Int. J. Appl. Math. Comput. Sci., 2001, Vol.11, No.5, 1025–1033

NUMERICAL ANALYSIS AND SYSTEMS THEORY[†]

Stephen L. CAMPBELL[*]

The area of numerical analysis interacts with the area of control and systems theory in a number of ways, some of which are widely recognized and some of which are not fully appreciated or understood. This paper will briefly discuss some of these areas of interaction and place the papers in this volume in context.

Keywords: numerical analysis, control, systems theory

1. Introduction

The area of numerical analysis interacts with the area of control and systems theory in a number of ways, some of which are widely recognized and some of which are not fully appreciated or understood. Many of these topics were part of the presentations at the 2000 Mathematical Theory of Networks and Systems Conference held in Perpignan, France. Several of these contributions are included in the papers in this volume.

In this introductory paper we shall briefly discuss some of the many ways in which control, systems theory, and numerical analysis interact. Some of these interactions are well-known but others are not as well appreciated. The discussion will serve not only to give an overview of some of these relationships but also to put the contributions of the rest of the papers in this volume into perspective as part of the growing interaction between these diverse areas. Numerical methods, of course, are involved in almost every real application. While it is vital, we are not interested here in the straightforward use of existing numerical software to solve standard problems in systems and control. Rather, we wish to focus on those situations where issues of a numerical nature need to be investigated and to point out how the papers at MTNS 2000 related to these investigations. Page limitations restrict the number of papers that can be included.

The development of numerical methods for systems modeled by partial differential equations is an important topic which was much evident at MTNS 2000. However, papers on distributed parameter systems presented at the conference will be included in the volume dedicated to Distributed Parameter Systems and Operator Theory in a different issue of this journal. In this volume we will focus on problems related to finite dimensional systems. It is interesting to note that while there have been numerous

[†] Research supported in part by the National Science Foundation under grants DMS–9714811 and DMS–9802259.

[*] Deptartment of Mathematics, North Carolina State University, Raleigh, NC, 27695–8205, U.S.A., e-mail: slc@math.ncsu.edu

books on topics in numerical optimization and some in numerical optimal control, for example, (Alexandrov and Hussaini, 1997; Betts, 2001; Borggaard *et al.*, 1998; Pytlak, 1999), there are relatively few books on numerical analysis and systems and control theory. Exceptions are (Datta, 1999; van Overschee and de Moor, 1996; Pichler *et al.*, 2000). Software environments include MATLAB (Mokhtari and Marie, 2000) and SciLAB (Gomez *et al.*, 1999), to name just two.

2. Carrying Out the Theory

Sometimes the numerical procedures directly carry out the theoretical development of a control problem. In other cases, the numerical algorithms require a theory of their own which is different. An example from linear algebra is the computation of eigenvalues. The theory a software package such as MATLAB actually uses to compute eigenvalues is totally different than the way they are presented and developed in a mathematics or engineering class.

When carrying out an algorithm a number of issues become important, including efficiency, convergence, robustness, accuracy, and scalability. Algorithms that are suitable for one size of a problem may be completely inappropriate for larger problems. Scalability refers to the method being as appropriate for larger more complex problems. Ideally, the computational effort for a scalable algorithm might increase only linearly in the problem size. Robustness refers to the fact that mathematically equivalent algorithms may behave quite differently when implemented due, e.g., to floating point errors. Related to this is how the conditioning of the problem affects algorithmic behavior. There are a number of ways to measure sensitivity or conditioning. One of the most familiar of them is used when solving a linear system $Ax = b$. The condition number is usually taken as the ratio of the largest and smallest singular values of A, $\sigma_M(A)/\sigma_m(A)$.

2.1. Sensitivity and Robustness

One of the most fundamental ideas in systems and control is that of using feedback for pole placement. Suppose that one has the system

$$x' = Ax + Bu, \tag{1a}$$

$$y = Cx, \tag{1b}$$

and suppose that output feedback $u = Ky + v$ is used to give a system

$$x' = (A + BKC)x + Bv.$$

The pole assignment problem is to find an output feedback gain matrix K so that $A + BKC$ has assigned eigenvalues (poles). If $C = I$, the problem is referred to as the full state feedback pole assignment problem.

