ is ill-conditioned, using the generalized Lyapunov equation may give more accurate results.

The sign function method was first introduced in 1971 by Roberts for solving algebraic Riccati equations [28]. Roberts also shows how to solve Lyapunov equations via the matrix sign function in case $\Lambda(\hat{A})$ is contained in the open left half complex plane. (The algorithm can also be applied in case $\Lambda(\hat{A})$ is contained in the open right half complex plane as is the case when $\hat{A} = c(A)$ and A is anti-stable.)

Let $Z \in \mathbb{R}^{l \times l}$ have no eigenvalues on the imaginary axis and denote by

$$Z = S \begin{bmatrix} J^- & 0 \\ 0 & J^+ \end{bmatrix} S^{-1}$$

its Jordan decomposition with $J^- \in \mathbb{C}^{k \times k}$, $J^+ \in \mathbb{C}^{(l-k)\times(l-k)}$ containing the Jordan blocks corresponding to the eigenvalues in the open left and right half planes, respectively. Then the *matrix sign function* of Z is defined as

$$\operatorname{sign}(Z) := S \begin{bmatrix} -I_k & 0 \\ 0 & I_{n-k} \end{bmatrix} S^{-1}.$$

Note that $\operatorname{sign}(Z)$ is unique and independent of the order of the eigenvalues in the Jordan decomposition of Z [20]. Many other equivalent definitions for $\operatorname{sign}(Z)$ can be given. For more details see, e.g., the survey paper [19].

The sign function can be computed via the Newton iteration for the equation $Z^2 = I_l$ where the starting point is chosen as Z, i.e.,

$$Z_0 := Z, \qquad Z_{k+1} := \left(Z_k + Z_k^{-1}\right)/2, \qquad k = 0, 1, 2, \ldots. \tag{9}$$

It is shown in [28] that $\operatorname{sign}(Z) = \lim_{k\to\infty} Z_k$ and moreover that

$$\operatorname{sign}\left(\begin{bmatrix} \hat{A}^T & 0 \\ \hat{Q} & -\hat{A} \end{bmatrix}\right) + I_{2n} = 2\begin{bmatrix} 0 & 0 \\ X & I \end{bmatrix}; \tag{10}$$

i.e., under the given assumptions, (7) can be solved by applying the iteration (9) to $Z_0 := \begin{bmatrix} \hat{A} & 0 \\ \hat{Q} & -\hat{A}^T \end{bmatrix}$. In [28] it is also observed that applying the Newton iteration (9) to this matrix and exploiting the block-triangular structure of all matrices involved, (9) boils down to

$$\begin{aligned} A_0 &:= \hat{A}, & A_{k+1} &:= \tfrac{1}{2}\left(A_k + A_k^{-1}\right), \\ Q_0 &:= \hat{Q}, & Q_{k+1} &:= \tfrac{1}{2}\left(Q_k + A_k^{-1}Q_kA_k^{-T}\right), \end{aligned} \qquad k = 0, 1, 2, \ldots, \tag{11}$$

and hence from (10) it follows that $X = \frac{1}{2}(\lim_{k\to\infty} Q_k)$.

When $\hat{A}$ is obtained as $\hat{A} = c(A)$, with A an anti-stable matrix, $\lim_{k\to\infty} A_k = I_n$ and a suitable stopping criterion is $\|A - I_n\| < c\,\sqrt{\varepsilon}\|A\|$; two more iterations are performed once this criterion is satisfied as the quadratic convergence of the iteration ensures the maximum attainable accuracy.

Other iterative schemes for computing the sign function like the Newton-Schulz iteration or Halley's method (see, e.g., [19]) can also be implemented efficiently to solve Lyapunov equations.

The solution of the Lyapunov equation via (11) only requires basic numerical linear algebra tools like inversion and/or solution of linear systems. Here we propose to use a matrix inversion algorithm based on Gauss-Jordan transformations which provides a higher efficiency than the usual procedure based on LU factorization [26]. Hence, the sign function method is an appropriate tool to design and implement efficient and portable numerical software for distributed memory parallel computers.

