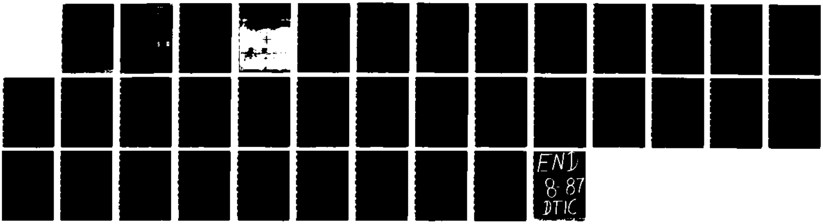
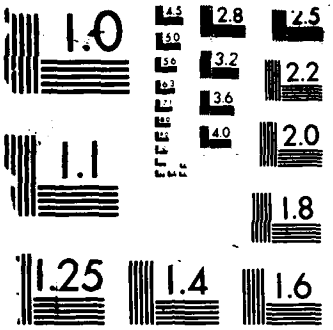
AD-A182 671 A NUMERICAL ALGORITHM FOR OPTIMAL FEEDBACK GAINS IN HIGH DIMENSIONAL LQR ((U) BROWN UNIV PROVIDENCE RI LEFSCHETZ CENTER FOR DYNAMICAL SYSTE H T BANKS ET AL OCT 86 LCDS/CCS-86-47 1/1

UNCLASSIFIED F/G 12/4 NL




1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

(2)

AD-A182 671

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFOSR-TR- 87-0792	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Numerical Algorithm for Optimal Feedback Gains in High Dimensional LQR Problems	5. TYPE OF REPORT & PERIOD COVERED 10-15-85 - 10/27/86 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Banks, H.T. & K. Ito	8. CONTRACT OR GRANT NUMBER(s) AFOSR-84-0398 AFOSR-85-0303
9. PERFORMING ORGANIZATION NAME AND ADDRESS Lefschetz Center for Dynamical Systems Division of Applied Mathematics Brown University, Providence, RI 02912	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F-2304 A1
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bolling Air Force Base Washington, DC 20332 NM	12. REPORT DATE October, 1986 13. NUMBER OF PAGES 29
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Same as 11	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Key Words: LQR problems, feedback gains, distributed parameter systems, computational algorithm, Chandrasekhar system, Newton-Kleinman scheme, Smith

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

INCLUDED - on reverse

87

20. Abstract.

We propose a hybrid method for computing the feedback gains in linear quadratic regulator problems. The method, which combines use of a Chandrasekhar type system with an iteration of the Newton-Kleinman form with variable acceleration parameter Smith schemes, is formulated so as to efficiently compute directly the feedback gains rather than solutions of an associated Riccati equation. The hybrid method is particularly appropriate when used with large dimensional systems such as those arising in approximating infinite dimensional (distributed parameter) control systems (e.g., those governed by delay-differential and partial differential equations). Computational advantages of our proposed algorithm over the standard eigenvector (Potter, Laub-Schur) based techniques are discussed and numerical evidence of the efficacy of our ideas presented.

A NUMERICAL ALGORITHM FOR OPTIMAL FEEDBACK GAINS IN HIGH DIMENSIONAL LQR PROBLEMS

by

H.T. Banks and K. Ito

October 1986 LCDS/CCS #86-47

BROWN
17 64

IN DEO SPERAMUS

Lefschetz Center for Dynamical Systems
and
Center for Control Sciences

A NUMERICAL ALGORITHM FOR OPTIMAL FEEDBACK GAINS IN HIGH DIMENSIONAL LQR PROBLEMS

by

H.T. Banks and K. Ito

October 1986 LCDS/CCS #86-47

A NUMERICAL ALGORITHM FOR OPTIMAL FEEDBACK GAINS IN HIGH DIMENSIONAL LQR PROBLEMS

H.T. BANKS

K. ITO

Center for Control Sciences
Division of Applied Mathematics
Brown University
Providence, RI

October 1986

Research supported in part by the National Aeronautics and Space Administration under NASA grant NAG-1-517, the Air Force Office of Scientific Research under contracts No. AFOSR-84-0398 and AFOSR-85-0303, and the National Science Foundation under NSF grant MCS-8504316. Parts of the research were carried out while the authors were visiting scientists at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665, which is operated under NASA contracts NAS1-17070 and NAS1-18107.

A Numerical Algorithm for Optimal Feedback Gains in High Dimensional LQR Problems

H.T. Banks
K.Ito

ABSTRACT

Authors ~~We~~ propose a hybrid method for computing the feedback gains in linear quadratic regulator (LQR) problems. The method, which combines use of a Chandrasekhar type system with an iteration of the Newton-Kleinman form with variable acceleration parameter Smith schemes, is formulated so as to efficiently compute directly the feedback gains rather than solutions of an associated Riccati equation. The hybrid method is particularly appropriate when used with large dimensional systems such as those arising in approximating infinite dimensional (distributed parameter) control systems (e.g., those governed by delay-differential and partial differential equations). Computational advantages of our proposed algorithm over the standard eigenvector (Potter, Laub-Schur) based techniques are discussed and numerical evidence of the efficacy of our ideas presented.

Key Words: LQR problems, feedback gains, distributed parameter systems, computational algorithm, Chandrasekhar system, Newton-Kleinman scheme, Smith method.

1. Introduction

A great deal of effort in recent years in control of distributed systems has focused on approximation techniques (for example, see [1]-[6], [8], [11]-[14], [16], [18], [21], [22], [24], [26], [32]) to reduce inherently infinite dimensional problems to (large) finite dimensional analogues. Relatively little attention has been given to the development of efficient computational methods for the resulting large but finite dimensional control problems. In this paper we consider such questions for one classical formulation of the feedback control problem, the well-known linear quadratic regulator (LQR) problem.

There are several approaches one can take in such an endeavor. With the emergence of new computer architectures (vector and parallel), one exciting possibility involves the development of new algorithms to be used with nonsequential computers. While we are currently investigating ideas in this direction, our presentation here is to report on some of our efforts to develop better algorithms for use with conventional serial computers.

As is well-known, the LQR problem can be reduced to the solution of a matrix Riccati equation in order to construct the feedback gain matrix. The most widely available method for solution of the Riccati equation is the Potter method [30], which is based on obtaining the eigenvectors and eigenvalues of an associated $2n \times 2n$ Hamiltonian system when the underlying dynamical control system is of dimension n. A related, but improved version involving computation of Schur vectors for the system was proposed by Laub in [27]. While both of these "eigenvector" methods can be used satisfactorily (for a discussion of real advantages offered by the Laub-Schur approach over Potter's method, see [27]) for systems with n relatively small, say $n < 100$, the computational effort (and time) grows like n^3 and becomes prohibitive for large systems. More recently, the idea of using Chandrasekhar systems

([9], [20], [33, p.304-310], [36]) when the number of states is large compared to the number of control inputs (exactly the situation in a number of cases where one approximates a distributed system) has been suggested by a number of authors [33, p.309], [7], [8], [17], [31]. However, as we shall discuss below, there can be numerical difficulties in using the Chandrasekhar approach. On the other hand, it is known that iterative methods such as the Newton algorithm as formulated by Kleinman in [23] can be quite efficiently implemented (even for some large systems) if good initial estimates are provided and if one can solve efficiently the resulting Lyapunov equations. In this presentation, we discuss the formulation and numerical testing of a hybrid method that represents an attempt to combine the good features of the Chandrasekhar approach (growth like n in computational effort) with those of the Newton-Kleinman (quadratic convergence when good initial estimates are provided) along with innovative use of the Smith algorithm for solution of Lyapunov equations.

