Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

An Efficient Long-Time Integrator for Chandrasekhar Equations

Jeff Borggaard and Miroslav Stoyanov

***Abstract*— A drawback of using Chandrasekhar equations for regulator problems is the need to perform long-time integration of these equations to reach a steady state. Since the equations are stiff, this long-time integration frequently defeats the computational advantages the Chandrasekhar equations have over solving the algebraic Riccati equations. In this paper, we present a strategy for approximating the long-time behavior of the Chandrasekhar equations. Our approach leverages recent developments in building accurate, empirical, reduced-order models for high-order systems. The aim here is to build a reduced-order model for the Chandrasekhar equations that is accurate near the steady state gain. We then assemble a corresponding low-dimensional Riccati equation that can be solved easily. For this study, we use the proper orthogonal decomposition (POD) to generate the reduced-order model. A heuristic for building a suitable input collection for POD is proposed. Numerical experiments using a 2D advection-diffusion-reaction (ADR) equation demonstrate the computational feasibility of our approach.**

I. INTRODUCTION

We consider control problems of the form

$$\min_{u(\cdot)} \int_0^\infty \{\langle y(t), y(t)\rangle + \langle u(t), Ru(t)\rangle\} \; dt \qquad (1)$$

subject to

$$\begin{aligned} \dot{z}(t) &= Az(t) + Bu(t), \qquad z(0) = z_0 && (2)\\ y(t) &= Cz(t), && (3) \end{aligned}$$

where $z(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$, for $t > 0$ and matrices A, B, C, and R have appropriate dimensions. In this study, we further assume $m \ll n$ and $p \ll n$. Problems with this structure frequently arise in the numerical discretization of distributed parameter control problems.

The solution to the regulator problem is linear state feedback

$$u(t) = -Kz(t), \qquad t > 0, \qquad (4)$$

where the feedback gain K is defined using the unique positive definite solution, Π, to the algebraic Riccati equation

$$A^T\Pi + \Pi A - \Pi BR^{-1}B^T\Pi + C^TC = 0 \qquad (5)$$

as $K = R^{-1}B^T\Pi$ (see e.g. [1] for details). For the problem dimensions under consideration here, an $n \times n$ matrix Riccati solution Π is used to form the $m \times n$ matrix K. Thus, large values of n require us to exploit the structure in the problem. This can be done in the algorithm to solve the Riccati equation (eg. exploiting sparsity in A, B and C [2] or exploiting the low rank of B and C) or one may use the Chandrasekhar equations discussed below.

A. Chandrasekhar Equations

The Chandrasekhar equations were introduced in the control literature in the 1970's to solve the Riccati equations (cf. [3], [4], [5]). In the 1980's, they were used as a methodology for solving the infinite horizon control problem when the system involves a distributed parameter system (cf. [6], [7]). In the infinite horizon/regulator case, the Chandrasekhar equations have the form

$$\begin{aligned} -\dot{K}(t) &= R^{-1}B^TL(t)L^T(t), \quad K(0) = 0 \in \mathbb{R}^{m\times n} && (6)\\ -\dot{L}(t) &= (A - BK(t))^T L(t), \;\; L(0) = C^T \in \mathbb{R}^{n\times p}. && (7) \end{aligned}$$

The solution to the control problem (1)–(4) is then given by

$$K = \lim_{t\to-\infty} K(t).$$

This approach replaces the need to find the (dense) $n \times n$ solution to the Riccati equation by the integration of $(m{+}p)n$ equations backwards in time towards a steady state solution. While the reduced storage costs of the Chandrasekhar equations make some large problems tractible that may not be otherwise solvable, the slow convergence towards a steady state solution magnify the computational costs associated with the integration. An attempt to overcome this challenge, proposed by Banks and Ito [8] in 1991, is to integrate (6)–(7) until $K(t)$ begins to converge, then to correct the gain using Kleinman-Newton iterations. In this mode, the Chandrasekhar equations become an effective way to compute an initial guess for iterative methods.

Our proposed methodology follows a similar pattern: integrate the system (6)–(7) until $K(t)$ begins to converge over $[T, 0]$ ($T < 0$). However, at this point, we perform model reduction on (7) to introduce a reduced problem to continue integration backwards from $t = T$.

