

A Reliable and Computationally Efficient Algorithm for Imposing the Saddle Point Property in Dynamic Models*

Gary S. Anderson[†]

September 5, 2000

(Anderson and Moore, 1983; Anderson and Moore, 1985) describe a powerful method for solving linear saddle point models. The algorithm has proved useful in a wide array of applications including analyzing linear perfect foresight models, providing initial solutions and asymptotic constraints for nonlinear models. The algorithm solves linear problems with dozens of lags and leads and hundreds of equations in seconds. The technique works well for both symbolic algebra and numerical computation.

Although widely used at the Federal Reserve, few outside the central bank know about or have used the algorithm. This paper attempts to present the current algorithm in a more accessible format in the hope that economists outside the Federal Reserve may also find it useful. In addition, over the years there have been undocumented changes in approach that have improved the efficiency and reliability of algorithm. This paper describes the present state of development of this set of tools.

Draft: Revision 1.8 – September 5, 2000 1

1 Introduction and Summary

George Moore and I developed this group of algorithms to facilitate research and analysis activities at the Federal Reserve.¹ Economists at the Board have an operational need for tools that are useful for building, estimating and simulating moderate to large scale rational expectations models. This context dictates a need for careful attention to computational efficiency and numerical stability of the algorithms.

The algorithms have proved very durable and useful for staff at the central bank. The algorithm provides helpful diagnostic information at each stage of the computation. Figure 1 presents a flow chart summarizing the AIM algorithm.

Many economists at the federal reserve use the algorithm in their daily work.(Bomfim, 1996; Fuhrer and Moore, 1995a; Fuhrer and Moore, 1995b; Fuhrer and Moore, 1995; Fuhrer and Madigan, 1997; Fuhrer, 1997b; Fuhrer, 1997a; Fuhrer, 1996; Fuhrer *et al.*, 1995; Fuhrer and Hooker, 1993; Fuhrer, 1994; Fuhrer, 1997c; Orphanides *et al.*, 1997; Andrew *et al.*, 1998;

Orphanides, 1998) Yet, few economists outside the central bank seem to know about the method.

This paper attempts to make the method and approach more widely available by describing the theory alongside its implementation. The “C”, Matlab, Gauss and Mathematica code are available from the author at

(<http://www.bog.frb.fed.us/pubs/oss/oss4/aimindex.html>).

The most distinctive features of this approach are:

- its algorithm for computing the minimal dimension state space transition matrix(See Section 6.1)
- its use of biorthogonality to characterize the asymptotic constraints that guarantee stability(See Section 2.2).

This results in a significant reduction in the size of the eigenvalue calculations and an increase in reliability of the results for most problems.

Although the ideas that make the algorithm work are intuitive, an efficient implementation requires some subtle features. Experience with the algorithm has inspired some modifications, extensions and improvements in the original algorithm described by Anderson and Moore(Anderson and Moore, 1985). These include:

¹At the Board, economists commonly refer to this family of algorithms as the AIM algorithm. A metaphor relating our approach to the “shooting method” inspired the name.

- Development of a symbolic algebra version of the algorithm
- Use of QR-Decomposition for rank determination
- Techniques for accommodating unit roots

This paper describes the new implementation of the algorithm.

Economists at the Federal Reserve routinely use these algorithms for solving system of equations with dozens of variables each with dozens of lags and leads. Other authors(Binder and Peseran, 1995; Zadrozny, 1996; Sims, 1996) have contributed solutions for these types of problems. By exploiting the structure of the problem and doing calculations in place where possible, the algorithm limits the cost of adding lags and leads to the model better than these alternative approaches. The number of floating point operations increases with the cube of the number of variables in the state space. In contrast to other approaches, the number of floating point operations only increases with the square of the number of lags and leads.

This paper describes how one can apply widely available numerical routines from LAPACK to efficiently and reliably attack each phase of the problem. In addition, the example shows how symbolic algebra programs can exploit the same solution strategy to get closed form solutions for some dynamic models.

Section 2.1 describes the procedures for computing the state space transition matrix. Section 2.2 shows how to obtain asymptotic linear constraints and how to apply the constraints for stability. Section A presents floating point operation counts for each phase of the algorithm. Section 7.1 presents the results of numerical experiments with the algorithm.