We expect these ideas to be quite useful in design of control laws for some of the models currently being investigated in connection with large flexible structures as well as in some of the population dispersal and control studies that we are currently pursuing with biologists and ecologists. Some of the large flexible structures involve rather sophisticated distributed parameter models (e.g. see [4], [34]), especially when one wishes to include complicated damping mechanisms involving time or spatially related hysteresis [6], [34] or nonlinear effects [19]. For such models, the computational task. can be rather demanding whether one is carrying out parameter identification [3] or feedback control calculations with traditional eigenvector based methods (the authors of [6] have indicated experiences with runs requiring 9 hours of VAX time when using an approximate system with dimension equivalent to $n = 238$).

For our presentation, we assume that one has used their favorite approximation scheme (finite-elements, spline, spectral, etc.) to reduce the problems of interest to an LQR problem with finite dimensional system. More precisely, throughout our discussions we consider the LQR problem: minimize the cost functional

$$(1.1) \qquad J(u) = \int_0^\infty \{|Cx(t)|^2 + |u(t)|^2\}dt$$

subject to the state dynamics

$$\dot{x}(t) = Ax(t) + Bu(t) , \quad x(0) = x_0 .$$

Here $A \in R^{n\times n}$, $B \in R^{n\times m}$, and $C \in R^{p\times n}$. (We have, without loss of generality, for our discussions here normalized our problem so that the control term in the cost functional (1.1) appears with a weighting matrix I.) We shall assume that (A,B) is stabilizable and (C,A) is detectable [25], [37]. Then the optimal feedback control for the LQR problem involving (1.1) is given by

$$u(t) = -B^TPx(t)$$

where P is the unique non-negative symmetric solution of the algebraic Riccati equation

$$(1.2) \qquad A^TP + PA - PBB^TP + C^TC = 0.$$

In this paper we propose an algorithm which leads to direct calculation of the feedback gain matrix $K = B^TP$ without computation of P. In addition to providing substantial savings in computational time over eigenvector methods, our algorithm requires much less storage and can easily be implemented to take advantage of special structures (e.g. sparsity) in the system matrix A.

To outline the steps in our algorithms, we first recall that the optimal feedback gain K is given by the limit $K = \lim K(t)$ as $t \to -\infty$ of solutions of the Chandrasekhar system [9], [20]

$$\dot{K}(t) = -B^T L^T(t)L(t)\ , \quad K(0) = 0 \tag{1.3}$$
$$\dot{L}(t) = -L(t)(A-BK(t))\ , \ L(0) = C,$$

where $K \in R^{m\times n}$ and $L \in R^{p\times n}$. In fact (see [20], [37]) $P = \lim \int_t^0 L^T(s)L(s)ds$ as $t \to -\infty$. The first step in the proposed hybrid algorithm involves a numerical integration of (1.3) backward in time on an appropriate interval $[-t_f,0]$. For the second step, the value $K(-t_f)$ obtained through this numerical integration of the Chandrasekhar system is then refined by use of the Newton-Kleinman algorithm [23], if we use $K(-t_f)$ as an initial value K_0 for the Newton method.

To motivate our effort in these two steps, we note that the convergence $K(t) \to K$ as $t \to -\infty$ can be very slow when the eigenvalues of A-BK lie close to the imaginary axis. Moreover, $L=0$, K_∞ arbitrary are solutions in the asymptotic limit sense to (1.3). That is, if we denote by $f(K,L)$ the right side of system (1.3), then K_∞ arbitrary and $L=0$ are solutions of $f(K_\infty,L) = 0$. Hence $K(t) \to K_\infty$, $L(t) \to 0$ as $t \to -\infty$ doesn't, in general, have a unique limit numerically. Thus, as is pointed out in [33,p. 316-318], if one is to use the Chandrasekhar approach alone, one needs a very accurate numerical solver for (1.3). This can be computationally quite expensive if we are dealing with a large system and/or a stiff system. Hence, we propose to use a rather crude, fast integration method for the Chandrasekhar component of our algorithm and take the resulting numerical solution $K(-t_f)$ as a start-up value for the Newton iterations. If this crude estimate from the Chandrasekhar step is a sufficiently good initial guess, then we can expect to meet the Newton-Kleinman requirements that $A-BK_0$ be a stability matrix and to obtain quadratic convergence in this second component of the algorithm.

The first step of our hybrid method requires the solution of $n(m+p)$ simultaneous equations, while each iteration of the usual Newton-Kleinman step requires the solution of a Lyapunov equation for the $n\times n$ symmetric estimates of P.

However, as we shall see below, one can use factorization ideas [20] and the Smith method [35] for Lyapunov equations to reformulate the Newton-Kleinman method as a direct iterative method for the m×n gain K, thereby providing additional computational advantages. To speed up our calculations and improve convergence in the Smith algorithm, we propose a variable stepsize Smith method to solve the Lyapunov equations as described in Section 4 below. In Section 2 we outline a numerical scheme for the Chandrasekhar system, while the reformulated Newton-Kleinman iterative procedure to compute directly the gain K is detailed in Section 3. Finally, in Section 5, we discuss further some advantages and disadvantages of the proposed algorithm and report on our experience with several numerical examples to illustrate the feasibility of our hybrid approach.

2. Numerical Solution of the Chandrasekhar System

We return to consider more closely the Chandrasekhar system:

$$(2.1) \qquad \dot{K}(t) = -B^T L^T(t) L(t) \ , \quad K(0) = 0$$

$$(2.2) \qquad \dot{L}(t) = -L(t)(A - BK(t)) \ , \ L(0) = C \ ,$$

where $A \in R^{n \times n}$, $L,C \in R^{p \times n}$, and $K,B^T \in R^{m \times n}$. We first observe that the second equation (2.2) is linear in L. In cases where A arises from a discretization or approximation of partial differential equations, equation (2.2) tends to be a stiff system and thus it is advisable to use an implicit numerical scheme. We propose nere the second order Adams-Moulton algorithm [10, p.235]. A second observation is that the right hand side of equation (2.1) is independent of K and thus an explicit scheme is appropriate; we propose the second order Adams-Bashforth algorithm [10, p.226].

These observations lead us to propose the following algorithm for the Chandrasekhar system (2.1), (2.2): Given a step size h>0, approximations K_i and L_i to K(-ih) and L(-ih) are generated by

$$(2.3) \qquad K_{i+1}^{(0)} = K_i + h(\tfrac{3}{2} B^T L_i^T L_i - \tfrac{1}{2} B^T L_{i-1}^T L_{i-1})$$

$$(2.4) \qquad K_{i+1/2}^{(0)} = (K_{i+1}^{(0)} + K_i)/2$$

$$(2.5) \qquad L_{i+1} = L_i + \frac{h}{2}(L_{i+1} + L_i)(A - BK_{i+1/2}^{(0)})$$

$$(2.6) \qquad K_{i+1} = K_i + \frac{h}{2}(B^T L_{i+1}^T L_{i+1} + B^T L_i^T L_i)$$

where $K_0 = 0$ and $L_{-1} = L_0 = C$.