B. Rationale

The plausibility for our strategy comes from the structure of the Chandrasekhar equations. As the gain converges, say $K(t) \approx \tilde{K}$ for $t < T$, equation (7) has the form

$$-\dot{L}(t) = \left(A - B\tilde{K}\right)^T L(t), \qquad t < T < 0,$$

with $L(T)$ given. Thus, one would expect the "high frequency" content of the final condition $L(T)$ to decay rapidly and thus, solutions to (7) from $t = T$ would be well approximated by the "low frequency" content in $L(T)$. In other words, the solution to this hypothetical equation would be dominated by the rightmost r eigenvectors of $(A{-}B\tilde{K})^T$.

There are many ways in which model reduction can be carried out for (7). In this study, we find an empirical basis

J. Borggaard and M. Stoyanov are with the Department of Mathematics and the Interdisciplinary Center for Applied Mathematics, Virginia Tech `jborggaard@vt.edu`, `stoyanov@vt.edu`

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

for L using the proper orthogonal decomposition (POD) [9] (introduced in the control literature under names such as principal component analysis [10] and reduced-basis methods [11], [12], among others) on an input collection consisting of stored values of the trajectory L.

The derivation of the Chandrasekhar equations introduces the factorization

$$-\dot{\Pi}(t) = L(t)L^T(t)$$

into the differential Riccati equation. Thus,

$$\Pi = \lim_{t \to -\infty} \Pi(t) = \int_{-\infty}^{0} L(t)L^T(t)\ dt$$

using the final condition $\Pi(0) = 0$, and

$$\Pi = \underbrace{\int_{-\infty}^{T} L(t)L^T(t)\ dt}_{\Pi_{\text{res}}} + \int_{T}^{0} L(t)L^T(t)\ dt. \tag{8}$$

The last term above is integrated using the Chandrasekhar equation, while we approximate the first integral is approximated using a low-dimensional basis for L. The expression (8) above shows a clear connection between finding a good basis for L and a good basis for Π_{res}.

II. METHODOLOGY

In this section, we outline a methodology for solving the control problem (1)–(3) using an efficient long-time integration of the Chandrasekhar equations. First of all, we follow [8] to integrate (6)–(7) until $K(t)$ "begins to converge" (or similarly until $L(t)$ can be very accurately approximated with a low-dimensional basis). This can be determined in practice by monitoring K, L and the reduced-order equations for L during the integration backward from 0 to $T < 0$. For this integration, we use the following algorithm with the notation $K_i \approx K(-i\Delta t)$ and $L_i \approx L(-i\Delta t)$ to allow for implementation in conventional forward integration packages.

Algorithm II.1 (Banks and Ito [8]) *Given A, B and C. Set $K_0 = 0$ and $L_{-1} = L_0 = C^T$. For $i = 0, 1, \ldots$*

```
1) $K_{i+1}^{(0)} = K_i + \Delta t R^{-1} B^T \left(\frac{3}{2} L_i L_i^T - \frac{1}{2} L_{i-1} L_{i-1}^T\right)$
2) $K_{i+1/2}^{(0)} = \frac{1}{2} K_{i+1}^{(0)} + \frac{1}{2} K_i$
3) $L_{i+1} = L_i + \Delta t \left(A - B K_{i+1/2}^{(0)}\right)^T \left(\frac{1}{2} L_{i+1} + \frac{1}{2} L_i\right)$
4) $K_{i+1} = K_i + \Delta t R^{-1} B^T \left(\frac{1}{2} L_{i+1} L_{i+1}^T + \frac{1}{2} L_i L_i^T\right)$
```

Test for convergence and continue if necessary.

Note that the third step in Algorithm II.1 is implicit due to the stiffness in the matrix $A - BK(t)$. This algorithm concludes at an iteration k whence we have the approximation $L_k \approx L(T)$.

Secondly, from a collection of $\ell + 1$ stored time snapshots of the solution or by performing additional integration to (7), compute a reduced-basis using POD, i.e.

$$V = [v_1 | v_2 | \cdots | v_r] = POD(L_{k-\ell}, \ldots, L_{k-1}, L_k)$$

or

$$V = [v_1 | v_2 | \cdots | v_r] = POD(L_k, L_{k+1}, \ldots, L_{k+\ell}),$$

respectively, where $r \ll n$ is the number of basis vectors needed to accurately represent L over $(-\infty, T]$. We are choosing the latter of these two approaches here and increasing the time step since integration accuracy is not as critical in generating the basis. Note that there are three parameters used in the generation of V: r, ℓ ($> r$), and T. We will comment on choices for these parameters below.