The paper by Chu (2001) surveys several of the existing numerical methods for the pole assignment problem and then derives new convergence results for some of the optimization-based methods. Chu considers a number of different condition numbers and how minimizing with respect to them affects the algorithms. In particular, he considers the condition number $\|X\|_F^2 - \ln(|\det(X)|)$ where $\|\cdot\|_F$, is the Frobenius matrix norm and X is the matrix of normalized eigenvectors that go with the desired eigenvalues. If X has entries x_{ij}, then $\|X\|_F^2 = \sum_{i=1}^n \sum_{j=1}^m |x_{ij}|^2$.

Often linear systems are initially modeled implicitly so that instead of (1) we have a system in the form

$$Ex' = Ax + Bu, \tag{2a}$$

$$y = Cx + Du. \tag{2b}$$

The transfer function of (2) is then

$$G(s) = D + C(sE - A)^{-1}B. \tag{3}$$

$G(s)$ gives the input-output relationship between the Laplace transforms of y and u ignoring the free response. The matrix E may or may not be singular depending on the application.

Solving equations involving $G(s)$ leads to expressions involving various generalized inverses of $G(s)$. For small systems, these and other operations with $G(s)$ can be carried out by symbolic algorithms. But this becomes impossible as the state dimension increases, and it becomes important to compute these generalized inverses numerically. Note that one needs a numerical algorithm for computing a function of s. Varga (2001) discusses how to do this using robust numerical methods for computing canonical forms of the pencil $\{E, A\}$.

2.2. Large Systems

As the system size increases, additional numerical issues arise. A fundamental computational problem that arises in most algorithms is that of solving a linear system $Ax = b$, often a great number of times. Calvetti *et al.* (2001) discuss the solution to this problem in the case when A is large and ill conditioned, i.e., the rank of A is not known *a priori*. Rather than trying to compute an accurate solution of $Ax = b$, one computes approximate solutions from a better conditioned problem. One application of this is in image processing, and examples from the enhancement of stellar images are given in (Calvetti *et al.*, 2001). In this application the goal is to solve a lower rank problem because it is, in some sense, more accurate. One algorithm is rarely best in all circumstances. Deciding which type of algorithm is best in different circumstances is important. Several of the papers in this volume, including (Calvetti *et al.*, 2001), examine the relative advantages and disadvantages of different algorithms.

Systems and control also raise other important problems where the goal is somewhat different. Often one has a high-dimensional model but it is desired to compute a lower dimensional approximation problem. There are several reasons for doing this. One is because it is much easier to design and implement controllers for low dimensional systems. Also, in real time and model-based control it is often desirable to have low dimensional models that can be easily evaluated and which capture key attributes of larger, more complex problems. The type of approximation depends on what control and systems issues are of importance. This problem of approximation of complex systems by lower dimensional ones was a major numerical topic at MTNS 2000.

Antoulas and Sorensen (2001) give an overview of model reduction algorithms for very large systems. The problem is to approximate the system with state dimension n,

$$x' = Ax + Bu, \tag{4a}$$

$$y = Cx + Du, \tag{4b}$$

by the system with state dimension r,

$$x_r' = A_r x_r + B_r u, \tag{5a}$$

$$y_r = C_r x_r + D_r u, \tag{5b}$$

using algorithms for which

1. approximation error is small and there is a global error bound,
2. system properties of stability and passivity are preserved,
3. the procedure is computationally efficient, and
4. the procedure is automatic and based on error tolerance.

The last item is important. It means that the user needs only to set the tolerance and then the algorithm finds the approximation. In particular, the user does not have to know in advance what the dimension of x_r is. Of course, the goal is to have x_r such that $r \ll n$ whenever possible.