5 Implementation issues

The proposed stabilization method for a discrete-time linear system represented by a matrix pair (A, B) can be described algorithmically as follows.

1. Apply iteration (5), with $Z_0 = A$, $Y_0 = I_n$, until convergence.
2. Compute an RRQR factorization
$$Y_s = \bar{U}\bar{R}\Pi,$$
where $\bar{U}$ is orthogonal, Π is a permutation matrix, and $\bar{R}$ is upper triangular, with rank $(\bar{R}) = k$.
3. Compute an RQ factorization
$$\bar{U}^T(Z_s + Y_s) = RU,$$
where U is orthogonal and R is upper triangular.
4. Let $U = (U_1, U_2)$ be a partitioning of U, with $U_1 \in \mathbb{R}^{n\times k}$ and $U_2 \in \mathbb{R}^{n\times(n-k)}$. Compute $A_{22} = U_2^T A U_2$ and $B_2 = U_2^T B$.
5. Solve the Stein equation
$$A_{22} X A_{22}^T - X - 2B_2B_2^T = 0,$$
using the sign function method applied to the Lyapunov equation resulting from the Cayley transformation as in (7), (8).
6. Compute $F_2 = -B_2^T(X + B_2B_2^T)^+ A_{22}$ and the stabilizing feedback as $F = (0, F_2)U^T$.

In stage 2, we can use a QR factorization with column pivoting to obtain an approximate rank-revealing factorization [14]. Theoretically this orthogonal factorization may fail to reveal the rank, though in practice this is a reliable numerical tool [9, 27].

In case A has eigenvalues on the unit circle we can nevertheless apply the algorithm to the matrix pair $(A/\tau, B)$ for $\tau \in \mathbb{R}$ slightly smaller than 1. Thus, we divide the spectrum of A

along a circle of radius τ and stabilize those eigenvalues with absolute magnitude larger than τ. Choosing τ carefully so that A/τ has no eigenvalues close to the unit circle we can avoid numerical difficulties associated with eigenvalues close to the unit circle. (Notice that, in such a case, we have to modify the corresponding value of α in Theorem 3.1.) The same technique can also be used to obtain a certain degree of stability, i.e., $\Lambda(A + BF) \subset \{\lambda \in \mathbb{C}; |\lambda| < \tau\}$.

In control problems usually the state matrix only has a few unstable eigenvalues. The subsystem to stabilize in Stages 4, 5, and 6 is small and the cost of these stages is therefore negligible when compared to that of Stages 1, 2 and 3.

Our algorithms are implemented using ScaLAPACK (scalable linear algebra package) and PB-BLAS (parallel block basic linear algebra subprograms) [10]. These are public-domain parallel libraries for MIMD computers which can be run on any machine that supports either PVM [13] or MPI [15]. ScaLAPACK provides scalable parallel distributed kernels for many of the matrix algebra kernels available in LAPACK [1]. This library employs BLAS and LAPACK for serial computations and the BLACS (basic linear algebra communication subprograms) for communication.

6 Experimental results

In this section we evaluate and compare the performance of the following algorithms for partial stabilization of discrete linear control systems:

- `PDGEDST`: Parallel routine based on the matrix disc function for spectral division and the Cayley transformation+sign function for solving the Stein equation.
- `DGEDSTQR`: Serial routine based on the QR algorithm [14] for both dividing the spectrum and solving the corresponding Stein equation.

Both routines are implemented using Fortran 77 and the kernels in LAPACK. A parallel implementation of `DGEDSTQR` is not possible as this algorithm requires several matrix kernels which are not available in ScaLAPACK (basically, reordering of eigenvalues in the Schur form and solution of a Stein equation with triangular coefficient matrix).