Finally, we approximate the solution L in the orthogonal basis generated by columns of V,

$$L(t) = Vc(t) \qquad \text{where} \qquad c(t) \in \mathbb{R}^r.$$

Thus, (6)–(7) have the form

$$\begin{aligned} -\dot{K}(t) &= R^{-1} B^T V c(t) c^T(t) V^T && (9) \\ -\dot{c}(t) &= V^T \left(A - BK(t)\right)^T V c(t) && (10) \end{aligned}$$

from final conditions $K(T)$ and $c(T) = V^T L(T)$. These equations could be integrated directly since the equation that needs to be handled implicitly, (10), is low-dimensional. It may also be advantageous to introduce a new variable $\xi = KV$ in (9), although introducing a new level of approximation.

However, as in [8], we will consider the associated Riccati equation for the "tail" of the integration. If we write

$$K = K(T) + K_{\text{res}}, \tag{11}$$

then $K_{\text{res}} = R^{-1} B^T \Pi_{\text{res}}$ and Π_{res} is the solution to

$$\begin{aligned} &\left(A - BK(T)\right)^T \Pi_{\text{res}} + \Pi_{\text{res}} \left(A - BK(T)\right) \\ &\qquad - \Pi_{\text{res}} B R^{-1} B^T \Pi_{\text{res}} + L(T) L(T)^T = 0. \end{aligned}$$

However, instead of using a Kleinman-Newton iteration on this Riccati equation, we project this onto V. Define P as the solution to

$$\tilde{A}^T P + P\tilde{A} - P\tilde{B} R^{-1} \tilde{B}^T P + c(T) c(T)^T = 0. \tag{12}$$

where $\tilde{A} = V^T \left(A - BK(T)\right) V$ and $\tilde{B} = V^T B$. Since r is small, the solution of (12) for P is easy to obtain. Our approach is summarized below.

Algorithm II.2 (Long-Time Integrator for (6)–(7)) *Given A, B and C.*

```
1) Apply Algorithm II.1 until time $T < 0$.
2) Build a reduced-basis $V$ for $L(t)$, $t < T$.
3) Solve (12) for $P$, set $\tilde{K}_{\text{res}} = R^{-1} B^T V P V^T$.
4) Use (11) to obtain $K$.
```

A. Comments

- The size of the reduced-basis, r, and the final time, T, are determined by how well L can be approximated in the columns of V over $(-\infty, T]$. This assessment can be built into the time integration as a stopping criteria. Since the initial time integration dominates the cost of this algorithm, r should be taken large enough so that additional vectors lead to negligible improvement.

- Steps 3) and 4) above are mathematically equivalent to integrating (9)–(10) from $t : T \to -\infty$. However, the small size of the Riccati problem (12) removes the need to apply Algorithm II.1 to (9)–(10).
- Steps 3) and 4) represent an attempt to approximate Π_{res} and use this to compute K_{res}. The accuracy of this calculation depends on the adequacy of the columns of V to approximate L. Namely, if
$$\int_{-\infty}^{T} \|L(t) - VV^T L(t)\| \; dt < \epsilon/2,$$
then we have the estimate
$$\|\Pi_{\text{res}} - VV^T \Pi_{\text{res}} VV^T\| \leq \epsilon \|\Pi_{\text{res}}\|^{1/2}.$$
- A natural question is "how close is the solution VPV^T to Π_{res} when P is obtained from (12)?" The answer depends on $\epsilon\|\Pi_{\text{res}}\|^{1/2}$ and a Lyapunov bound. This will be the subject of a follow-up paper.
- We are actually interested in the approximation of K_{res} by $\tilde{K}_{\text{res}}$. In this paper, we use a numerical example to investigate how well this can be approximated in practice.
- In our experiments $\ell > 3r$ was sufficient.

III. NUMERICAL EXAMPLE

We demonstrate the effectiveness of our algorithm by solving a control problem involving the two-dimensional advection-diffusion-reaction (ADR) equation with two control inputs on the boundary and one control output. This linear distributed parameter control problem is motivated by the problem of controlling the two-dimensional Burgers equation.