Let $G(s) = D + C(sI - A)^{-1}B$ and $G_r(s) = D_r + C_r(sI - A_r)^{-1}B_r$ be the transfer functions of the original and the reduced models (4) and (5), respectively. Balanced stochastic truncation is an approach to model reduction that tries to minimize $\|G - G_r\|$ for a given value of r. Benner *et al.* (2001) examine efficient numerical algorithms for balanced stochastic truncation.

Li and White (2001) also consider approximation for systems in the form of (1). However, they examine approximations which maintain controllability and observability with well conditioned Grammians. They are especially interested in models from circuits where E and A take the special form of

$$E = \begin{bmatrix} Q & 0 \\ 0 & L \end{bmatrix}, \quad A = \begin{bmatrix} -G & -M \\ M^T & 0 \end{bmatrix}.$$

2.3. Filtering and Related Problems

Another part of control and systems theory where numerical methods are fundamental is filtering and estimation (Golub and Dooren, 1991). Filtering problems have been a major source of problems in numerical linear algebra. The paper by Calvetti *et al.* (2001) applies a numerical procedure to image processing. There are two other applications in this volume.

Solutions to the rational covariance extension problem are parametrized by the spectral zeros. The rational filter with a specified numerator solving the rational covariance extension problem can be determined from a known convex optimization problem. However, the optimization problem may become ill-conditioned for some parameter values. Enqvist (2001) develops a modification of the optimization problem to avoid the ill-conditioning, and the modified problem is solved efficiently by a continuation method.

Many applications involve adaptive filters. The Fast Transversal Filters algorithm is attractive for many adaptive filtering applications except for the fact that it has a tendency to diverge when operating in finite precision arithmetic. Bunch *et al.* (2001) develop and analyze a modification to the Fast Transversal Filters algorithm to give it more robustness.

3. Systems Theory and Its Impact on Numerical Analysis

Up to now we have discussed ways in which numerical analysis can be applied to control problems and some of the challenges this poses for numerical analysis. However, there is another direction of interaction, which is less developed, and that constitutes the application of control ideas to problems in numerical analysis. We shall give two examples where this interaction has gone the other way and stimulated research and applications in numerical analysis.

3.1. Stepsize Control

One of the important problems in numerical analysis is the design and analysis of methods for the numerical integration of differential equations. Suppose that we have a differential equation

$$x' = f(x,t), \quad x(0) = x_0, \tag{6}$$

defined on the interval $[0,T]$. In a typical numerical algorithm for (6) such as a Runge-Kutta method, there are time points $0 = t_0 < t_1 < \cdots < t_K = T$, and stepsizes $h_k = t_{t+1} - t_k$. Let the estimate for $x(t_k)$ be denoted by x_k. The algorithm then proceeds iteratively. Given values $\{x_0, \ldots, x_k\}$, time t_k and h_k, the algorithm computes a new estimate x_{k+1} at time t_{k+1}.

The algorithm defines a new dynamical system whose states are x_k's or, more accurately, x_k's and h_k's. All modern industrial grade codes vary the stepsize h_k to achieve various goals including the stability of the discrete process and bounding the tracking error $e_k = x_k - x(t_k)$, which is called the global error in numerical analysis. Traditionally in the development of numerical codes the stepsize is controlled by a number of *ad hoc* rules, "fudge factors", and heuristics based on estimates of the local error, the stability of the method, and the convergence properties of any iterative solvers used in the implementation of the algorithm. The logic in this stepsize control may become quite complex and is usually based on the experience of the developer.

Notice that there are at least three distinct dynamical systems. One is the original dynamical system (6). The second is the algorithm which defines a discrete dynamical system whose states are (x_k, h_k). The third dynamical system has states which are the error terms e_k. These three systems are interrelated. Reliable calculation of a stable system (6) requires that the error equation be controlled. Desired performance criteria, such as error tolerances, are usually placed on the error terms.

The usual way to try and control the error equation is through varying the stepsize. Variation of the stepsize is a type of feedback control on a dynamical system. It can also be considered as the feedback in a discrete observer of a continuous problem in the presence of noise arising from numerical and function errors.

