Numerical Linear Algebra Techniques for Large Scale Matrix Problems in Systems and Control

Paul Van Dooren
University of Illinois at Urbana-Champaign
Coordinated Science Laboratory
1308 W. Main Str.
Urbana, IL 61801

Abstract

During the last few decades linear algebra has played an important role in advances being made in the area of systems and control. The most profound impact has been in the computational and implementational aspects, where numerical linear algebraic algorithms have strongly influenced the ways in which problems are being solved. The advent of special computing architectures such as vector processors and distributed processor arrays has also emphasized parallel and real-time processing of basic linear algebra modules for this application area. This paper discusses a number of numerical linear algebra techniques for large scale problems in systems and control. We focus on "special matrix"-problems, i.e. matrices which are either sparse, patterned or structured.

1. Introduction

Large plants in control typically arise from discretizations of continuum problems (such as those that appear in mechanics or chemistry). The models obtained from that are then automatically sparse, such as finite element methods used for mechanical problems. Moreover the underlying structure of the problem often yields additional quantitative properties, such diagonal dominance [17] or near decoupling of the matrices involved [24], [18]. It is also commonly observed that large scale problems in other application areas typically involve matrices with some kind of sparsity or structure. A recent report on large matrix problems [10] shows that indeed there are *no large general dense matrix problems* to be tackled in most application areas.

For large scale systems, the design of the controller, on the other hand, is often faced with the practical constraint that it has to be of relatively low order. The main reasons for this are that the controller has to run in real time (at a specific clock rate) or has to be implemented on existing hardware of limited capacity. A typical example of this is the laser beam tracking device of the compact disc player which has to be implemented on an existing digital signal processor with relatively small memory and processing speed (as compared to the tracking speed requested from it). This constraint e.g. limits the order of the controller to 15, whereas finite element models of order up to 1000 have been used for modeling the rigid body movements of the CD player (these are the ones affecting the tracking performance) [25].

Large scale system models are almost always represented in state space or generalized state space form. Since the introduction of state space models for representing linear time invariant systems several linear algebra techniques have been proposed for solving various analysis and design problems in control [27][19][28]. The more classical transfer matrix techniques have gradually lost popularity because of the wealth of techniques available these days when using state space models. Recently, robust control has changed the emphasis back to transfer function models, because that framework is inherent to the formulation of robust control. Yet, computational methods proposed to tackle these problems again seem to prefer translating the problem back to state space and eventually using linear algebra techniques to solve these. Unfortunately, some of the currently proposed solutions are still cumbersome and un-

appealing.

Often one wants to combine powerful concepts of both approaches in order to yield reliable algorithms for a class of important design problems: factorization and optimal approximation of transfer functions via state space techniques. In both these problems the formulation is more elegant and compact in the transfer function domain whereas solution methods are expected to be more reliable in the state space domain. In [3][14] it is shown how various transfer matrix factorizations can be solved in state-space using linear algebra techniques. Matrix factorizations occurring in robust control can be reduced to state space problems as well but these involve then Riccati equations and/or eigenvalue problems of of relatively large dimension [9][22][32][15]. In [29] it is shown that various factorization problems may in fact be solved via eigenvalue problems of reduced dimension, and this in a relatively straightforward manner. This idea can be extended to normalized coprime factorizations where current methods still require unreduced Riccati equations to be solved. This ought to yield a new approach which is both faster and numerically more reliable. Similar ideas can also be extended to model reduction of large sparse systems by using sparse matrix techniques to construct approximate projectors leading to lower order models.

2. Sparse system models

Large scale plants typically have system matrices $\{A, B, C, D\}$ with special structure such as sparsity (due to modeling techniques using e.g. finite element methods) or diagonal dominance (inherited from the physical properties of the system being modeled [24], [17]). Constructing reduced order models $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ for such systems requires the solution of eigenvalue problems (e.g. for calculating dominant invariant subspaces), singular value problems (e.g. for calculating principal components) or Lyapunov equations (e.g. for computing Gramians). A very simple approach for model reduction consists in simulating input/output pairs by running the recurrence

$$\begin{bmatrix} x_{i+1} \\ y_i \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} x_i \\ u_i \end{bmatrix}, \qquad (1)$$

and then identifying an approximate system $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$ from the pairs $\{u_i, y_i\}$. This approach is not very appealing for model reduction of dense systems, but for sparse system models it has the great advantage that simulation is cheap and that fast algorithms can be used for the identification step. This is e.g the case when using Hankel type algorithms for the identification step. Moreover, by appropriately choosing the input sequence $\{u_i\}$ one can hope to emphasize particular modes of the system. Notice also that none of the equations involved in the model reduction need be solved *exactly* since the derived model will be an approximation anyway. When a system is given originally in *generalized state space form* $\{E, A, B, C, D\}$ then the inversion of E is needed for the simulation step, but this could be approximated as well. Some bound on the quality of the approximation is needed, though, as a function of the norm of the "residual" of the equations involved. The idea of allowing approximate solutions is of course crucial when using iterative methods for solving the underlying large scale systems with special structure.