A. Problem Description

Our example is motivated by the boundary control of the two-dimensional Burgers equation to a steady-state reference solution first posed in Camphouse and Myatt [13]. The model equations are

$$w_t + \left(\frac{1}{2}c_1 w^2\right)_\xi + \left(\frac{1}{2}c_2 w^2\right)_\eta = \mu\left(w_{\xi\xi} + w_{\eta\eta}\right) \quad (13)$$

with domain sketched in Fig. 1. As in [13], we consider parameter values $c_1 = 1$ and $c_2 = 0$, and will use either $\mu = 1/200$ or $\mu = 1/300$ (the latter was considered in [13]). Control is applied on the top Γ_1 and bottom Γ_2 surfaces of the internal rectangular obstruction. The control objective is to steer the flow to the steady-state solution, denoted by W. Thus, let W solve

$$c_1 W W_\xi + c_2 W W_\eta = \mu\left(W_{\xi\xi} + W_{\eta\eta}\right)$$

with

$$W|_{\Gamma_{in}} = \frac{1}{.24^2}\eta(.48 - \eta), \qquad \frac{\partial W}{\partial n}|_{\Gamma_{out}} = 0,$$

and

$$W|_{\Gamma - \Gamma_{in} - \Gamma_{out}} = 0.$$

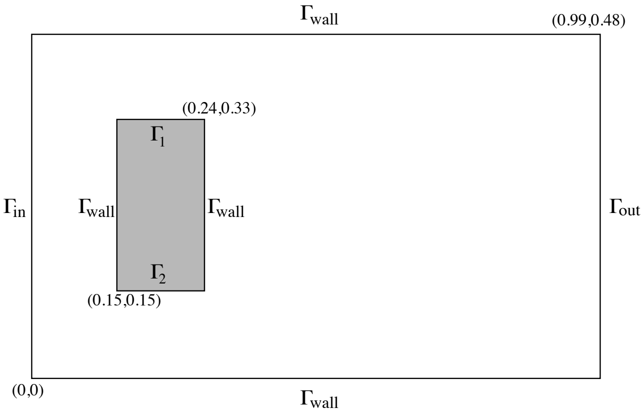


Fig. 1. Computational domain for 2D Burgers equation

The control objective is achieved by linearizing the two-dimensional Burgers equation about W and using linear feedback control to stabilize the perturbation z from the steady-state. If we define $w = W + z$ and substitute into (13), we obtain

$$z_t + c_1\left(W z_\xi + z W_\xi\right) + c_2\left(W z_\eta + z W_\eta\right) = \mu\left(z_{\xi\xi} + z_{\eta\eta}\right) \quad (14)$$

with

$$z|_{\Gamma_{in}} = 0, \qquad \frac{\partial z}{\partial n}|_{\Gamma_{out}} = 0,$$

$$z|_{\Gamma_1} = u_1, \qquad z|_{\Gamma_2} = u_2, \qquad \text{and} \qquad z|_{\Gamma_{wall}} = 0.$$

Note that we have discarded the nonlinear term $c_1 z z_\xi + c_2 z z_\eta$ and used the fact that $c_1 W W_\xi + c_2 W W_\eta = \mu\left(W_{\xi\xi} + W_{\eta\eta}\right) = 0$. If we wish to control w to the steady-state W, then we want to drive z to zero.

Linear feedback control amounts to designing a control for the advection-diffusion-reaction equation (14) to stabilize z. We follow [13] and seek to minimize the output

$$y(t) = \int_{\xi=.6}^{\xi=.8} \int_{\eta=0.03}^{\eta=0.45} z(\xi, \eta, t) d\eta \; d\xi.$$

This problem for minimizing $y(t)$ above is discretized using finite elements to arrive at a linear control problem of the form (1)–(3). We consider various problem sizes (n) by changing the number of finite elements used to approximate the operators in (14). The number of control inputs ($m = 2$) and control outputs ($p = 1$) are independent of discretization size.