Several remarks may be useful at this point.

(Remark 1) The stiffness of the matrix A dictates the choice of stepsize h.

(Remark 2) The predicted values $K_{i+1}^{(0)}$ and the corrected values K_{i+1} satisfy

$$K_{i+1} - K_{i+1}^{(0)} = \frac{h}{2}B^T(L_{i+1}^TL_{i+1} - 2L_i^TL_i + L_{i-1}^TL_{i-1})$$
$$\approx \frac{h}{2}\, B^T\, \frac{d^2}{dt^2}\, (L^TL)$$

and this relationship can be used for stepsize refinement, i.e., to give local bounds depending on stepsize which can be used in error control.

(Remark 3) The formula (2.5) can be rewritten as

(2.7)
$$L_{i+1} = L_i(I + \frac{h}{2}\, A_i)(I - \frac{h}{2}\, A_i)^{-1}$$
$$= 2L_i(I - \frac{h}{2}A_i)^{-1} - L_i$$

where $A_i \equiv A - BK_{i+½}^{(0)}$. Defining $H = I - \frac{h}{2}A$ we have that $I - \frac{h}{2}A_i = H + \frac{h}{2}BK_{i+½}^{(0)}$ where $B \in R^{n\times m}$ and $K_{i+½}^{(0)} \in R^{m\times n}$. Thus by the Sherman-Morrison-Woodbury formula [29, p.50] (used frequently when updating an n×n matrix by rank m matrices)

(2.8)
$$(I - \frac{h}{2}A_i)^{-1} = H^{-1} - \frac{h}{2}H^{-1}B(I + \frac{h}{2}K_{i+½}^{(0)}H^{-1}B)^{-1}K_{i+½}^{(0)}H^{-1}$$

where $I + \frac{h}{2}K_{i+½}^{(0)}H^{-1}B \in R^{m\times m}$. The matrix $K_{i+½}^{(0)}H^{-1}$ in (2.8) can be computed by

(2.9)
$$K_{i+½}^{(0)}H^{-1} = K_iH^{-1} + \frac{h}{2}(\frac{3}{2}B^TL_i^TL_iH^{-1} - \frac{1}{2}B^TL_{i-1}^TL_{i-1}H^{-1})$$
$$K_{i+1}H^{-1} = K_iH^{-1} + \frac{h}{2}(B^TL_{1+i}^TL_{1+i}H^{-1} + B^TL_i^TL_iH^{-1}).$$

Hence we see from (2.7) - (2.9) that the step (2.5) only involves the operation L_iH^{-1} plus inversion of an m×m matrix $I + K_{i+½}^{(0)}\, H^{-1}B$. Thus the step (2.5) can be reformulated so that it requires only an m×m matrix inversion plus matrix-vector multiplications if the LU decomposition of $H = I - \frac{h}{2}A$ is computed a priori. This procedure can be most advantageous computationally when m and p are small compared to n.

(Remark 4) For some problems one might wish to use a completely implicit scheme in place of (2.3) - (2.6) to enhance stability and reduce sensitivity to step size choice. Then one might consider iterations $K_{i+1}^{(j)}$, $L_{i+1}^{(j)}$ generated by

$$\begin{aligned}
L^{(j)}_{i+1} &= L_i + \frac{h}{2}(L^{(j)}_{i+1} + L_i)(A-BK^{(j-1)}_{i+\frac{1}{2}}) \\
K^{(j)}_{i+1} &= K_i + \frac{h}{2}(B^T L^{(j)T}_{i+1} L^{(j)}_{i+1} + B^T L_i^T L_i) \\
K^{(j)}_{i+\frac{1}{2}} &= (K^{(j)}_{i+1} + K_i)/2
\end{aligned} \tag{2.10}$$

and thus produce iterates with limits (as $j \to \infty$) K_{i+1}, L_{i+1} satisfying

$$\begin{aligned}
L_{i+1} &= L_i + \frac{h}{2}(L_{i+1} + L_i)(A-B(K_{i+1} + K_i)/2) \\
K_{i+1} &= K_i + \frac{h}{2}(B^T L^T_{i+1} L_{i+1} + B^T L_i^T L_i) \ .
\end{aligned} \tag{2.11}$$

3. An Iterative Method for Computing the Optimal Feedback Gain K

A widely used iterative method for finding the non-negative solution of the algebraic Riccati equation (1.2) is Newton's method as modified by Kleinman [23]. We show that this method can be reformulated so that, when combined with a factored form of the well known Smith method [35], one can compute directly a sequence of iterates K_i for the feedback gain K.

First we recall the Newton iterative algorithm as formulated by Kleinman:

(1) Choose a gain matrix K_0 so that $A\text{-}BK_0$ is a stability matrix;

(2) Update K_i by $K_{i+1} = B^TP_i$ where P_i is the solution of the Lyapunov equation

$$(A\text{-}BK_i)^TP_i + P_i(A\text{-}BK_i) + K_i^TK_i + C^TC = 0.$$

It can be shown that $0 \leq P_{i+1} \leq P_i$ for any i, and $P = \lim P_i$ where the convergence is quadratic. This algorithm can be viewed as an iterative method for the gain K, i.e. $K = \lim K_i$ where $K_{i+1} = F(K_i)$ with $F(K) = B^TP$ and P is the solution of the Lyapunov equation:

$$(A\text{-}BK)^TP + P(A\text{-}BK) + K^TK + C^TC = 0. \tag{3.1}$$

Thus, in order to calculate F(K) one has to solve the Lyapunov equation (3.1) for the symmetric matrix P. However, one can form an alternative version that allows one to directly calculate F(K) using the Smith method for a Lyapunov equation in X of the form

$$S^TX + XS + D^TD = 0 \tag{3.2}$$

where $S \in R^{n\times n}$ is a stability matrix and $D \in R^{p\times n}$.

To this end, we replace step (2) in the Newton-Kleinman method by:

(2') For $i \geq 1$, update K_i by $K_{i+1} = K_i - B^TZ_i$ where $Z_i = P_{i-1} - P_i$ is the solution of the Lyapunov equation

$$(A\text{-}BK_i)^TZ_i + Z_i(A\text{-}BK_i) + D_i^TD_i = 0 \tag{3.3}$$

with $D_i \equiv K_i - K_{i-1}$.

The method with (2') offers several advantages over that using (2). The Lyapunov equation in (2') has fewer inhomogeneous terms than does the one in (2) and the term D_i has rank m which depends only on the number of inputs (controls) to the system. In the proposed modified Smith method described below, one is able to compute directly the m×n update matrix $J^i = B^TZ_i$ without computing Z_i (see (3.12) below). Since $Z_i \to 0$ as $i \to \infty$ is expected, choosing the start-up value $J_0^i = 0$ in the factored Smith algorithm (where $J^i = B^TZ_i$ is computed as the limit as $k \to \infty$ of a sequence J_k^i) is a natural as well as convenient choice.