3. Robust control

Modern control theory has been largely influenced by the recent development of H_∞ techniques. The underlying theory is now rather well understood but the computational techniques are lagging behind. The design of the robust controller consists in solving factorization problems of transfer matrices and/or related state-space matrix equations [5], [2]. Examples of transfer function factorizations needed here are all-pass factor extraction, inner outer factorization and normalized coprime factorization. Numerical algorithms for such factorizations have been analyzed from a state space and generalized state space point of view [20] [29]. It was found that simple recursive solutions can be obtained for most of them when starting from state space models in so-called condensed forms (such as Schur or generalized Schur forms). The advantage of this approach is a reduced computational complexity and good numerical properties in the corresponding algorithms (see [20], [29] and references therein). One example where we can expect to shortcut currently proposed algorithms

is the construction of normalized coprime factorizations of a given transfer function $R(s)$:

$$\begin{array}{l} R(s) = N(s) \cdot D^{-1}(s) \ , \ with \\ D^*(-\overline{s}) \cdot D(s) + N^*(-\overline{s}) \cdot N(s) = I. \end{array} \tag{2}$$

Presently, one constructs first a stable coprime factorization (with respect to some region Γ), and then one normalizes it via a spectral factorization problem. A more direct approach consists in noticing that $D(s)$ and $N(s)$ must be submatrices of a Γ-stable all-pass $U(s)$ that displays the kernel of $[R(s) \mid -I]$ as follows :

$$\begin{array}{l} \left[\begin{array}{cc} R(s) & -I \end{array} \right] \cdot U(s) \doteq \\ \left[\begin{array}{cc} R(s) & -I \end{array} \right] \cdot \left[\begin{array}{cc} D(s) & \tilde{N}(s) \\ N(s) & \tilde{D}(s) \end{array} \right] = \left[\begin{array}{cc} \tilde{R}(s) & 0 \end{array} \right]. \end{array} \tag{3}$$

This problem now becomes one of rank factorization with all-pass factors and ideas of [29] can be used to tackle this. Notice that this approach avoids breaking down the original problem in seemingly unrelated subproblems.

4. State space projections

A large class of model reduction methods of a system given in state space form $\{A, B, C, D\}$ can be interpreted as performing a similarity transformation T yielding $\{A^{(t)}, B^{(t)}, C^{(t)}, D\}$ $\doteq$ $\{T^{-1}AT, T^{-1}B, CT, D\}$, and then extracting from that the leading diagonal system $\{\hat{A}, \hat{B}, \hat{C}, D\}$, i.e.

$$\left[\begin{array}{c|c} T^{-1}AT & T^{-1}B \\ \hline CT & D \end{array} \right] \doteq \left[\begin{array}{cc|c} A_{1,1}^{(t)} & A_{1,2}^{(t)} & B_1^{(t)} \\ A_{2,1}^{(t)} & A_{2,2}^{(t)} & B_2^{(t)} \\ \hline C_1^{(t)} & C_2^{(t)} & D \end{array} \right] \tag{4}$$

$$\left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] \doteq \left[\begin{array}{c|c} A_{1,1}^{(t)} & B_1^{(t)} \\ \hline C_1^{(t)} & D \end{array} \right].$$

When writing $T \doteq [X_1 \mid X_2]$, $T^{-1} \doteq [Y_1^t \mid Y_2^t]^t$, then $\Pi \doteq X_1 Y_1$ is a projector on X_1 along Y_1, and the subsystem $\{\hat{A}, \hat{B}, \hat{C}, D\}$ is in fact the original system restricted to that projector : $\{\Pi A \Pi, \Pi B, C\Pi, D\}$.