2 The Algorithms

Problem Statement

$$\sum_{i=-\tau}^{\theta} H_i x_{t+i} = \Psi z_t, t = 0, \dots, \infty \quad (1)$$

with initial conditions, if any, given by constraints of the form

$$x_i = x_i^{data}, i = -\tau, \dots, -1 \quad (2)$$

where both τ and θ are non-negative, and x_t is an L dimensional vector with

$$\lim_{t \rightarrow \infty} \|x_t\| = < \infty \quad (3)$$

The paper describes several algorithms for analyzing and solving various aspects of this problem.

Algorithm 1 Given *structural model matrices*, $H_i, i = -\tau, \dots, \theta$, determine a matrix Z_f of *auxiliary initial conditions* and a *transition matrix* A such that the x_t satisfying

$$Z_t \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0$$

$$\begin{bmatrix} x_{t-\tau+1} \\ \vdots \\ x_{t+\theta} \end{bmatrix} = A \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix}$$

satisfy the linear homogeneous system(See Section 2.1).

Algorithm 2 Given a transition matrix, A , and auxiliary initial conditions, Z_f , determine an *asymptotic constraints matrix*, Q , such that

$$Q \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0 \implies \lim_{k \rightarrow \infty} \|x_{t+k}\| < \infty$$

for all x_t satisfying the linear homogenous system(See Section 2.2).

Algorithm 3 Given an asymptotic constraints matrix, Q , determine the existence and uniqueness of *convergent autoregression matrices*, $B_i, i = -\tau, -1$, such that

$$x_t = \sum_{i=-\tau}^{-1} B_i x_{t+i}$$

satisfies the linear homogeneous system and $\lim_{k \rightarrow \infty} \|x_{t+k}\| = < \infty$ (See Section 2.2).

Algorithm 4 Given structural model matrices, $H_i, i = -\tau, \dots, \theta$ and Ψ , convergent autoregression matrices $B_i, i = -\tau, -1$ determine the existence and uniqueness of *inhomogeneous factor matrices*, Φ and F such that

$$x_t = \sum_{i=-\tau}^{-1} B_i x_{t+i} + \begin{bmatrix} 0 & \dots & 0 & I \end{bmatrix} \sum_{s=0}^{\infty} (F^s \begin{bmatrix} 0 \\ \Phi \Psi z_{t+s} \end{bmatrix})$$

satisfies the linear inhomogeneous system(See Section 3).

Algorithm 5 Given structural model matrices, $H_i, i = -\tau, \dots, \theta$ and Ψ , convergent autoregression matrices $B_i, i = -\tau, -1$, expectations for the exogenous factors, $E[z_{t+s}|I_t], s = 0, \dots, \infty$, convergent autoregression matrices, $B_i, i = -1 \dots, -\tau$ and inhomogeneous factor matrices, Υ and Φ and setting

$$E[x_{t+k}|I_t] = \sum_{i=-\tau}^{-1} B_i E[x_{t+k+i}|I_t] + [0 \quad \dots \quad 0 \quad I] \sum_{s=0}^{\infty} (F^s \begin{bmatrix} 0 \\ \Phi \Psi E[z_{t+k+s}|I_t] \end{bmatrix}),$$

$$k = 0, \dots, \infty$$

determine the existence and uniqueness of an observable structure matrix S , and stochastic transition matrices φ, ρ such that

$$\epsilon_t = S \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_t \end{bmatrix}$$

$$\begin{bmatrix} x_{t-\tau+1} \\ \vdots \\ x_t \end{bmatrix} = \rho \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t-1} \end{bmatrix} + \varphi [\epsilon_t + \Psi E[z_t|I_t]]$$

(See Section 4).

Saddle point problems combine initial conditions and asymptotic convergence to identify their solutions. The uniqueness of solutions to system 6 requires that the transition matrix characterizing the linear system have an appropriate number of explosive and stable eigenvalues (Blanchard and Kahn, 1980), and that the asymptotic linear constraints are linearly independent of explicit and implicit initial conditions (Anderson and Moore, 1985).

The solution methodology entails

1. using equation 6 to compute a state space transition matrix.
2. Computing the eigenvalues and the invariant space associated with large eigenvalues
3. Combining the constraints provided by:
 - (a) the initial conditions,
 - (b) auxiliary initial conditions identified in the computation of the transition matrix and
 - (c) the invariant space vectors

Figure 1 presents a flow chart of the summarizing the algorithm. For a description of a parallel implementation see (Anderson, 1997b) For a description of a continuous time application see (Anderson, 1997a).