All our experiments were performed using IEEE double precision arithmetic ($\varepsilon \approx 2.204 \times 10^{-16}$) on a cluster of Intel P-II processors at 300MHz, with 128 MBytes of RAM each. The BLAS implementation we used achieves around 180 Mflops (millions of floating–point arithmetic operations per second) for the matrix-matrix product (routine `DGEMM`). The nodes of the system were connected via a Myrinet multistage interconnection network. A simple loop–back test offered a latency of 33 μsec. and a bandwidth around 200 Mbit/sec. for this network.

In our first experiment we generate a random matrix pair (A, B), with $n = m = 20$ and entries uniformly distributed in $[-1, 1]$. In Figure 1 we report the distribution of the spectrum of A and $A + BF$, with F computed by means of algorithm `PDGEDST`. The figure shows that, after applying the algorithm, the closed–loop system is stable. No significant differences were found when algorithm `DGEDSTQR` was employed.

We next evaluate the performance of the parallel stabilizing algorithms based on the disc and sign functions. Following the usual case in control problems, we generate random matrix pairs (A, B) with a large number of stable eigenvalues (about 99%). The stable and unstable eigenvalues are uniformly distributed in $[0, 1)$ and $(1, 10]$, respectively. The cost of the parallel stabilizing algorithms is basically due to the cost of separating the spectrum of the stable and

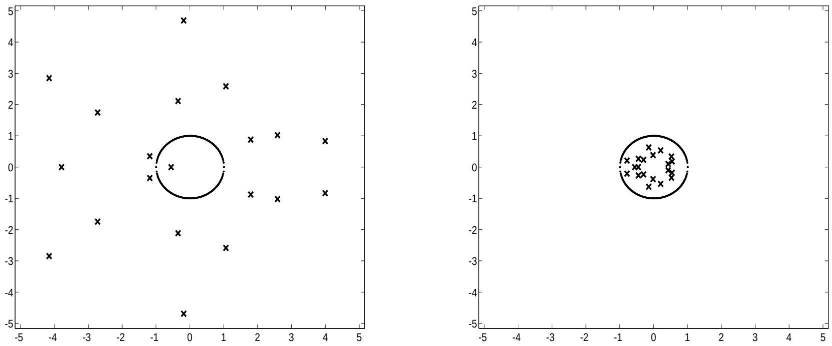

Figure 1: Distribution of $\Lambda(A)$ (left) and $\Lambda(A+BF)$ (right), with A and B random matrices ($n = m = 20$), and F computed by means of algorithm PDGEDST.

anti-stable part of the system as stabilizing the anti-stable part only requires the resolution of a small Stein equation (of dimension $\mathcal{O}(n/100)$). Actually, this equation is so small that it can be solved serially on 1 node using, e.g., the QR algorithm [5]. No noticeable differences were found in both the execution time and the numerical accuracy in that case.

Figure 2 shows the execution times of the stabilizing algorithms DGEDSTQR (results on 1 processor) and PDGEDST (results on $n_p = 2, 4, \ldots, 16$ processors) for a large linear control system. In general, using the QR algorithm to separate the spectrum is computationally less expensive than using the disc function for the same purpose (depending on the number of iterations required by the inverse-free iterative scheme). However, the parallelism of the disc function approach provides a considerable reduction in the execution time when the number of processors is increased from 2 to 4 and from there to 6. The reduction is less significant for a larger number of processors as the ratio between the problem size and the number of processors becomes too small.

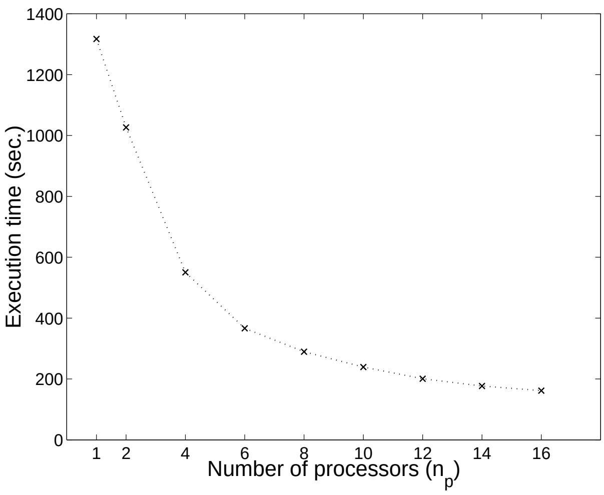

Figure 2: Execution times of the stabilization algorithms DGEDSTQR and PDGEDST for n=m=1400.