Note that it is more natural to compare convergence of functional gains [14] for distributed parameter control problems. The notion of convergence is then in a function space rather than a space that varies with n. In our notation, these functional gains, h_1 and h_2, have the form

$$[Kx](t) = \begin{bmatrix} \int_\Omega h_1(\xi, \eta) z(\xi, \eta, t) \; d\Omega \\ \int_\Omega h_2(\xi, \eta) z(\xi, \eta, t) \; d\Omega \end{bmatrix}.$$

The finite element approximation of h_1 and h_2 are then considered as the mesh is refined (as $n \to \infty$).

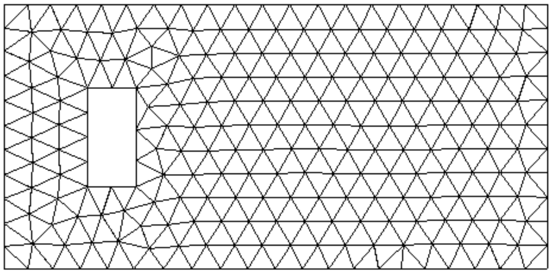


Fig. 2. Finite Element Mesh: 182 DOF

TABLE I

RELATIVE ERROR FROM EARLY TERMINATION

final time (T)	relative error in gains (E_{rel})
-0.5	0.999989
-1.0	0.999643
-1.5	0.984436
-2.0	0.932892
-4.0	0.514311
-10.0	0.028214

B. Integration of Chandrasekhar Equations

In our first experiment, we applied Algorithm II.2 to the ADR control problem above with a very coarse mesh corresponding to $n = 182$, see Fig. 2. In all of our experiments, we use a uniform time step of $\Delta t = 1/256 = 0.0039$. For this discretization, we used the parameter value $\mu = 1/200$. As a measure of the accuracy of our algorithm, we compute the relative error of the gains

$$E_{\text{rel}} = \frac{\int_\Omega (h_1^c - h_1^r)^2 + (h_2^c - h_2^r)^2 \; d\Omega}{\int_\Omega (h_1^r)^2 + (h_2^r)^2 \; d\Omega}.$$

where superscripts refer to gains obtained by c: Chandrasekhar equations by Algorithm II.2 and r: Riccati equations directly (using the algorithm `lqr` in MATLAB).

The slow convergence of the Chandrasekhar equations towards a steady state solution is documented in Table I. Here, we see that the that there is a 2.8% relative error when integrating the equations out to $T = -10$. This represents 2560 time steps in Algorithm II.1. This slow convergence is disappointing and it responsible for the limitations of the Chandrasekhar equation approach. We do point out that the error reduction from 0 to -2 is much worse than from -2 to -4. In fact, the relative error does appear to drop quickly from -4 to -10, but the computational cost is too great if we want very accurate solutions. Obviously, we would like to minimize the cost of the first step in Algorithm II.2 by only integrating the Chandrasekhar equations as far as needed to develop a good approximation for the tail of L. We study the influence of the integration time T below.

C. Effectiveness of Algorithm II.2

Our second experiment tests the accuracy of Algorithm II.2. To begin with, we plot the first functional gain h_1 obtained after integrating the Chandrasekhar equations to $T = -2$ in Figure 3 and after completion of Algorithm II.2 in Figure 4 (the results for gains h_2 are similar). These figures demonstrate that the functional gain changes substantially in steps 2) – 4). Thus, the algorithm makes significant updates to $K(T)$ even though the premise of the algorithm is that K is nearly converged. The first 6 POD modes used to generate V using an additional $\ell = 30$ integration steps are plotted in Fig. 5. Similarity between mode 1 and $h_1(T)$ are clear. The change (in structure) in h_1 we observe is then facilitated by the remaining 5 modes.