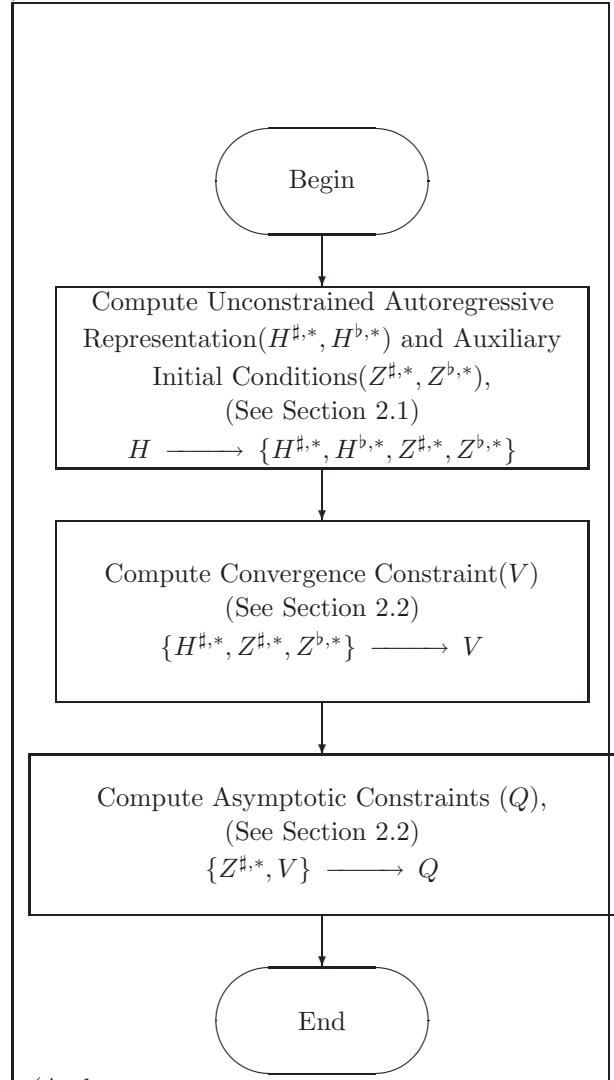


Figure 1: Algorithm Overview

THE EXAMPLE:

The paper describe how to apply the algorithm to the following model. (Sims, 1996)

$$\begin{aligned}
w_t &= \frac{1}{N} E_t \left[\sum_{i=0}^{n-1} W_{t+i} \right] - \alpha(u_t - u_n) + \nu_t \\
W_t &= \frac{1}{N} \sum_{i=0}^{n-1} w_{t-i} \\
u_t &= \theta_{t-1} + \gamma W_t + \mu + \epsilon_t \\
E_t[\nu_{t+i}] &= E_t[\epsilon_{t+i}] = 0 \forall i \geq 0
\end{aligned}$$

In Section 2, $N = 2$. Section 7.1, explores the relationship between N and computation time.

2.1 Algorithm 1 State Space Transition Matrix Computation

The first phase of the algorithm applies full rank linear transformations to the original linear system to express $x_{t+\theta}$ in terms of $x_{t-\tau} \dots x_{t+\theta-1}$. This algorithm generalizes and implements a procedure known as the shuffling algorithm. (Luenberger, 1978) It produces an autoregressive representation for the evolution of the components of the state space vectors along with a set of vectors that provide implicit initial conditions. Section 6.1 will explain the importance of the latter set of vectors.