Figure 3 reports the execution time of algorithms DGEDSTQR (results on 1 processor) and PDGEDST (results on $n_p = 2, 4, \dots, 16$ processors) for linear control systems (A, B) of varying dimension $n = m$. (As most of the computational cost is spent dividing the spectrum of the state matrix no significant difference was found for a usual case in control, $m \ll n$.)

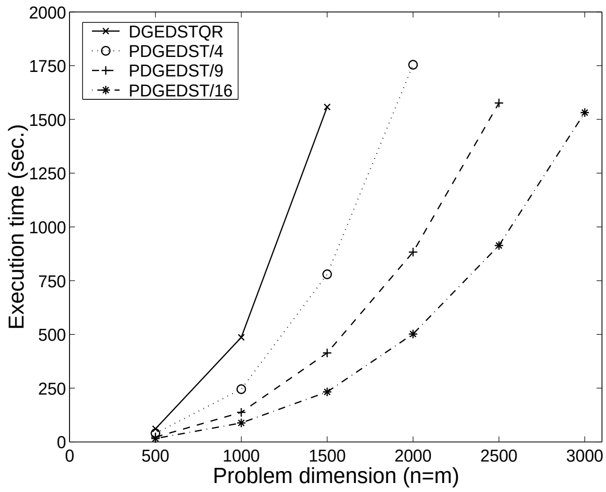

Figure 3: Execution times of DGEDSTQR and PDGEDST on 4, 9, 16 processors.

We also analyze the scalability of the parallel stabilizing algorithms. For this purpose, we fix the dimension of the problem per node to $n/\sqrt{n_p} = 1000$ and compute the Mflop ratio per node (millions of flops per second on a node) of the algorithm. Figure 4 reports a high scalability of our parallel algorithms as there is only a minor decrease in the Mflop ratio as the number of processors is increased.

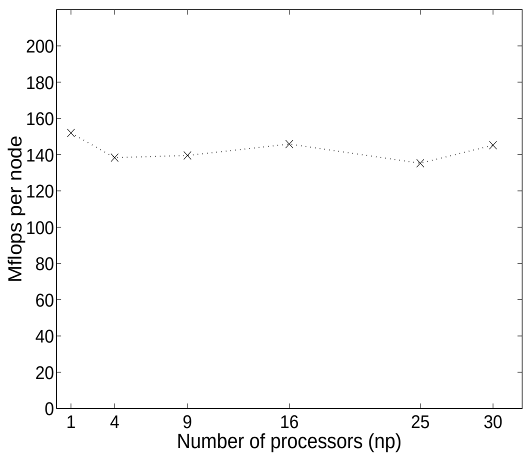

Figure 4: Mflop ratio per node of the parallel stabilizing algorithm PDGEDST for $n/\sqrt{n_p}$=1000.

7 Concluding remarks

We have presented parallel algorithms for the stabilization of large discrete-time linear control systems. Our new solvers employ an inverse-free iteration for the matrix disc function to initially separate the unstable part of the spectrum of the state matrix. The subsystem is stabilized using the second method of Lyapunov and the Stein equation arising in this stage is solved by means of the sign function applied to a Cayley transformed pair.

Our two-stage approach can be used to stabilize large linear control systems, with a few thousands of state variables, and only requires scalable matrix algebra kernels which are highly efficient on parallel distributed architectures. The experimental results on a Beowulf cluster show the performance of our new parallel routines.

References

[1] E. Anderson *et al.* *LAPACK Users' Guide.* SIAM, Philadelphia, PA, second edition, 1994.