Note that the step (2') requires that one have $Z_1 = P_0 - P_1$ in hand and hence we must start this procedure with P_0, P_1 (and K_0,K_1) given whereas (2) requires only that one start with K_0 given. Then K_1 is computed by $K_1 = B^TP_0$ with P_0 the solution of

$$(A-BK_0)^TP_0 + P_0(A- BK_0) + K_0^T K_0 + C^TC = 0.$$

Since our Smith algorithm below is formulated to solve Lyapunov equations of the form (3.2), we can, to maintain this form, initially solve the equation twice. That is, if we solve for $\tilde{Z}_0$ the solution of

$$(3.4)\qquad (A-BK_0)^T\tilde{Z}_0 + \tilde{Z}_0(A-BK_0) + K_0^T K_0 = 0$$

and for $\hat{Z}_0$ the solution of

$$(3.5)\qquad (A-BK_0)^T\hat{Z}_0 + \hat{Z}_0(A-BK_0) + C^TC = 0 \quad ,$$

then we can obtain K_1 by $K_1 = B^T\tilde{Z}_0 + B^T\hat{Z}_0$. Since the Smith method as formulated here actually returns B^TX where X is the solution to (3.2), we thus will use this Smith algorithm twice (with $S = A-BK_0$), once with $D^TD = K_0^TK_0$, once with $D^TD = C^TC$ and then simply add the solutions to obtain K_1.

We turn next to the desired factored form of the Smith method as applied to equation (3.2). Let X_0 be an arbitrary n×n symmetric matrix and let a sequence

$\{X_k\}$ of n×n symmetric matrices be generated by

(3.6) $$X_{k+1} = U_r^T X_k U_r + Y_r ,$$

where r is a postive constant (the Smith stepsize) and

(3.7) $$U_r = (I\text{-}rS)^{-1} (I+rS)$$

(3.8) $$Y_r = 2r(I\text{-}rS^T)^{-1} D^T D(I\text{-}rS)^{-1} \quad .$$

Then one can argue that $\{X_k\}$ converges to X, the solution of (3.2). The method and its analysis is based on the observation that for any positive constant r, the equation (3.2) is equivalent to

$$X = U_r^T X U_r + Y_r$$

which can be used to define a contraction map in the obvious manner.

We modify this standard formulation of the Smith method to suit our particular needs here (computing $B^T X$ instead of X). From (3.6) we have

$$X_{k+1} - X_k = U_r^T (X_k - X_{k-1}) U_r \ , \ k \geq 1.$$

Hence, if $X_k - X_{k-1} = M_k^T M_k$ (i.e. if we have a factorable difference), then

$$X_{k+1} - X_k = U_r^T M_k^T M_k U_r = (M_k U_r)^T (M_k U_r).$$

If the start-up value X_0 is zero, then we can write

(3.9) $$X_1 - X_0 = 2rM_1^T M_1 \ , \quad M_1 \equiv D(I\text{-}rS)^{-1} \ .$$

By induction on k, (3.6) is then equivalent to

(3.10) $$M_{k+1} = M_k U_r$$

(3.11) $$X_{k+1} = X_k + 2rM_{k+1}^T M_{k+1} \ .$$

In this manner $B^T X$ can be obtained as the limit of $J_k = B^T X_k$ where J_k is generated by

$$J_{k+1} = J_k + 2rB^T M_{k+1}^T M_{k+1} \ .$$

Thus, the update step (2') is carried out by the following Smith algorithm:

(3.12)(*i*) Set $S_i = A\text{-}BK_i$ and $D = K_i - K_{i-1}$;

(3.12)(*ii*) Choose a positive constant r and form U_r and M_1 by (3.7) and (3.9) with $S = S_i$; put $J_0 = 0$ and $J_1 = J_0 + 2rB^T M_1^T M_1$.

(3.12)(*iii*) Iterate for $k = 1, 2, \ldots,$ on

$$M_{k+1} = M_k U_r$$

$$J_{k+1} = J_k + 2rB^T M_{k+1}^T M_{k+1} \, .$$

In summary, we have described in this section a Newton-Kleinman scheme combined with the Smith method for the resulting Lyapunov equation at each step in the Newton-Kleinman. We have reformulated the Newton-Kleinman iteration and factored the Smith algorithm so as to result in algebraic savings in computing directly the gain estimates K_i.

3. The Smith Method and Variable Stepsize

As is well-known, the rate of convergence in the Smith method discussed in the last section depends upon the choice of the acceleration or step parameter (See [33, p.291-297] for several discussions. Note that our parameter r is the negative reciprocal of the parameter in Russell's discussions). To increase speed in convergence, one may employ the accelerated Smith method [33], [35] which can yield quadratic convergence as compared to the linear convergence obtained with (3.6). However, unlike (3.6), the accelerated Smith method is not self-correcting [33] and here we propose to speed up convergence in an alternative way which has proved both reliable and efficient in some of our numerical tests. Specifically, we propose to use a succession of acceleration parameter values r_i (much in the spirit of other well-known iterative methods such as alternating directions [15], [28]) to accelerate convergence in the basic Smith method. Our formulation of this "variable stepsize" Smith method is based upon the observation that for fixed $r > 0$ and $k \geq 1$, the Smith algorithm can be written as

$$\begin{aligned} S^T X_k + X_k S + D^T D &= E_k \\ E_k &\equiv (I + rS)^T M_k^T M_k (I + rS) \end{aligned} \tag{4.1}$$

where M_k is defined by (3.9) and (3.10). To see this, we note that from (3.6) we have

$$(I - rS)^T X_k (I - rS) = (I + rS)^T X_{k-1} (I + rS) + 2rD^T D \ ,$$

or

$$\begin{aligned} &(I + rS^T) X_k (I + rS) - (I - rS)^T X_k (I - rS) + 2rD^T D \\ &\qquad = (I + rS)^T (X_k - X_{k-1})(I + rS) \ . \end{aligned}$$

Hence, from (3.11) we obtain

$$2r(S^T X_k + X_k S + D^T D) = 2r(I + rS)^T M_k^T M_k (I + rS)$$

which implies (4.1). Moreover, from (3.9) and (3.10) we have

(4.2) $$E_k = (U_r^T)^k D^T D (U_r)^k \quad , \quad k \geq 1 \quad .$$

Thus, if we use the iteration (3.6) with acceleration parameter r_1 for k_1 iterates, we obtain an iterate $X^{(1)}$ and equation error $E^{(1)}$ satisfying

(4.3) $$\begin{aligned} S^T X^{(1)} + X^{(1)} S + D^T D &= E^{(1)} \\ E^{(1)} &= (U_{r_1}^T)^{k_1} D^T D (U_{r_1})^{k_1} . \end{aligned}$$

Let us define the difference $\Sigma^{(1)} = X - X^{(1)}$ where X is the sought-after solution of (3.2). Then it is readily seen that $\Sigma^{(1)}$ satisfies a Lyapunov equation similar to that of (3.2) :

(4.4) $$S^T \Sigma + \Sigma S + E^{(1)} = 0.$$

If we next apply the iteration (3.6) k_2 times with acceleration parameter r_2 to the residual equation (4.4) we obtain