Given a system

$$\begin{bmatrix} H_{-\tau}^{\sharp*} & \dots & H_{\theta}^{\sharp*} \end{bmatrix} \begin{bmatrix} x_{-\tau} \\ \vdots \\ x_{\theta} \end{bmatrix} = 0$$

with $H_{\theta}^{\sharp*}$ non singular. Let

$$\Gamma^{\sharp} = -(H_{\theta}^{\sharp*})^{-1} \begin{bmatrix} H_{-\tau}^{\sharp*} & \dots & H_{\theta-1}^{\sharp*} \end{bmatrix} \quad (4)$$

Then

$$x_{\theta} = \Gamma^{\sharp} \begin{bmatrix} x_{-\tau} \\ \vdots \\ x_{\theta-1} \end{bmatrix}$$

This unconstrained auto-regression in x_t provides exactly what one needs to construct the state space transition matrix.

$$A^{\sharp} = \begin{bmatrix} 0 & I \\ \Gamma^{\sharp} & \end{bmatrix}$$

so that

$$\begin{bmatrix} x_{-\tau+1} \\ \vdots \\ x_{\theta} \end{bmatrix} = A \begin{bmatrix} x_{-\tau} \\ \vdots \\ x_{\theta-1} \end{bmatrix}$$

Algorithm 1 transforms an equation system like equation 6 into an equivalent system of the form given in Equation 4. The process of transformation generates auxiliary initial conditions that the dynamic system must obey in addition to the explicit initial conditions in equation system (6-3).

Figure 2 presents a flow chart for an algorithm for computing the components of the state space transition matrix and the auxiliary initial conditions (shift right equations).

Figure 3 presents a graphic characterization of the relevant sets of linear constraints. In the figure the regions where the coefficients are potentially non-zero are shaded gray. If $H_{\theta}^{\sharp,0}$ is singular, one can find a linear combination of the rows which preserves the rank of $H_{\theta}^{\sharp,0}$ but which annihilates one or more of its rows. Consequently, one can pre-multiply the matrices presented in Figure 3 to get the distribution of zeros displayed in Figure 4.

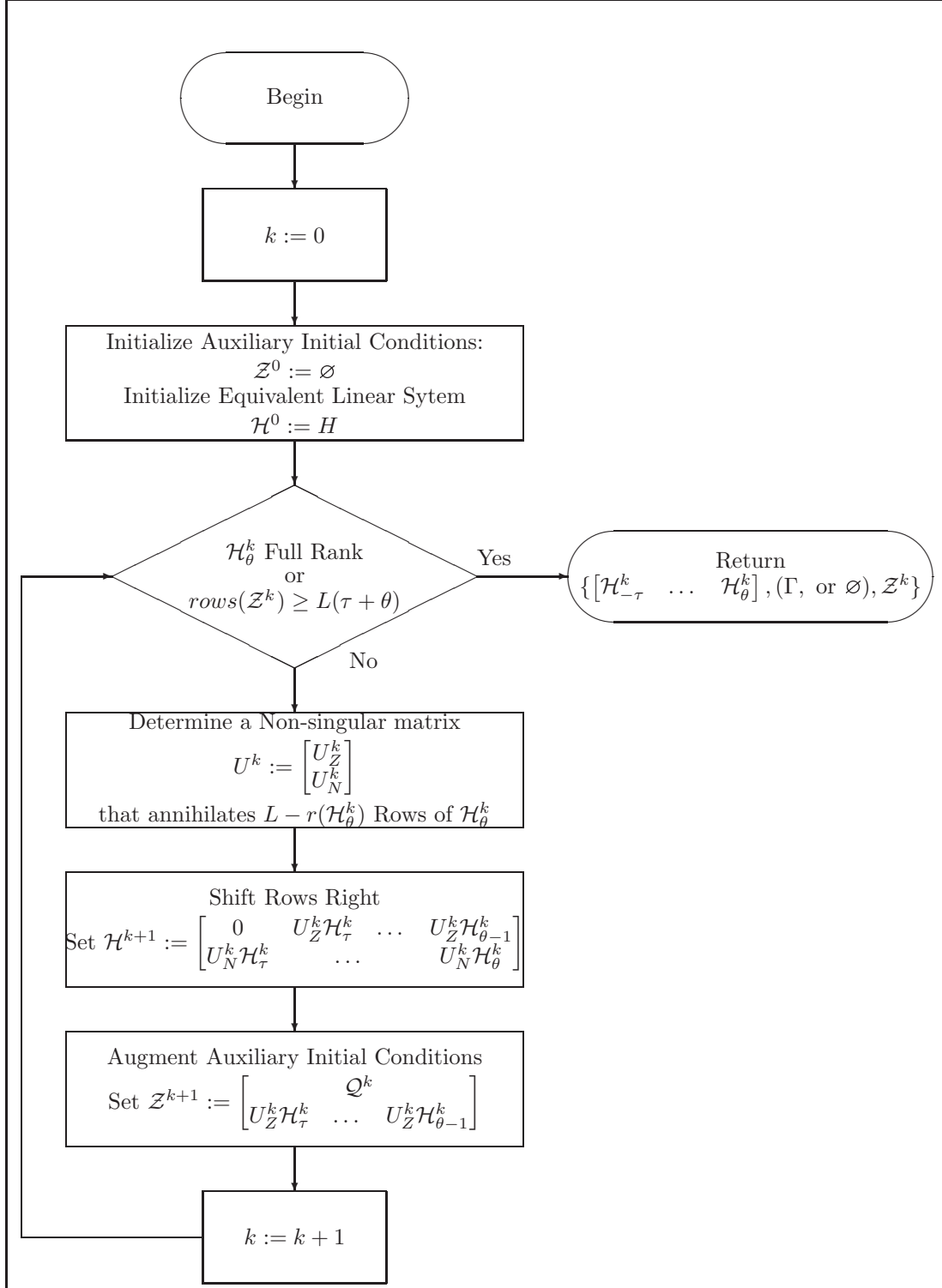
One can regroup the rows in the new tableau to get $H^{\sharp,1}$. By construction, the rank of $H_{\theta}^{\sharp,1}$ can be no less than the rank of $H_{\theta}^{\sharp,0}$. One can prove that repeating this process of annihilating and regrouping rows ultimately produces an $H^{\sharp,k} = H^{\sharp,*}$ with $H_{\theta}^{\sharp,*}$ non-singular.