[2] E. S. Armstrong and G. T. Rublein. A stabilization algorithm for linear discrete constant systems. *IEEE Trans. Automat. Control*, AC–21:629–631, 1976.

[3] M. Arnold. Algorithms and conditioning for eigenvalue assignment. Master's thesis, Northern Illinois University, Dept. of Mathematical Sciences, DeKalb, Il, 1993.

[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. *See also:* Tech. Report LBL-34969, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720.

[5] R.H. Bartels and G.W. Stewart. Solution of the matrix equation $AX + XB = C$: Algorithm 432. *Comm. ACM*, 15:820–826, 1972.

[6] P. Benner. *Contributions to the Numerical Solution of Algebraic Riccati Equations and Related Eigenvalue Problems.* Dissertation, Fak. f. Mathematik, TU Chemnitz–Zwickau, Chemnitz, FRG, 1997.

[7] P. Benner, M. Castillo, E.S. Quintana-Ortí, and V. Hernández. Parallel partial stabilizing algorithms for large linear control systems. *J. Supercomputing*, 15(2):193–206, 2000.

[8] P. Benner, E.S. Quintana-Ortí, and G. Quintana-Ortí. Numerical solution of Schur stable linear matrix equations on multicomputers. Berichte aus der Technomathematik, Report 99–13, FB3 – Mathematik und Informatik, Universität Bremen, 28334 Bremen (Germany), November 1999. Available from `http://www.math.uni-bremen.de/zetem/berichte.html`.

[9] C.H. Bischof and G. Quintana-Ortí. Computing rank-revealing QR factorizations of dense matrices. *ACM Trans. Math. Software*, 24(2):226–253, 1998.

[10] L.S. Blackford *et al.* *ScaLAPACK Users' Guide.* SIAM, Philadelphia, PA, 1997.

[11] L. Dai. *Singular Control Systems*, volume 118 of *Lecture Notes in Control and Inform. Sci.* Springer, New York, NY, 1989.

[12] V. Dragan and A. Halanay. *Stabilization of Linear Systems.* Birkhäuser, Basel, Switzerland, 1994.

[13] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, B. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine – A User's Guide and Tutorial for Network Parallel Computing.* MIT Press, 1994.

[14] G.H. Golub and C.F. Van Loan. *Matrix Computations.* Johns Hopkins University Press, Baltimore, third edition, 1996.

[15] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* MIT Press, 1994.

[16] C. He and V. Mehrmann. Stabilization of large linear systems. In L. Kulhavá, M. Kárný, and K. Warwick, editors, *Preprints of the European IEEE Workshop CMP'94, Prague, September 1994*, pages 91–100, 1994.

[17] 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.

[18] G. Henry, D. Watkins, and J.J. Dongarra. A parallel implementation of the nonsymmetric QR algorithm for distributed memory architectures. LAPACK Working Note 121, University of Tennessee at Knoxville, 1997.

[19] C. Kenney and A.J. Laub. The matrix sign function. *IEEE Trans. Automat. Control*, 40(8):1330–1348, 1995.

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

[21] A.N. Malyshev. Parallel algorithm for solving some spectral problems of linear algebra. *Linear Algebra Appl.*, 188/189:489–520, 1993.

[22] V. Mehrmann. *The Autonomous Linear Quadratic Control Problem, Theory and Numerical Solution.* Number 163 in Lecture Notes in Control and Information Sciences. Springer-Verlag, Heidelberg, July 1991.

[23] V. Mehrmann and H. Xu. An analysis of the pole placement problem. I. The single-input case. *Electr. Trans. Num. Anal.*, 4:89–105, 1996.

[24] V. Mehrmann and H. Xu. Analysis of the pole placement problem II. The multi-input case. *Electr. Trans. Num. Anal.*, 5:77–97, 1997.

[25] G. Miminis and C.C. Paige. An algorithm for pole assignment of time invariant linear systems. *Internat. J. Control*, 35:341–354, 1982.