(4.5) $$S^T X^{(2)} + X^{(2)} S + E^{(1)} = E^{(2)}$$

where $X^{(2)}$ is the final iterate using r_2 and the equation error $E^{(2)}$ is given by

$$E^{(2)} = (U_{r_2}^T)^{k_2} E^{(1)} (U_{r_2})^{k_2} .$$

If we proceed to define the difference $\Sigma^{(2)} = X - (X^{(1)} + X^{(2)})$, then from (4.4) and (4.5) we see that $\Sigma^{(2)}$ satisfies a Lyapunov equation

$$S^T \Sigma + \Sigma S + E^{(2)} = 0 .$$

We continue this procedure, using a sequence of acceleration values $\{r_i\}$ along with corresponding iteration counts $\{k_i\}$ to produce a sequence $\{X^{(i)}\}$ of nonnegative, symmetric matrices. For $i \geq 1$, we have

(4.6) $$S^T X^{(i)} + X^{(i)} S + E^{(i-1)} = E^{(i)}$$

(4.7) $$E^{(i)} = (U_{r_i}^T)^{k_i} E^{(i-1)} (U_{r_i})^{k_i} , \; E^{(0)} \equiv D^T D .$$

Thus, if $\bar{X}_j \equiv \sum_{i=1}^{j} X^{(i)}$, then $\bar{X}_j$ satisfies

(4.8) $$S^T \bar{X}_j + \bar{X}_j S + D^T D = E^{(j)}$$

and hence $\bar{X}_j \leq X$ and $\bar{X}_{j-1} \leq \bar{X}_j$, $j \geq 1$.

Using arguments similar to those in [33, p.291-297] one can show that for

$0 < \underline{r} \leq r \leq R$, with $\underline{r}$, R positive constants, there exists a constant ω, $0 < \omega < 1$, depending only on $\underline{r}$ and R, such that for $\omega < \rho < 1$,

$$|U_r^k| \leq M(\rho)\rho^k , \quad k = 0, 1, \ldots ,$$

where $M(\rho)$ is independent of r. Thus, if $\underline{r} \leq r_i \leq R$, then for any $0 < \varepsilon < 1$, there exists an integer $k(\varepsilon)$ such that for $k_i \geq k(\varepsilon)$, $i \geq 1$,

$$|U_{r_i}^{k_i}| \leq 1 - \varepsilon . \tag{4.9}$$

Hence, using (4.7) we have

$$|E^{(j)}| \leq (1-\varepsilon)^{2j} \; |D|^2$$

so that $E^{(j)} \to 0$ as $j \to \infty$ and therefore $\tilde{X}_j \to X$ as $j \to \infty$.

For the hybrid method proposed in this paper, we have combined the variable stepsize method just outlined with the reformulated Smith method of (3.12). We then obtain the following algorithm for solving for the feedback gains K_i:

Algorithm (4.10):

Set $S_i = A - BK_i$, $D_1 = K_i - K_{i-1}$ and $J_0 = 0$. For given acceleration parameters r_1, $r_2, \ldots$, and iteration count indices $k_1, k_2, \ldots$, we iterate on $j = 1, 2, \ldots$, in the following steps:

(4.10a) Compute $U_{r_j} = (I - r_jS_i)^{-1}(I + r_jS_i)$ and

$$M_1 = D_j(I - r_jS_i)^{-1} ,$$

$$J_1 = J_0 + 2r_jB^TM_1^TM_1$$

(4.10b) Iterate for $k = 1, 2, \ldots, k_j - 1$ in

$$M_{k+1} = M_kU_{r_j}$$

$$J_{k+1} = J_k + 2r_jB^TM_{k+1}^TM_{k+1}$$

(4.10c) Compute $D_{j+1} = M_{k_j}(I + r_jS_i)$, set $J_0 = J_{k_j}$ and return to (4.10a) with $j = j + 1$.

The steps in Algorithm (4.10) are to be repeated, i.e. iteration through $r_1, r_2, \ldots$, until some convergence criterion is met. In performing the steps in (4.10b), one can use the Sherman-Morrison-Woodbury formula in a procedure such as that outlined in (2.7) - (2.9) in section 2.

As is often the case in variable stepsize algorithms, the choice of the acceleration parameters $r_1, r_2, \ldots$, and the associated iteration counts $k_1, k_2, \ldots$, provides both freedom and some frustration in the search for "best" choices. If one follows the guide provided by ADI methods (see [15, p.37]), one might choose a set of values r_j to be used in some cyclic order. The best choices of values for the r_j often depend on the eigenvalues of $S_i = A - BK_i$. For example, consider the case where S_i has only real eigenvalues λ_j, each with multiplicity m_j, $j = 1, 2, \ldots, m$. Then a choice of $r_j = -1/\lambda_j$ and $k_j = m_j$ in the algorithm produces convergence in a finite number (m) of steps. That is, this choice yields $E^{(m)} = 0$ in (4.8).

Of course, the complete eigenstructure of S_i will not be known (nor do we suggest that any sophisticated analysis along these lines be included with each use of Algorithm (4.10) to obtain the gains K_i). A possible alternative is to use one of the polynomial acceleration methods [15, Chp 3, 4].

In closing this section, we note that the analogies of our variable step Smith method with the ADI methods used to solve partial differential equations can be made a little more precise. Briefly, in ADI spliting methods [28, p.146-148], one attempts to solve a discretization of the evolution equation

$$\dot{\Phi} = A\Phi + f$$

when $A \gg 0$ can be written $A = A_1 + A_2$ with $A_i \gg 0$ (for example, factored into components corresponding to spatial discretizations in the x and y directions respectively for an equation in a two dimensional spatial domain). This can be shown [28, p.150] to lead to an iterative scheme

$$(I + \frac{h}{2}A_1)(I + \frac{h}{2}A_2)\Phi^{j+1} = (I - \frac{h}{2}A_1)(I - \frac{h}{2}A_2)\Phi^j + hf^j \tag{4.11}$$

where the index j is related to time stepping. On the other hand, if one considers the Smith method (3.6)-(3.8) for

$$S^TX + XS = F$$

and chooses $r = \frac{h}{2}$, one obtains the iteration

$$(I - \frac{h}{2}S^T)X^{j+1}(I - \frac{h}{2}S) = (I + \frac{h}{2}S^T)X^j(I + \frac{h}{2}S) + hF\ . \tag{4.12}$$

In these iterations one may identify the n×n matrix $X = [x_1, \dots, x_n]$, $x_i \in R^n$ and the n^2 vector Φ = column $[x_1, \dots, x_n]$. If we then identify $A_1\Phi$ with $-S^TX$ (i.e., $A_1 = -\ I\otimes S^T$) and $A_2\Phi$ with $-XS$ (i.e. $A_2 = -S\otimes I$), we can immediately see the equivalence between (4.11) and (4.12).

5. Summary Remarks and Numerical Examples

In the preceeding sections we have presented an algorithm which offers some definite advantages in computing directly the feedback gains K for high dimensional LQR problems such as those arising in approximating partial or delay differential equation control problems. As we shall see with several numerical examples in this section, it can substantially outperform standard eigenvector methods on such problems. As we have pointed out, a fundamental algebraic operation (in both the Chandrasekhar update (2.6), (2.7) and in the reformulated Smith methods (4.10b)) involves computation of

$$(5.1) \qquad L(I - r(A - BK))^{-1}$$

where L and K are p×n and m×n matrices respectively. Our algorithm uses the Sherman-Morrison-Woodbury formula which can provide significant computational savings when m and p are small compared to n. For systems involving sparse matrices A (a frequent occurence in many approximation schemes), the needed calculations can be carried out quite efficiently.