Generally, numerical analysts are not familiar with control theory. It is natural to ask whether one can use control theory to design better stepsize strategies. One can ask for even more. Can one design control strategies that can be applied across a family of numerical methods? Both of these questions are being investigated.

The application of control theory to stepsize control was investigated in (Gustafsson, 1993; 1994; Gustafsson *et al.*, 1988; Gustafsson and Söderlind, 1997; Hall and Usman, 2001; Söderlind, 1998; Usman and Hall, 1998). When applied to a specific existing method, the resulting codes are generally not noticeably faster. They do seem to be stabler and more robust.

This theory has then been carried forward and developed in the Godess (Generic ODE solving system) project. The use of control theory meant that the stepsize strategy could be separated from the particular discretization or approximation process. In the Godess system the method development, control structure development, code design and implementation are carried out in parallel and the individual parts can be tested and worked on separately. This makes the rapid implementation and testing of ideas for new numerical methods much easier and more reliable.

3.2. Optimal Control Codes

Often when applying the numerical theory to a control problem, the numerical theory guides the implementation of the software. However, sometimes the numerical theory fails to describe correctly what is happening, and a new numerical theory utilizing the fact that a control problem is involved must be used.

One example of this occurs in optimal control. There are several numerical methods for solving optimal control problems. One of them is the direct transcription approach (Betts, 2001). In this method the dynamics are discretized using a discretization from a numerical integrator. The resulting discrete nonlinear programming problem is then solved using an optimization algorithm. The solution is evaluated and, if necessary, the problem is resolved using a finer time discretization.

One such direct transcription code is SOCS (Sparse Optimal Control Software), developed at the Boeing corporation. In working with SOCS it was found that the existing numerical theory did not correctly describe what was occurring when inequality constraints were active at the optimal solution (Betts *et al.*, 2000; Betts *et al.*, 2001; Campbell *et al.*, 2000). When state inequality constraints were active, the optimal solution satisfied a differential algebraic equation, or DAE. DAEs are often called descriptor systems in the systems literature. Associated with a DAE is an integer called its index (Brenan *et al.*, 1996). In some cases these DAEs had index three. The theory for numerical methods for DAEs of index three shows that the discretizations used by SOCS are not convergent when used as an integrator of DAEs index three (Campbell *et al.*, 2000). This was confirmed experimentally on the problems in question. Yet SOCS was able to solve these state constrained optimal control problems with no difficulty. In fact, it was able to solve problems where the active constraints gave index-seven problems and the theory said the discretizations should have been wildly divergent.

This situation is analyzed and explained in (Betts *et al.*, 2000; 2001) using a mixture of control and optimization theory and is an example where the existing numerical theory had to be modified when used in a control setting. In fact, the usual theory of error propagation took on a completely different role. In the standard theory of numerical methods for differential equations, one would have said that a small error early in a calculation could grow dramatically and dominate the calculation. But when combined with control theory, it became a statement that the optimization algorithm could generate small perturbations and that a small perturbation correctly placed enabled the numerical algorithm to overcome the large error predicted by the usual theory and eliminate it from the computation. This was what was observed computationally.

4. Conclusion

There is a growing interaction between numerical analysis and systems and control theory that is enriching both the areas. This interaction was widely evident at MTNS 2000 and will undoubtedly continue to be prominent at future MTNS meetings.

References

Alexandrov N.M. and Hussaini M.Y. (Eds) (1997): *Multi Disciplinary Design Optimization: State of the Art.* — Philadelphia: SIAM.

Antoulas A.C. and Sorensen D.C. (2001): *Approximation in large-scale dynamical systems: An overview.* — this issue.

Benner P., Quintana-Orti E.S. and Quintana-Orti G. (2001): *Efficient numerical algorithms for balanced stochastic truncation.* — this issue.

Betts J.T. (2001): *Practical Methods for Optimal Control Using Nonlinear Programming.* — Philadelphia: SIAM.