Algorithm 1

```

1 Given  $H$ ,
2 compute the unconstrained autoregression.
3 func  $\mathcal{F}_1(H) \equiv$ 
4    $k := 0$ 
5    $\mathcal{Z}^0 := \emptyset$ 
6    $\mathcal{H}^0 := H$ 
7    $\Gamma = \emptyset$ 
8   while  $\mathcal{H}_{\theta}^k$  is singular  $\cap$  rows( $\mathcal{Z}^k$ )  $< L(\tau + \theta)$ 
9     do
10       $U^k = \begin{bmatrix} U_Z^k \\ U_N^k \end{bmatrix} := \text{rowAnnihilator}(\mathcal{H}_{\theta}^k)$ 
11       $\mathcal{H}^{k+1} := \begin{bmatrix} 0 & U_Z^k \mathcal{H}_{\tau}^k & \dots & U_Z^k \mathcal{H}_{\theta-1}^k \\ U_N^k \mathcal{H}_{\tau}^k & \dots & \dots & U_N^k \mathcal{H}_{\theta}^k \end{bmatrix}$ 
12       $\mathcal{Z}^{k+1} := \begin{bmatrix} U_Z^k \mathcal{H}_{\tau}^k & \dots & U_Z^k \mathcal{H}_{\theta-1}^k \\ \mathcal{Q}^k & \dots & \dots \end{bmatrix}$ 
13       $k := k + 1$ 
14   od
15    $\Gamma = -H_{\theta}^{-1} \begin{bmatrix} H_{-\tau} & \dots & H_{\theta-1} \end{bmatrix}$ 
16    $A = \begin{bmatrix} 0 & I \\ \Gamma & \end{bmatrix}$ 
17   return  $\{ \mathcal{H}_{-\tau}^k \dots \mathcal{H}_{\theta}^k \}, A, \mathcal{Z}^k \}$ 
18 .

```

Figure 2: Algorithm to Compute Unconstrained Auto-regression: Algorithm 1: $\mathcal{F}_1(\mathcal{H})$