We further note that the Chandrasekhar and Newton-Kleinman-Smith components as formulated in our algorithm lead to ready estimates between the true gain K and the iterates K_i in terms of equation errors in the steps being performed.

One component (the variable step Smith) of the algorithm is most effectively carried out if one possesses some a priori knowledge of bounds on the closed loop eigenvalues. If the closed loop eigenvalues lie close to the imaginary axis, then convergence in the Smith method can be very slow. Eigen or Schur vector methods [27], [30] are less sensitive in this regard. For low order systems, the Schur vector approach is more reliable and less expensive computationally than our algorithm. Our hybrid algorithm depends critically on a number of choices (e.g., stopping criteria in the Newton-Kleinman and Smith components, stepsize sequence $\{r_j\}$ and iteration count sequence $\{k_j\}$ in the variable step component) to be made by the user and the "best" choices are heavily problem dependent. Hence one can expect our hybrid algorithm to require more experimentation and fine tuning than other more standard methods. However, as we shall demonstrate with examples, for the case where n is large compared to m and p, it can offer considerable computational savings with no loss in accuracy over the methods mentioned above.

We have tested (and are continuing our efforts in this direction) our hybrid algorithm on several numerical examples. We shall report on just two of these here to illustrate our findings. All our computations were carried out in double precision on an IBM 3081 at Brown University. We gratefully acknowledge the assistance of Yun Wang in our carrying out of the extensive computational studies reported for the boundary flux control in the diffusion equation problem of Example 2 below.

Example 1: As one of our examples, we considered an example (Example 6 of [27]) which Laub used to test his Schur based methods. The system is the n-dimensional system of (1.1) with

$$A = \begin{bmatrix} 0 & 1 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & & & & 0 \\ . & & & & & & \\ . & & & & & & \\ . & & & & & 0 & 1 \\ 0 & . & . & . & . & . & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ . \\ . \\ . \\ 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & . & . & . & 0 \end{bmatrix}$$

which leads to an ill conditioned Riccati equation. This problem corresponds to one in which n integrators are connected in series with a feedback controller to be applied to the nth integrator in order to stabilize the system. Only deviations of x_1 from the origin are penalized in the cost functional. The true optimal gain is an n vector $K = (\bar{K}^1, \ldots, \bar{K}^n)$ and for this example one can argue that $\bar{K}^1 = 1$. In [27], Laub used his Schur techniques to study this example and reported difficulties with loss of accuracy at a relatively low value of n, n = 21. We carried out runs with our hybrid algorithm and obtained quite favorable performance. Some of our findings included:

(a) For n = 40, we used the Chandrasekhar component to integrate to $t_f = 100$ and produce an initial estimate $K_0^1 = .99041$, which when used in the Newton-Kleinman (fixed step size r = .5 in the Smith) produces the estimate $K_5^1 = 1.0$ - in a total of 2.93 seconds of CPU time. When we used a cruder solution in the Chandrasekhar component ($t_f = 200$ but with step size twice that in the first run) to produce $K_0^1 =$ .9394, followed by the N - K ($r_1 = .5$, $r_2 = 1.0$ in the variable step Smith) we obtained $K_6^1 = 1.0$ -, all in 2.39 seconds.

(b) For n = 50, we produced $K_0^1 = .9224745$ at $t_f = 220$ and after the N-K-Smith (fixed step r = .5 in the Smith) obtained $K_5^1 = 1.0000000003820$ in a total of 4.44 CPU seconds. For the same runs with variable step ($r_1 = .5$, $r_2 = .7$) Smith we obtained a K_5^1 as above with 3820 replaced by 3817 in a total of 4.31 seconds.

(c) We compared runs with the Chandrasekhar component only against the Potter method for n = 10, 21, 40. Obtaining essentially the same estimates for n = 10 and 21 (at n = 40, the Potter degenerates numerically to produce useless estimates) we had CPU times of $CH_{n=10} = .753$ seconds, $POTT_{n=10} = .188$ seconds, $CH_{n=21} = 1.52$ seconds, $POTT_{n=21} = 1.22$ seconds, $CH_{n=40} = 4.35$ seconds, $POTT_{n=40} = 6.81$ seconds.

We found for this example that the eigenvector methods are best for small n, but as n grows, the Chandrasekhar alone, and, even more so, the hybrid method will out perform the eigen-Schur methods in both accuracy and CPU times. A more striking demonstration of this behavior will be given in the next example.

Example 2: We consider the linear quadratic regulator problem: minimize the cost functional

(5.2) $$J(u) = \int_0^\infty (|Cz(t)|^2 + |u(t)|^2)dt$$

subject to the partial differential equation

(5.3) $$\frac{\partial}{\partial t}z(t,x) = \frac{\partial^2}{\partial x^2} z(t,x) \ , \ x \in (0,1)$$

$$z(0,x) = \Phi(x)$$

with boundary conditions

(5.4) $$\frac{\partial}{\partial x} z(t,0) = u(t) \quad \text{and} \quad \frac{\partial}{\partial x} z(t,1) = 0$$

where $c(\cdot)$ is square integrable on [0,1] and

$$Cz(t) = \int_0^1 c(x) \ z(t,x) \ dx$$

We can discretize or approximate (5.3)-(5.4) using the standard Galerkin method [2]; i.e. the approximating solution $z^N(t,x)$ to (5.3)-(5.4) is given by

(5.5) $$z^N(t,x) = \sum_{i=0}^{N} w_i(t)\ell_i(x) \ , \ w_i(t) \in R^1,$$

where $\ell_i = \ell_i^N$ *is the first order spline defined by*

$$\ell_i^N(x) = \begin{cases} N(x - \frac{i-1}{N}), & \frac{(i-1)}{N} \leq x \leq \frac{i}{N} \\ N(\frac{i+1}{N} - x), & \frac{i}{N} \leq x \leq \frac{(i+1)}{N} \\ 0 & \text{otherwise} \end{cases}$$

and $z^N(t,x)$ satisfies

(5.6) $$\int_0^1 \frac{\partial}{\partial t} z^N(t,x)\psi^N(x)dx = -\int_0^1 \frac{\partial}{\partial x} z^N \frac{\partial}{\partial x}\psi^N \ dx - u(t)\psi^N(0)$$

$$\text{for all } \psi^N \in Z^N = \text{span } \{\ell_0^N, \ell_1^N, \ldots, \ell_N^N\} .$$

Then, (5.6) leads to the nth order ($n = N + 1$) ordinary differential equation for $w^N = \text{col}(w_0, \ldots, w_N)$;