Betts J.T., Biehn N., Campbell S.L. and Huffman W.P. (2000): *Convergence of nonconvergent IRK discretizations of optimal control problems.* — Proc. *IMACS 2000*, Lausanne (CD-ROM).

Betts J., Biehn N. and Campbell S.L. (2001): *Convergence of nonconvergent IRK discretizations of optimal control problems with state inequality constraints.* — preprint.

Borggaard J.T., Burns J., Cliff E. and Schreck S. (1998): *Computational Methods in Optimal Design and Control.* — Boston: Birkhäuser.

Brenan K.E., Petzold L.R. and Campbell S.L. (1996): *Numerical Solution of Initial Value Problems in Differential Algebraic Equations.* — Philadelphia: SIAM.

Bunch J.R., Le Borne R.C. and Proudler I.K. (2001): *Measuring and maintaining consistency: A hybrid FTF algorithm.* — this issue.

Calvetti D., Lewis B. and Reichel L. (2001): *On the choice of subspace for iterative methods for linear discrete ill-posed systems.* — this issue.

Campbell S., Biehn N., Jay L. and Westbrook T. (2000): *Some comments on DAE theory for IRK methods and trajectory optimization.* — J. Comp. Appl. Math., Vol.120, No.1–2, pp.109–131.

Chu E.K. (2001): *Optimization and pole assignment in control system design.* — this issue.

Datta B.N. (1999): *Applied and Computational Control, Signals, and Circuits.* — Boston: Birkhäuser.

Enqvist P. (2001): *A homotopy approach to rational covariance extension with degree constraint.* — this issue.

Golub G.H. and Dooren P.V. (Eds.) (1991): *Numerical Linear Algebra, Digital Signal Processing and Parallel Algorithms.* — Berlin: Springer-Verlag.

Gomez C. (Ed.) (1999): *Engineering and Scientific Computing with SciLab.* — Boston: Birkhauser.

Gustafsson K. (1993): *Object-oriented implementation of software for solving ordinary differential equations.* — Sci. Prog., Vol.2, No.4, pp.217–225.

Gustafsson K. (1994): *Control-theoretic techniques for stepsize selection in implicit Runge-Kutta methods.* — ACM Trans. Math. Softw., Vol.20, No.4, pp.496–517.

Gustafsson K., Lundh M. and Söderlind G. (1988): *A PI stepsize control for the numerical integration of ordinary differential equations.* — BIT, Vol.28, No.2, pp.270–287.

Gustafsson K. and Söderlind G. (1997): *Control Strategies for the Iterative Solution of Nonlinear Equations in ODE Solvers.* — SIAM J. Sci. Stat. Comp., Vol.18, No.1, pp.23–40.

Hall G. and Usman A. (2001): *Modified order and stepsize strategies in Adams codes.* — J. Comp. Appl. Math., to appear.

Li J. and White J. (2001): *Reduction of large circuit models via low rank approximate Gramians.* — this issue.

Mokhtari M. and Marie M. (2000): *Engineering applications of MATLAB 5.3 and Simulink 3.* — Springer-Verlag.

van Overschee P. and de Moor B. (1996): *Subspace Identification for Linear System Theory, Implementation, Applications.* — Boston: Kluwer.

Pichler F., Moreno-Diaz R. and Albrecht R. (Eds.) (2000): *Computer Aided Systems Theory—EUROCAST 99.* — Berlin: Springer-Verlag.

Pytlak R. (1999): *Numerical Methods for Optimal Control Problems with State Constraints.* — Berlin: Springer.

Söderlind G. (1998): *The automatic control of numerical integration.* — CWI Quart., Vol.11, No.1, pp.55–74.

Usman A. and Hall G. (1998): *Equilibrium states for predicter-corrector methods.* — J. Comp. Appl. Math., Vol.89, No.2, pp.275–308.

Varga A. (2001): *Computing generalized inverse systems using matrix pencil methods.* — this issue.