[26] E.S. Quintana-Ortí, G. Quintana-Ortí, X. Sun, and R. van de Geijn. A note on parallel matrix inversion. *SIAM J. Sci. Comput.*, 2001. To appear.

[27] 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.

[28] 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.

[29] V. Sima. *Algorithms for Linear-Quadratic Optimization*, volume 200 of *Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, NY, 1996.

[30] R. van de Geijn. *Using PLAPACK: Parallel Linear Algebra Package*. MIT Press, Cambridge, MA, 1997.

[31] A. Varga. A Schur method for pole assignment. *IEEE Trans. Automat. Control*, AC-27:517–519, 1981.

[32] A. Varga. A multishift Hessenberg method for pole assignment of single-input systems. *IEEE Trans. Automat. Control*, AC-41(12):1795–1799, 1996.

Berichte aus der Technomathematik

ISSN 1435-7968

http://www.math.uni-bremen.de/zetem/berichte.html

— Vertrieb durch den Autor —

Reports

Stand: 20. März 2001

98–01. Peter Benner, Heike Faßbender:
An Implicitly Restarted Symplectic Lanczos Method for the Symplectic Eigenvalue Problem, Juli 1998.

98–02. Heike Faßbender:
Sliding Window Schemes for Discrete Least-Squares Approximation by Trigonometric Polynomials, Juli 1998.

98–03. Peter Benner, Maribel Castillo, Enrique S. Quintana-Ortí:
Parallel Partial Stabilizing Algorithms for Large Linear Control Systems, Juli 1998.

98–04. Peter Benner:
Computational Methods for Linear–Quadratic Optimization, August 1998.

98–05. Peter Benner, Ralph Byers, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Solving Algebraic Riccati Equations on Parallel Computers Using Newton's Method with Exact Line Search, August 1998.

98–06. Lars Grüne, Fabian Wirth:
On the rate of convergence of infinite horizon discounted optimal value functions, November 1998.

98–07. Peter Benner, Volker Mehrmann, Hongguo Xu:
A Note on the Numerical Solution of Complex Hamiltonian and Skew-Hamiltonian Eigenvalue Problems, November 1998.

98–08. Eberhard Bänsch, Burkhard Höhn:
Numerical simulation of a silicon floating zone with a free capillary surface, Dezember 1998.

99–01. Heike Faßbender:
The Parameterized SR Algorithm for Symplectic (Butterfly) Matrices, Februar 1999.

99–02. Heike Faßbender:
Error Analysis of the symplectic Lanczos Method for the symplectic Eigenvalue Problem, März 1999.

99–03. Eberhard Bänsch, Alfred Schmidt:
Simulation of dendritic crystal growth with thermal convection, März 1999.

99–04. Eberhard Bänsch:
Finite element discretization of the Navier-Stokes equations with a free capillary surface, März 1999.

99–05. Peter Benner:
Mathematik in der Berufspraxis, Juli 1999.

99–06. Andrew D.B. Paice, Fabian R. Wirth:
Robustness of nonlinear systems and their domains of attraction, August 1999.

99–07. Peter Benner, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Balanced Truncation Model Reduction of Large-Scale Dense Systems on Parallel Computers, September 1999.

99–08. Ronald Stöver:
Collocation methods for solving linear differential-algebraic boundary value problems, September 1999.

99–09. Huseyin Akcay:
Modelling with Orthonormal Basis Functions, September 1999.

99–10. Heike Faßbender, D. Steven Mackey, Niloufer Mackey:
Hamilton and Jacobi come full circle: Jacobi algorithms for structured Hamiltonian eigenproblems, Oktober 1999.

99–11. Peter Benner, Vincente Hernández, Antonio Pastor:
On the Kleinman Iteration for Nonstabilizable System, Oktober 1999.

99–12. Peter Benner, Heike Faßbender:
A Hybrid Method for the Numerical Solution of Discrete-Time Algebraic Riccati Equations, November 1999.

99–13. Peter Benner, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Numerical Solution of Schur Stable Linear Matrix Equations on Multicomputers, November 1999.