(5.7) $$Q^N\dot{w}^N(t) = -H^N w^N(t) - B^N u(t)$$

where

$$Q^N = \frac{1}{N}\begin{bmatrix} 1/3 & 1/6 & 0 & \cdot & \cdot & 0 \\ 1/6 & 2/3 & 1/6 & & & \cdot \\ 0 & \cdot & \cdot & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & 0 \\ \cdot & & & 1/6 & 2/3 & 1/6 \\ 0 & \cdot & \cdot & 0 & 1/6 & 1/3 \end{bmatrix} \quad \text{with } Q^N_{ij} = \int_0^1 \ell_i \ell_j dx \ ,$$

$$H^N = N\begin{bmatrix} 1 & -1 & 0 & \cdot & \cdot & 0 \\ -1 & 2 & -1 & & & \cdot \\ 0 & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & 0 \\ \cdot & & & -1 & 2 & -1 \\ 0 & \cdot & \cdot & 0 & -1 & 1 \end{bmatrix} \quad \text{with } H^N_{ij} = \int_0^1 \frac{d}{dx} \ell_i \frac{d}{dx} \ell_j dx \ ,$$

and

$$B^N = \text{col } (1 \ 0 \ . \ . \ . \ 0) \ .$$

For computational convenience, we change coordinates (for fixed N) in the system (5.7) by $x = Q^N w^N$ to obtain the approximate system

$$\dot{x} = -H^N(Q^N)^{-1}x - B^N u \quad .$$

Thus, in (1.1) we have $A = -H^N(Q^N)^{-1}$, $B = -B^N$ and $C = C^N(Q^N)^{-1}$ where C^N is the vector with components $c_i^N = \int_0^1 c(x)\ell_i(x)dx$, $0 \leq i \leq N$.

For the problem in this example, the approximating optimal feedback operator K^N is given [2] by:

$$K^N z = \int_0^1 k^N(x)z(x)dx$$

where $k^N(x) = \sum_{i=1}^{N} k_i \ell_i(x)$ and $K = (k_0, \ . \ . \ . \ , k_N)$ is the optimal feedback solution in the problem for (1.1) with A, B, C chosen as indicated above. We note in this case that for any $N \geq 1$, A has only one unstable eigenvalue (zero), (A,B) is stabilizable, and (A,C) is detectable.

For the special case when $c(x) = 1$, we find $C = (1, \ . \ . \ . \ , 1) \in R^{1\times(N+1)}$ and hence $CA = 0$. It is thus easy to see that the desired solution (K(t),L(t)) to the Chandrasekhar system (1.3) is given by

$$K(t) = k(t)C\ ,\quad L(t) = \ell(t)C,$$

where k,ℓ are scalar functions satisfying

$$\dot{k} = \ell^2\ ,\quad k(0) = 0$$

$$\dot{\ell} = -\ell k\ ,\quad \ell(0) = 1\ .$$

Therefore we find $k\dot{k} + \ell\dot{\ell} = 0$ so that $k^2(t) + \ell^2(t) = 1$. We thus find in this case that $k(t) \to 1$ as $t \to -\infty$ and hence $K = \lim K(t) = C$. For this case, the Chandrasekhar system for the infinite dimensional LQR problem (5.2)-(5.4) can also be analyzed [17], [36] and exactly the same argument as above shows the optimal feedback gain operator is given by

$$\mathcal{K}z = \int_0^1 1\cdot z(x)dx\ .$$

These analytic solutions can be used to test software packages and approximation schemes before more interesting, analytically intractable examples are considered.

Remark: The form (5.7) of system equations appears frequently in applications. Thus the critical computational factor (5.1) can be modified so that one can avoid computing A. For example, in this case it has the form

$$\begin{aligned} &L(I - r\,(-HQ^{-1} - BK))^{-1} \\ &\qquad = LQ(Q + rH + rBKQ)^{-1} \end{aligned} \tag{5.8}$$

where Q + rH is a symmetric, tridiagonal, positive matrix. Thus one can readily use the Cholesky decomposition algorithm for computing $LQ(Q + rH)^{-1}$ and combine this with the Sherman-Morrison-Woodbury formula (see Remark 3 of Section 2) to efficiently compute the critical expression (5.8).

We carried out extensive computations for this example with c(x) = 1 + x. We compared our hybrid method to the Potter algorithm and to use of the Chandrasekhar system alone. We have not used the Laub-Schur method on this example since we felt comparison with a readily available (to us) Potter package would give as a feel for the relative advantages and disadvantages of our scheme compared to eigen and Schur vector based techniques. (Analysis and computational experience indicate that the Potter method and the Laub-Schur method are both $O(N^3)$ with the latter method about twice as fast as the Potter method.) We required, whenever feasible, the same level of accuracy in computation of feedback gains and compared relative CPU times.

In studying our hybrid scheme, we tested numerous sets of Smith acceleration parameters $\{r_j\}$, $\{k_j\}$, stopping times t_f in the Chandrasekhar component and error stopping criteria in both the Chandrasekhar and Newton-Kleinman-Smith components. We summarize some of our findings to date.

In Table 5.1 we present comparative CPU times for the hybrid scheme vs. Potter as we increase N. Recall the corresponding finite dimensional approximation scheme has system with dimension $n = N+1$. In all of the runs reported in Table 5.1, the feedback gains for the hybrid and Potter calculations agreed to 9 decimal places so both scemes provided accurate solutions. In these runs, the hybrid scheme calculations used $t_f = 2.2$ (corresponding to $h = .1$) with $|L(-t_f)| = 10^{-3}$ in the Chandrasekhar component. The Newton-Kleinman component converged after 4 iterations (i.e. at K_4) and we used acceleration steps $r_1 = 1$, $r_2 = 10^{-1}$, $r_3 = 10^{-3}$, $r_4 = 10^{-5}$. Each Smith iteration was allowed a maximum of $k_j = 50$ per value of r_j

N	Hybrid (CPU Sec.)	Potter (CPU Sec.)
10	.17	.14
20	.31	.81
30	.56	2.45
40	.74	5.49
50	.91	10.71
60	1.09	18.09
70	1.26	27.97
80	1.43	41.56
100	1.76	
120	2.10	
140	2.45	
160	2.80	

Table 5.1

although in most cases the iteration satisfied a convergence criterion before this maximum was attained. Careful consideration of Table 5.1 reveals that the hybrid scheme is clearly O(N) while the Potter is $O(N^3)$; both rates are to be expected from our earlier observations about the methods. Note that at N = 80 the hybrid scheme is more than 25 times faster than the Potter scheme (with comparable accuracy, of course).

We also ran the hybrid scheme with N = 80 and a number of different fixed acceleration values r in the Smith component. The same Chandrasekhar component prameters as reported above were used. Table 5.2 contains relative CPU times as well as an indication of the N-K iterate for which convergence was achieved.