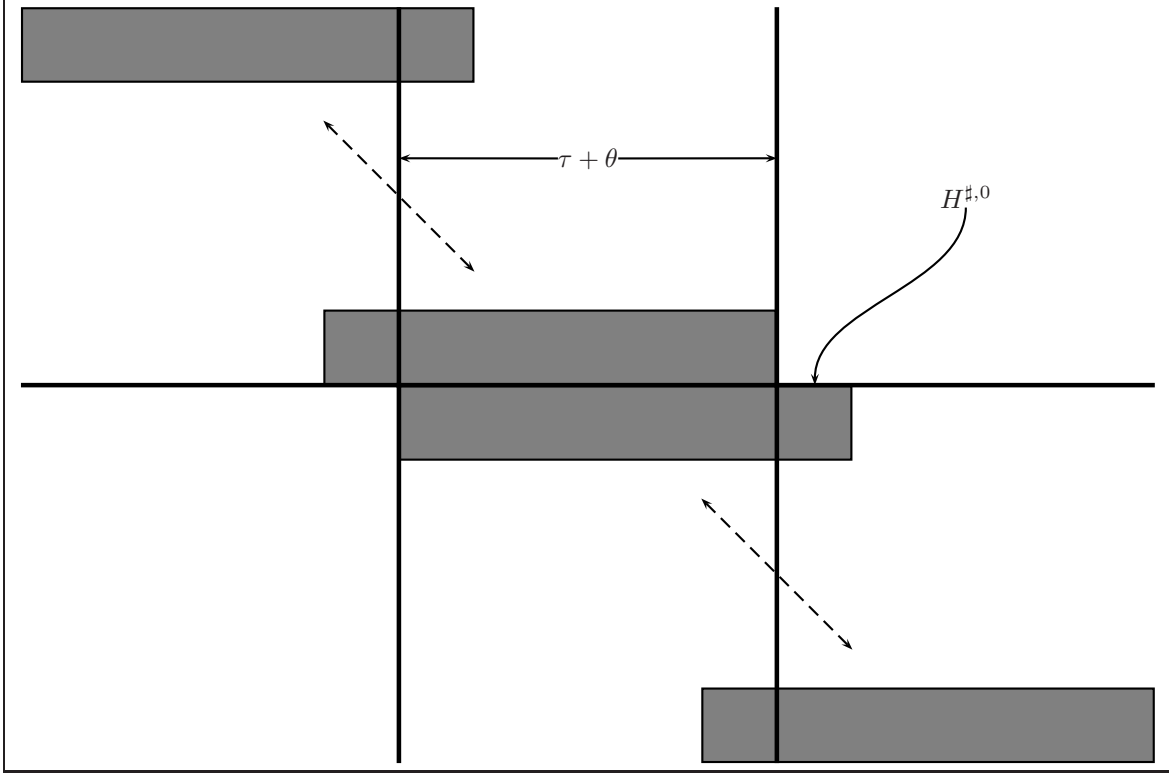


Figure 3: Matrix Tableau Characterization of Algorithm: Initial Tableau

Theorem 1 Let

$$\mathcal{H} = \begin{bmatrix} H_{-\tau} & & \dots & & H_{\theta} & & \\ & H_{-\tau} & \dots & & & & H_{\theta} \\ & & \ddots & & & & \\ & & & H_{-\tau} & \dots & & H_{\theta} \end{bmatrix}$$

There are two cases:

- When \mathcal{H} is full rank the algorithm terminates with $Z^{\#*}(Z^{b*})$ and non-singular $H_{\theta}^{\#*}(H_{\tau}^{b*})$
- When \mathcal{H} is not full rank the algorithm terminates when some row of $[\mathcal{H}_{-\tau}^k \dots \mathcal{H}_{\theta}^k]$ is zero.

Proof Consider the case when \mathcal{H} is full rank. Each step of the algorithm applies a rank preserving pre-multiplication by a nonsingular matrix. Each step of the algorithm where $H_{\theta}^{\#,k}$ is singular, increases the row rank of $Z^{\#,k}$ by one. At each step $Z^{\#,k}$ are full rank. The rank of $Z^{\#,k}$ cannot exceed $L(\tau + \theta)$. ■

Annihilating and regrouping rows ultimately produces $H^{\#,*}$. Figure 5 displays a potential terminal tableau for the example model. There are many potential choices for annihilating the rows of the H matrices, but they all produce linear systems imposing equivalent constraints on the trajectories of the model variables.

The following corollary indicates that models with unique steady-states always terminate with non singular $\mathcal{H}_{\theta}^{\#,*}$.