This sounds of course very simple but the crux is to find the projector Π that achieves two important goals. It should be *easy to construct*, i.e. one should be able to exploit sparsity of the model $\{A, B, C, D\}$ for the construction of Π. And the lifted solution should provide a *good approximation* for the true solution of the control problem, i.e. certain "performance bounds" ought to be satisfied. For the first goal there exist a number of iterative techniques such as Krylov type schemes (Lanczos, Arnoldi, GMRES, QMR [23], [11]). The second goal is of course problem dependent and is probably the most challenging one. One may have to assume certain system properties here (such as diagonal dominance) in order to derive sufficiently powerful results.

Such model approximations thus only differ in the choice of projector. Each choice typically tries to achieve some kind of decomposition. Modal approximation e.g. selects "dominant" frequencies in A and performs thus a block diagonal decomposition (13), i.e. with $A_{1,2}^{(t)}$ and $A_{2,1}^{(t)}$ being 0 and $A_{1,1}^{(t)}$ containing the dominant eigenfrequencies. When large scale systems $\{A, B, C, D\}$ are involved one can only afford approximate decompositions, i.e. with $A_{1,2}^{(t)}$ and $A_{2,1}^{(t)}$ small compared to $A^{(t)}$. Iterative techniques for large systems are indeed only applied a limited number of steps for reasons of complexity. Criteria have to be selected to obtain an appropriate system performance and how far iteration techniques have to be applied in order to satisfy corresponding bounds.

5. A projection construction

We give here an example of a projection construction based on the Quasi Minimal Residual (or QMR) algorithm [11]. Consider a $n \times n$ matrix A and two starting vectors $v_1 = b$ and $w_1 = c$. The two sided Lanczos algorithm then essentially constructs column by column matrices $V_m = [v_1 ... v_m]$ and $W_m = [w_1 ... w_m]$ satisfying

$$\begin{array}{l} AV_m = V_m T_m + \delta_{m+1} v_{m+1} e_m^T \\ A^T W_m = W_m T_m^T + \beta_{m+1} w_{m+1} e_m^T \\ V_m^T W_m = I_m, \end{array} \tag{5}$$

where T_m is a tridiagonal matrix of dimension $m \times m$ and V_m and W_m are $n \times m$ dense matrices. The construction of the matrices V_m, W_m and T_m only involves matrix multiplies of the type Av and $A^T w$ and is thus "cheap" when A is sparse and $m << n$. This algorithm may break down (e.g. when the bi-orthonogality condition $V_m^T W_m = I_m$ can not be satisfied) but a modification called the QMR algorithm [11] was recently

proposed that avoids these pitfalls. An interesting result of these "bases" V_m and W_m is that they provide good approximate solutions for the underlying Lyapunov equations

$$\begin{aligned} AP_c + P_c A^T &= bb^T \\ A^T P_o + P_o A &= c^T c. \end{aligned} \tag{6}$$

Based on these approximate solutions for P_o and P_c, one derives a reduced order model that in fact corresponds to a projector $\Pi \doteq V_m W_m^T$. Approximation bounds are linked to approximations of the matrix exponentials $c^T e^{At}$, $e^{At}b$ and $c^T e^{At}b$ [23]. Extensions to the block Lanczos case and various bounds for the resulting lower order approximation are still under investigation.

6. Riccati equations

Several techniques have been proposed for solving algebraic Riccati equations. The very first algorithms were linked to an underlying difference equation and had rather poor convergence properties. Then a class of quadratically convergent methods were proposed [1], [20] and now the most popular methods are based on eigenvalue techniques [26], [20], which also have quadratic convergence but at the same time good numerical stability properties. Several of these methods have been implemented on parallel machines, but hardly any work has been devoted to numerical aspects of methods exploiting special properties like sparsity or diagonal dominance. One could use here again projection ideas as stated above. If $\hat{P}$ is the solution of a Riccati equation connected to a model approximation $\{\hat{A}, \hat{B}, \hat{C}, \hat{D}\}$, what bounds can then be guaranteed for the induced solution in the original coordinate system ? Here again the idea is to use iterative techniques for constructing a projector Π that will give a lower order Riccati equation with a solution $\hat{P}$ which is "close" in some sense to the solution P of the original system. "Closeness" criteria ought to focus on the ultimate design requirement for which the Riccati equation is being solved (e.g. the underlying optimal control problem, spectral factorization or model reduction) rather than on the Riccati equation itself. The choice of projector Π of course also depends on the availability of affordable numerical techniques for constructing it.