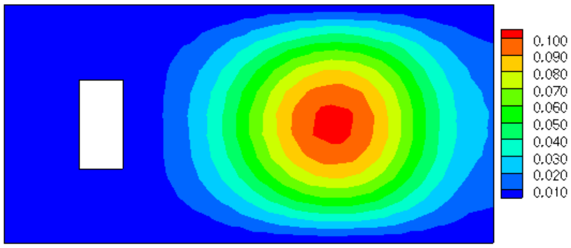


Fig. 3. Functional Gain at T=-2

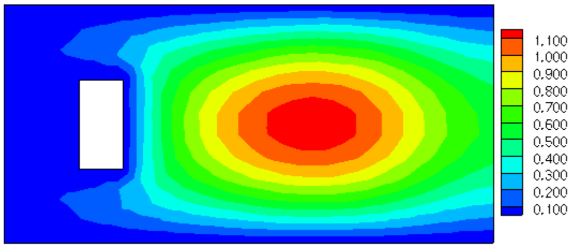
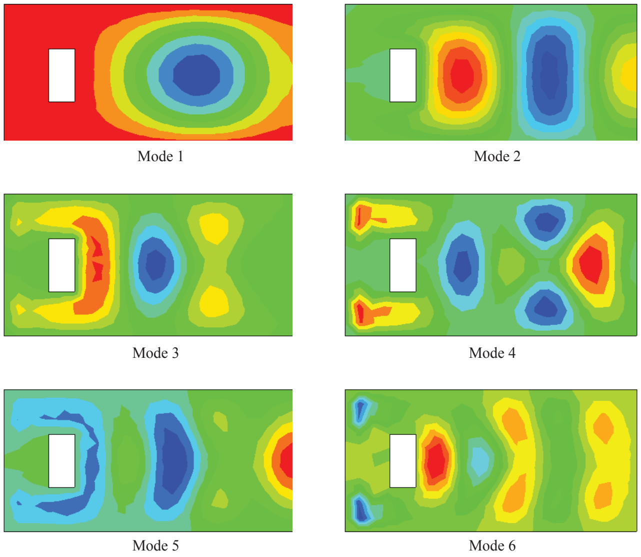


Fig. 4. Functional Gain with Correction



Mode 1

Mode 2

Mode 3

Mode 4

Mode 5

Mode 6

Fig. 5. POD Modes of L

The accuracy of Algorithm II.2, given by relative error $E_{\rm rel}$ for different integration times $|T|$ and different sizes of the reduced-order model (r) is presented in Fig. 6. As we expect, $E_{\rm rel}$ is reduced with longer integration time and larger model dimension. However, we observe that longer integration times lead to smaller model dimension requirements to get "maximum" accuracy. This supports our rationale that components of higher frequency modes are ultimately eliminated with integration. It is impressive that the relative error of 50% obtained at $T = -4$ can be reduced 4 orders of magnitude with a 4 dimensional model.

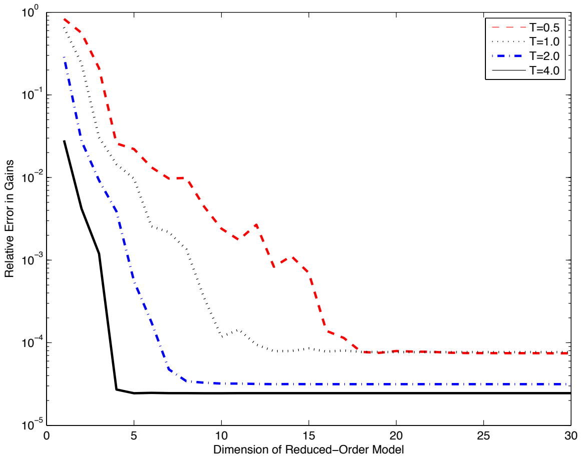


Fig. 6. Relative Error vs. Basis Size (r), Different T

Obviously, there seems to be a limit in the amount of useful information contained in our reduced-bases. To obtain more content in the basis, we would need more content in the input collection. However, the expense of working with a larger input collection soon outweighs any potential gain in the accuracy of the basis. More effective basis selection strategies, however, could lead to high payoff in this algorithm.

As a final experiment, we reproduce the gains obtained in Camphouse and Myatt [13]. In their study, they considered $\mu = 1/300$. Our coarse mesh above is insufficient to approximate Burgers equation with a standard Galerkin finite element procedure at this parameter value. Thus, we consider a much finer mesh in Fig. 8, leading to $n = 1413$ degrees of freedom. We again compare the difference in the functional gains after step 1) (for $T = -2$) and step 4) of Algorithm II.2 in Figs. 8 and 9, respectively. In Fig. 10, we again observe good improvement in the accuracy of the gains as the dimension of our reduced-order model is increased and as more time-snapshots are incorporated into building the POD basis from $T = -2$.