99–14. Eberhard Bänsch, Karol Mikula:
Adaptivity in 3D Image Processing, Dezember 1999.

00–01. Peter Benner, Volker Mehrmann, Hongguo Xu:
Perturbation Analysis for the Eigenvalue Problem of a Formal Product of Matrices, Januar 2000.

00–02. Ziping Huang:
Finite Element Method for Mixed Problems with Penalty, Januar 2000.

00–03. Gianfrancesco Martinico:
Recursive mesh refinement in 3D, Februar 2000.

00–04. Eberhard Bänsch, Christoph Egbers, Oliver Meincke, Nicoleta Scurtu:
Taylor-Couette System with Asymmetric Boundary Conditions, Februar 2000.

00–05. Peter Benner:
Symplectic Balancing of Hamiltonian Matrices, Februar 2000.

00–06. Fabio Camilli, Lars Grüne, Fabian Wirth:
A regularization of Zubov's equation for robust domains of attraction, März 2000.

00–07. Michael Wolff, Eberhard Bänsch, Michael Böhm, Dominic Davis:
Modellierung der Abkühlung von Stahlbrammen, März 2000.

00–08. Stephan Dahlke, Peter Maaß, Gerd Teschke:
Interpolating Scaling Functions with Duals, April 2000.

00–09. Jochen Behrens, Fabian Wirth:
A globalization procedure for locally stabilizing controllers, Mai 2000.

00–10. Peter Maaß, Gerd Teschke, Werner Willmann, Günter Wollmann:
Detection and Classification of Material Attributes – A Practical Application of Wavelet Analysis, Mai 2000.

00–11. Stefan Boschert, Alfred Schmidt, Kunibert G. Siebert, Eberhard Bänsch, Klaus-Werner Benz, Gerhard Dziuk, Thomas Kaiser:
Simulation of Industrial Crystal Growth by the Vertical Bridgman Method, Mai 2000.

00–12. Volker Lehmann, Gerd Teschke:
Wavelet Based Methods for Improved Wind Profiler Signal Processing, Mai 2000.

00–13. Stephan Dahlke, Peter Maass:
A Note on Interpolating Scaling Functions, August 2000.

00–14. Ronny Ramlau, Rolf Clackdoyle, Frédéric Noo, Girish Bal:
Accurate Attenuation Correction in SPECT Imaging using Optimization of Bilinear Functions and Assuming an Unknown Spatially-Varying Attenuation Distribution, September 2000.

00–15. Peter Kunkel, Ronald Stöver:
Symmetric collocation methods for linear differential-algebraic boundary value problems, September 2000.

00–16. Fabian Wirth:
The generalized spectral radius and extremal norms, Oktober 2000.

00–17. Frank Stenger, Ahmad Reza Naghsh-Nilchi, Jenny Niebsch, Ronny Ramlau:
A unified approach to the approximate solution of PDE, November 2000.

00–18. Peter Benner, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Parallel algorithms for model reduction of discrete–time systems, Dezember 2000.

00–19. Ronny Ramlau:
A steepest descent algorithm for the global minimization of Tikhonov–Phillips functional, Dezember 2000.

01–01. Efficient methods in hyperthermia treatment planning:
Torsten Köhler, Peter Maass, Peter Wust, Martin Seebass, Januar 2001.

01–02. Parallel Algorithms for LQ Optimal Control of Discrete-Time Periodic Linear Systems:
Peter Benner, Ralph Byers, Rafael Mayo, Enrique S. Quintana-Ortí, Vicente Hernández, Februar 2001.

01–03. Peter Benner, Enrique S. Quintana-Ortí, Gregorio Quintana-Ortí:
Efficient Numerical Algorithms for Balanced Stochastic Truncation, März 2001.

01–04. Peter Benner, Maribel Castillo, Enrique S. Quintana-Ortí:
Partial Stabilization of Large-Scale Discrete-Time Linear Control Systems, März 2001.