Similar ideas can also be applied to optimal control problems for finite horizon giving rise to Riccati differential equations. These can be solved via Ordinary Differential Equations techniques [6] but ODE solvers are for general nonlinear differential equations and do not exploit the fact that the differential equation is of Riccati type. One possible approximation in the linear system case is to solve the corresponding Riccati difference equation which is a nonlinear matrix recurrence of specific type. Exploiting sparsity here is easier because the formulation is closer to matrix operations for which sparse matrix techniques can be used. Again, in order to save computations one ought to approximate solutions using projection ideas for which the simulation does not diverge too much from the true solution. Note that these equations are now being used to solve very large scale problems in very diverse areas [16], [13] besides control.

7. Basic numerical tools

The topics discussed in previous sections all point to the significant role linear algebra problems play in the control, optimization and model reduction of multivariable linear systems. Over the years, numerous algorithms have been developed in that area. Because of the increasing complexity of the problems being tackled, some of them have become challenging from a numerical point of view as well. The interdisciplinary field of numerical linear algebra and linear system theory has lead to some significant developments in the last decade and several of these results are nowadays being implemented in software for CACSD.

However, numerical methods in this area are still far from complete. Most of the techniques available today are specifically aimed at dense systems [31], [20], [33], [30]. The methods developed for special architectures are often only described algorithmically [7], [8], or do not exploit any particular structure or sparsity of the matrices [12], [19]. There are indeed only few methods for large scale control problems that make specific use of the special structure of the matrices involved or that are tailored to efficient use of parallel architectures. The few methods available are typically based on Krylov type techniques [17], [4], [23], [11] or exploit the special structure of the matrices involved [8], [21]. In the particular area of reduced order robust control, most of the ideas are still described for dense

systems [2], [5], [9], and hardly any attention has been given to appropriate numerical techniques for large scale problems.

Although the individual topics discussed above may seem somewhat disparate, they all pertain to the common problem of designing low order robust controllers for large scale plants, and the techniques are closely related. The basic ideas in these techniques are :

(i) interpretation of basic problems in terms of state space models in order to reduce these to linear algebra problems

(ii) exploiting sparsity, diagonal dominance or near decomposition of the matrices involved in order to obtain algorithms of reasonable complexity

(iii) providing appropriate criteria as a measure of "closeness" of the solution constructed via the above techniques, e.g. in order to select the appropriate projector minimizing the corresponding "cost".

These different goals are intertwined in each particular problem, as should be evident from the different topics described above, and in each of them these goals clearly have to be traded against each other.

Acknowledgements

Part of this research was supported by the Research Board of the University of Illinois at Urbana-Champaign (Grant P 1–2–68114) and by the National Science Foundation (Grant CCR 9209349).

References

[1] B.D.O. Anderson, Second-order convergent algorithms for the steady-state Riccati equation, *Int. J. Control*, **28** : 295-306 (1978).

[2] B.D.O. Anderson, Y. Liu, Controller reduction: concepts and approaches, *IEEE Trans. Aut. Contr.* **AC-34** : 802-812 (1989).

[3] H. Bart, I. Gohberg, M. Kaashoek, P. Van Dooren, Factorization of transfer functions, *SIAM J. Contr.* **18** : 675-696 (1980).

[4] D. Boley, G. Golub, The Lanczos-Arnoldi algorithm and controllability, *Systems and Control Letters* **4** : 317-324 (1984).

[5] P. Bongers, O. Bosgra, Controller reduction with closed loop stability margins, *Selected topics in Identification, Modeling and Control* **3** : 35-41 (1991), Delft Univ. Press.

[6] K. Brenan, S. Campbell, L. Petzold, *The Numerical Solution of Initial Value Problems in Differential-Algebraic Equations*, Elsevier Science Publishers, 1989.

[7] J.P. Charlier, M. Vanbegin, P. Van Dooren, Systolic algorithms for digital signal processing and control, in *Advanced Computing Concepts and Techniques in Control Engineering*, NATO ASI Series, Eds. M. Denham and A. Laub, Vol. F47, pp.409-433, Springer-Verlag, 1988.

[8] B.N. Datta, Parallel algorithms in control theory, *Proceedings 30th IEEE Conf. Dec. & Contr.*, Dec. 1991, pp.1700-1704.

[9] J. Doyle, B. Francis and A. Tannenbaum, *Feedback Control Theory*, McMillan, 1992.

[10] A. Edelman, The first annual large dense linear system survey, Int. Rept. Univ. California, Berkeley CA 94720, 1991.