D. Comparison to Kleinman-Newton Iterations

As a final comparison, we compare our approach to the Kleinman-Newton iterations as proposed in Banks and Ito [8]. The initial step of integrating the Chandrasekhar equations is the same in both cases. Thus, the comparisons must be made on the remainder of the algorithms.

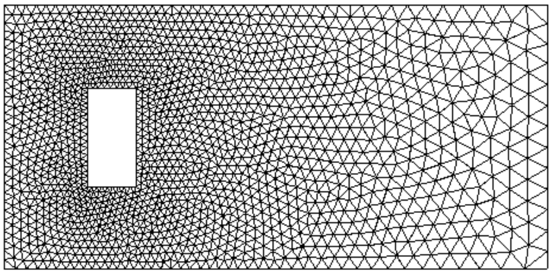


Fig. 7. Finite Element Mesh: 1413 DOF

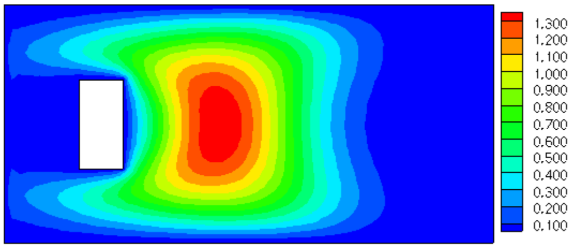


Fig. 8. Functional Gain at T=-2

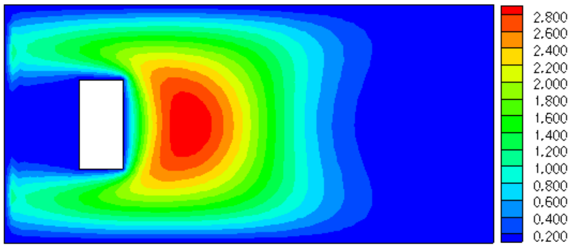


Fig. 9. Functional Gain with Correction

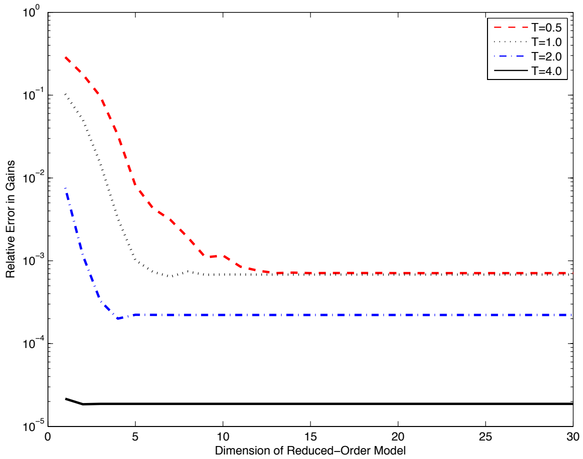


Fig. 10. Relative Error vs. Basis Size (r), Different T

In Algorithm II.2, the work is dominated by step 2. The only work involving matrices of dimension n in steps 3 and 4 include $B^T_{(m\times n)}V_{(n\times r)}$ and $(R^{-1}B^TVP)_{(m\times r)}V^T_{(r\times n)}$ matrix products in step 3 and a sum of $(m \times n)$ matrices in step 4. In step 2, we used a reduced SVD on a matrix of size $n \times \ell$, however, we could have simply performed a Gram-Schmidt procedure on this matrix featuring cost $O(n\ell^2)$.

In the Kleinman-Newton method, we implemented a Smith method with variable shifts as in [8]. For our comparison, we used optimal shifts that required all eigenvalues of $A - BK(T)$ (these shifts would not be available in practice). To achieve comparable errors, we required about 70 sparse $(n\times n)$ linear solves for $n = 182$ and about 140 sparse $(n\times n)$ linear solves for $n = 1413$. Thus, Algorithm II.2 leads to a substantial improvement since it has work less than the cost of 1 sparse $(n \times n)$ linear solve.

However, the Kleinman-Newton method can achieve far greater accuracy with additional iterations. Algorithm II.2 has accuracy limited by how well L is approximated in $\mathrm{span(V)}$. In addition, the Kleinman-Newton method has the additional feature that it can correct numerical integration errors that may have been introduced in approximating the Chandrasekhar equations.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

We presented an attractive alternative to the hybrid Chandrasekhar/Kleinman-Newton algorithm for solving large scale Riccati equations introduced in [8]. Our method, which may be labeled as a hybrid Chandrasekhar/reduced-Riccati algorithm, replaces the relatively more expensive Kleinman-Newton iterations with a very low order Riccati problem. The effectiveness of this algorithm was shown for a distributed parameter control problem consisting of boundary control of the advection-diffusion-reaction equations.