In Table 5.3 we list some CPU times when different sets of acceleration parameters $\{r_j\}$ were used. Again these runs were for N = 80 with the same Chandrasekhar solution as above. All of the converged Newton-Kleinman iterates were after 6 steps (i.e. K_6).

r	CPU (Sec.)	Converged N-K Gain
5	5.10	K_6
1	6.52	K_4
10^{-1}	6.27	K_4
10^{-2}	7.40	K_4
10^{-3}	10.10	K_4
10^{-4}	12.06	K_5
10^{-5}	10.07	K_4

Table 5.2

r	CPU (Sec.)
$(10^{-1}, 10^{-2})$	6.25
$(1, 10^{-1}, 10^{-2})$	4.88
$(1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5})$	1.98
$(1, 10^{-1}, 10^{-3}, 10^{-5}, 10^{-6}, 10^{-7})$	1.61

Table 5.3

Finally, we made runs (for N = 80) to find the best results that the

Chandrasekhar algorithm alone (i.e. accurate integration until $K(t) \to K$, $L(t) \to 0$) could produce. The best results we were able to achieve yielded an accurate value of K for $K(-t_f)$ with $t_f = 3.22$ with $|L(t_f)| \approx 10^{-6}$ obtained in 5.85 CPU seconds.

Based on our computational findings for the above two examples and our experience with several other examples for infinite dimensional systems (e.g., beams with tip bodies, etc.), we are quite confident that the hybrid scheme we propose in this paper can be profitably used with a number of large scale LQR problems. We are currently developing a rather general software package that implements the hybrid scheme in a manner so that a broad range of problems can be treated in the context of the ideas presented here.

REFERENCES

[1] H.T. Banks and J.A. Burns, Hereditary control problems: numerical methods on averaging approximations, SIAM J. Control Opt., 16(1978), 169-208.

[2] H.T. Banks and K. Kunisch. The linear regulator problem for parabolic systems, SIAM J. Control and Opt., 22(1984), 684-698.

[3] H.T. Banks and I.G. Rosen, Computational methods for the identification of spatially varying stiffness and damping in beams, LCDS Tech. Rep. 86-39, Brown University, August, 1986; Control: Theory and Advanced Technology, submitted.

[4] H.T. Banks and I.G. Rosen, Parameter estimation techniques for distributed systems, in Control and Estimation in Distributed Parameter Systems, SIAM Frontiers in Applied Math, to appear.

[5] H.T. Banks, I.G. Rosen and K. Ito, A spline based technique for computing Riccati operators and feedback controls in regulator problems for delay equations, SIAM J. Sci. Stat. Comp., 5 (1984), 830-855.

[6] J.A. Burns and R.H. Fabiano, Modeling and approximation for a viscoelastic control problem, Proc. Conf. Control of Partial Differential Equations (July 7-11, 1986; Vorau), Springer, to appear.

[7] J.A. Burns, K. Ito and R.K. Powers, Chandrasekhar equations and computational algorithms for distributed parameter systems, Proc. 23rd IEEE Conference on Decision and Control. Dec., 1984, Las Vegas, Nevada.

[8] J.A. Burns and R. Powers, Factorization and reduction methods for optimal control of hereditary systems, Mat. Apl. Comput., to appear.

[9] J. Casti, Dynamical Systems and Their Applications: Linear Theory, Academic Press, New York, 1977.

[10] S.D. Conte, Elementary Numerical Analysis, MCGraw Hill, New York, 1965.

[11] J.S. Gibson, An analysis of optimal modal regulation: convergence and stability, SIAM J. Control Opt. 19 (1981), 686-707.

[12] J.S. Gibson, Linear-quadratic optimal control of hereditary differential systems: infinite dimensional Riccati equations and numerical approximations, SIAM J. Control Opt., 21(1983), 95-139.

[13] J.S. Gibson and A. Adamian, Approximation theory for LQG optimal control of flexible structures, SIAM J. Control Opt. submitted.

[14] J.S. Gibson and A. Adamian, A comparison of three approximation schemes for optimal control of a flexible structure, in Control and Estimation in Distributed Parameter Systems, SIAM Frontiers in Applied Math, to appear.

[15] L.A. Hageman and D.M. Young, Applied Iterative Methods, Academic Press, New York, 1981.

[16] K. Ito, Legendre-tau approximation for functional differential equations III. Eigenvalue approximations and uniform stability, in Distributed Parameter Systems (F. Kappel, et.al., eds.) Lec. Notes in Control and Info. Sci. 75 Springer-Verlag, 1985, p. 191-212.

[17] K. Ito and R.K. Powers, Chandrasekhar equations for infinite dimensional systems, ICASE Report 84-13, NASA Langley Research Center, Hampton, VA., SIAM J. Control Opt, to appear.

[18] K.Ito and R.Teglas, Legendre-tau approximation for functional differential equations II. The linear quadratic optimal control problem, SIAM J. Control Opt., to appear.

[19] J-N Juang and L.G. Horta, Effects of atmosphere on slewing control of a flexible structure, AIAA/ASME/ASCE/AHS 27th Structures, Structural Dynamics, and Materials Conf. (May 19-21, 1986; San Antonio), Paper No. 86-1001-CP.

[20] T. Kailath, Some Chandrasekhar-type algorithm for quadratic regulators, Proc. IEEE Conference on Decision and Control. New Orleans, 1972, 219-223.

[21] F. Kappel and G. Propst, Approximation of feedback controls for delay systems using Legendre polynomials, Conf. Sem. Mat. Univ. Bari 201 (1984), 1-36.

[22] F. Kappel and D. Salamon, Spline approximation for retarded systems and the Riccati systems, SIAM J. Control Opt., to appear.

[23] D.L. Kleinman, On an iterative technique for Riccati equations computations, IEEE Trans. Automat. Contr., AC-13(1968), 114-115.

[24] K. Kunisch, Approximation schemes for the linear quadratic optimal control problem associated with delay equations, SIAM J. Control Opt. 20 (1982), 506-540.

[25] H. Kwakernaak and R. Sivan, Linear Optimal Control Systems, Wiley-Interscience, New York, 1972.

[26] I. Lasiecka and R. Triggiani, The regulator problem for parabolic equations with Dirichlet boundary control II: Galerkin approximation, preprint.

[27] A.J.Laub, A Schur method for solving algebraic Riccati equations, IEEE Trans. Automat. Contr., AC-24(1979), 913-921.

[28] G.I. Marchuk, Methods of Numerical Mathematics, Springer-Verlag, New York, 1975.

[29] J. Ortega and W. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[30] J.E. Potter. Matrix quadratic solutions, SIAM J. Appl. Math., 14(1966), 496-501.

[31] R.K. Powers, Chandrasekhar Equations for Distributed Parameter Systems, Ph.D. Thesis, Virginia Polytechnic Institute and State Unviersity, Blacksburg, VA., 1984.

[32] G. Propst, Piecewise linear approximation for hereditary control problems, Tech Rep. No. 60-1985, Inst. for Math, Univ. Graz, Graz, Austria.

[33] D.L. Russell, Mathematics of Finite Dimensional Control Systems: Theory and Design, Lecture Notes in Pure and Applied Math., vol. 43, Dekker, New York, 1979.

[34] D.L. Russell, On mathematical models for the elastic beam with frequency-proportional damping, in Control and Estimation in Distributed Parameter Systems, SIAM Frontiers in Applied Math., to appear.

[35] R.A. Smith, Matrix equation XA + BX = C, SIAM J. Appl Math., 16(1968), 198-201.

[36] M. Sorine, Sur le semi-groupe non lineaire associe a l'equation de Riccati, INRIA Report No. 167, October 1982.

[37] W.M. Wonham, On a matrix Riccati equation of stochastic control, SIAM J. Control, 6(1968), 681-699.

END

8-87

DTIC