Corollary 1 If $(\sum_{i=-\tau}^{\theta} H_i)$ is non singular then

1. \mathcal{H} is full rank.
2. The origin is the unique steady state of equation 6.
3. there exists a sequence of elementary row operations that transforms \mathcal{H} into \mathcal{H}^*

Proof Suppose \mathcal{H} Then there is a non zero vector $V \ni V\mathcal{H} = 0$. Consequently,

$$[V_{-\tau} \dots V_{\theta}] \mathcal{H} \begin{bmatrix} I & \dots & I \\ \vdots & \dots & \vdots \\ I & \dots & I \end{bmatrix} = 0$$

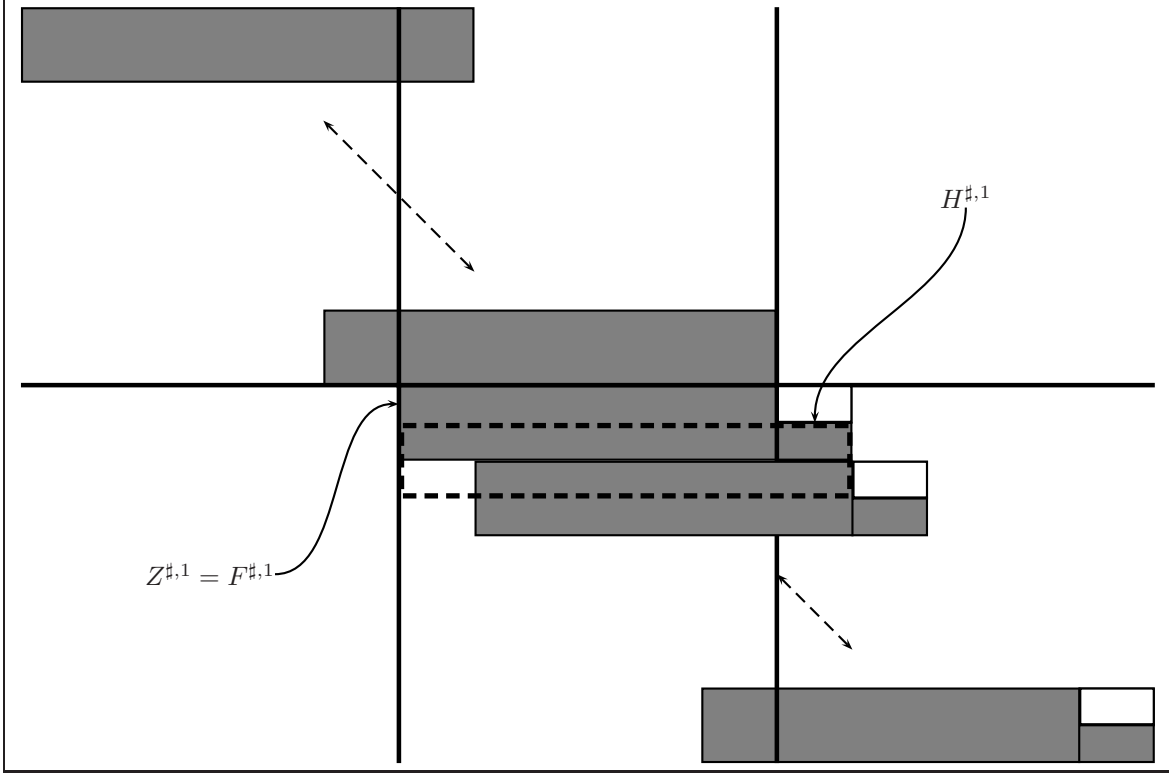


Figure 4: Matrix Tableau Characterization of Algorithm: Forward Row Annihilation

and

$$V_i \left(\sum_{j=-\tau}^{\theta} H_j \right) = 0 \forall i$$

So that $(\sum_{i=-\tau}^{\theta} H_i)$ must be singular. ■

This paper presents algorithms for efficiently obtaining the normal form characterized in Figure 6.

THE EXAMPLE:

The initial matrix $H^{\#,0}$, for the example model with variable order $\{\epsilon, \nu, u, w, W\}$ is

$$H^{\#,0} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & \alpha & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\theta & 0 & 0 & -1 & 0 & 1 & 0 & -\gamma & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

We would like to express the time $t + 1$ variables in terms of the time t and $t - 1$ variables. If the sub-matrix corresponding to the $t + 1$ variables were non

singular we could immediately write:

$$\begin{bmatrix} \epsilon_{t+1} \\ \nu_{t+1} \\ u_{t+1} \\ w_{t+1} \\ W_{t+1} \end{bmatrix} = (H_{\theta}^{\#,0})^{-1} \begin{bmatrix} H_{-1}^{\#,0} & H_0^{\#,0} \end{bmatrix} \begin{bmatrix} \epsilon_{-t+1} \\ \nu_{-t+1} \\ u_{-t+1} \\ w_{-t+1} \\ W_{-t+1} \\ \epsilon_t \\ \nu_t \\ u_t \\ w_t \\ W_t \end{bmatrix}$$