Note that this approach differs from the usual model reduction approach to large scale control problems since we use the large dimensional problem from 0 to T. The standard approach is to reduce the dimensional of the problem immediately, then solve the control problem. Although our approach is more expensive, we only introduce a reduced-basis once we have gleaned enough information from the full problem. In this sense, the hybrid approach introduced here tries to solve the control problem more accurately by introducing the approximation later in the solution process.

B. Future Works

There are certainly a number of remaining questions that will be answered in future works. First of all, we are basing our model reduction on the proper orthogonal decomposition. This is an important model reduction method, but is primarily used in nonlinear equations. It does make some sense to use this approach in our setting since we have the integration history of L available and (7) is nonlinear in the variables K and L. However, the differential equation for L (7) is linear in the L variable. Thus, other model reduction methods may allow for continued reduction in relative error with model size. Part of our future work will also consider the trade-offs in longer integration of Chandrasekhar equations in the present approach vs. number of Kleinman-Newton iterations to achieve the same relative error tolerance.

Furthermore, the analysis of this method can be approached by considering the accuracy in which Π_{res} can be approximated using a reduced Riccati equation (12). We have an initial estimate of the accuracy of this solution as a function of the accuracy of the basis generated by columns of V to represent L over $(-\infty, T]$. We intend to expand upon this analysis in a future paper.

V. ACKNOWLEDGMENTS

The authors gratefully acknowledge partial support from the Air Force Office of Scientific Research under grants FA9550-05-1-0449 and FA9550-08-1-0136 and the National Science Foundation under grant DMS 0513542. We would also like to thank the reviewers for comments that improved the clarity of the original submission.

REFERENCES

[1] E. Lee and L. Markus, *Foundations of Optimal Control Theory*. Krieger Publishing Company, Inc., 1986.

[2] P. Benner, "Solving large-scale control problems," *IEEE Control Systems Magazine*, vol. 24, no. 1, pp. 44–59, 2004.

[3] J. Casti, "Matrix Riccati equations, dimensionality reduction and generalized X-Y functions," *Utilitas Math*, vol. 6, pp. 95–110, 1974.

[4] T. Kailath, "Some Chandrasekhar-type algorithms for quadratic regulators," in *Proc. IEEE Conference on Decision and Control*, 1972.

[5] A. Lindquist, "On Fredholm integral equations, Toeplitz equations and Kalman-Bucy filtering," *Journal of Applied Math and Optimization*, vol. 1, pp. 355–373, 1975.

[6] M. Sorine, "Sur le semi-groupe nonlinéaire associé à l'equation de Riccati," INRIA, Tech. Rep. 167, 1982.

[7] R. Powers, "Chandrasekhar equations for distributed parameter systems," Ph.D. dissertation, Virginia Tech, 1984.

[8] H. Banks and K. Ito, "A numerical algorithm for optimal feedback gains in high dimensional linear quadratic regulator problems," *SIAM Journal on Control and Optimization*, vol. 29, no. 3, pp. 499–515, May 1991.

[9] G. Berkooz, P. Holmes, and J. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual Review of Fluid Mechanics*, vol. 25, pp. 539–575, 1993.

[10] B. Moore, "Principal component analysis in linear systems: controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. AC-26, 1981.

[11] J. Burns and B. King, "A reduced basis approach to the design of low-order feedback controllers for nonlinear continuous systems," *Journal of Vibration and Control*, vol. 4, pp. 297–323, 1998.

[12] K. Ito and S. Ravindran, "A reduced order method for simulation and control of fluid flows," *Journal of Computational Physics*, vol. 143, no. 2, pp. 403–425, 1998.

[13] R. Camphouse and J. Myatt, "Feedback control for a two-dimensional Burgers equation system model," *2nd AIAA Flow Control Conference*, 2004, AIAA Paper No. 2004-2411.

[14] J.-L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*. Berlin: Springer-Verlag, 1971.