[11] R. Freund, N. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* **60** : 315-339 (1991).

[12] J. Gardiner, *Iterative and parallel algorithms for the solution of algebraic Riccati equations*, Ph. D. Thesis, Univ. California, Santa Barbara, 1988.

[13] M. Ghil, S. Cohn, J. Tavantzis, K. Bube, E. Isaacson, Applications of estimation theory to numerical weather prediction, in *Dynamic Meteorology Data Assimilation Methods*, L. Bengtsson, M. Ghil, E. Kallen (Eds.), Appl. Math. Sc. Vol. 36 (Springer Verlag, Berlin), 1981, pp.139-224.

[14] K. Glover, All optimal Hankel norm approximations of linear time multivariable systems and their L_∞-error bounds, *Int. J. Contr.* **39** : 1115-1193 (1984).

[15] K. Glover, D. Mc Farlane, Robust stabilization of normalized coprime factor plant descriptions with H_∞-bounded uncertainty, *IEEE Trans. Aut. Contr.* **34** : 821-830 (1989).

[16] A. Heemink, Kalman filtering of two-dimensional shallow water flow on a vector processor, *Supercomputer* **20/21** : 32-44 (1987).

[17] S. Hodel, K. Poolla, Numerical solution of very large sparse Lyapunov equations through approximate power iteration, *Proceedings 29th IEEE Conf. Dec. & Contr.* : 291-296 (1990).

[18] P. Kokotovic, W. Perkins, J. Cruz, G. D'Ans, ϵ-coupling method for near-optimum design of large-scale linear systems, *Proceedings of the IEE* **5** : 889-892 (1969).

[19] A. Laub, Numerical linear algebra aspects of control systems design, *IEEE Trans. Aut. Contr.* **30** : 97-108 (1985).

[20] A. Laub, Invariant subspace methods for the numerical solution of Riccati equations, in *The Riccati equation*, Eds. Bittanti, Laub, Willems, Springer Verlag, 1990.

[21] M. Moonen, P. Van Dooren, J. Vandewalle, SVD-updating for tracking slowly time-varying systems. A parallel implementation, in *Signal Processing, Scattering and Operator Theory, and Numerical Methods*, Birkhäuser Boston Inc. 1990, pp. 487-494.

[22] C. Nett, C. Jacobson, M. Balas, A connection between state-space and doubly coprime fractional representations, *IEEE Trans. Aut. Contr.* **29** : 831-832 (1984).

[23] Y. Saad, Analysis of some Krylov subspace approximations to the matrix exponential operator, *SIAM Numer. Anal.* **29** : 209-228 (1992).

[24] R. Srikant, T. Basar, Sequential decomposition and policy iteration schemes for M-player games with partial weak coupling, *Automatica*, to appear, 1992.

[25] M. Steinbuch, J. Terlouw, O. Bosgra, S. Smit, Multivariable application of μ analysis of servo system design in the consumer electronics industry, *IEE Colloquium on Successful Industrial Applications of Multivariable Analysis*, London, 1990.

[26] P. Van Dooren, A generalized eigenvalue approach for solving Riccati equations, *SIAM Sci. & Stat. Comp.* **2** 121-135 (1981).

[27] P. Van Dooren, The generalized eigenstructure problem in linear system theory, *IEEE Trans. Aut. Contr.* **26** : 111-129 (1981).

[28] P. Van Dooren, Numerical aspects of system and control algorithms, *Journal A* **30** : 25-32 (1989).

[29] P. Van Dooren, Rational and polynomial matrix factorizations via recursive pole-zero cancellation, *Linear Algebra & Applications* **137/138** : 663-697 (1990).

[30] P. Van Dooren, M. Verhaegen, On the use of unitary state-space transformations, in *Special Issue of Contemporary Mathematics on Linear Algebra and its Role in Linear System Theory*, AMS, 1985.

[31] A. van den Boom, A. Brown, F. Dumortier, A. Geurts, S. Hammarling, R. Kool, M. Vanbegin, P. Van Dooren, S. Van Huffel, SLICOT, a subroutine library for control and system theory, *IFAC Symp. on CADCS*, Swansea UK, July 1991.

[32] M. Vidyasagar, *Control System Synthesis : a Factorization Approach*, MIT Press, Cambridge, MA, 1985.

[33] R. Walker, C. Gregory, S. Shah, MATRIX$_x$: A data analysis, system identification, control design and simulation package, *Control Systems Magazine* **2:4** 30-37 (1982).