Since $(H_{\theta}^{\#,0})$ is singular, we use equations dated subsequent to the present time period to construct a set of linear constraints where the leading block is non singular.

$$H^{\#,*} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\theta & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & \alpha & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

So that

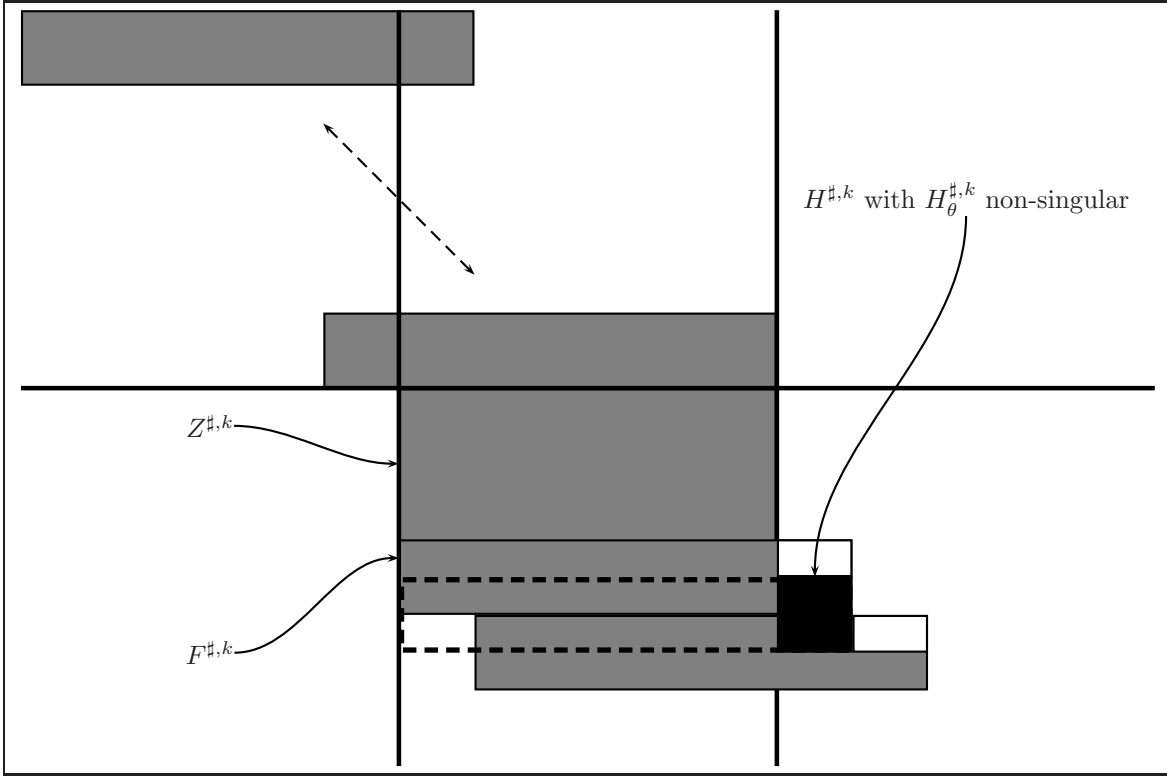


Figure 5: Matrix Tableau Characterization of Algorithm: Full Rank Leading Block

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2\gamma & \rho & 2\gamma & -\gamma \\ 0 & 0 & 0 & 0 & 0 & 0 & -4 & 4\alpha & 3 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 & 2\alpha & 2 & -1 \end{bmatrix} \quad (5)$$

One can adapt the algorithm to compute

$$x_{-\tau} = \Gamma^b \begin{bmatrix} x_{\theta} \\ \vdots \\ x_{-\tau+1} \end{bmatrix}$$

Section 6.1 explains the utility of exploiting the auxiliary initial conditions(shift left equations) of $Z^{b,*}$.

Where with $(H_{\theta}^{b,*})^{-1}$ non-singular:

$$\Gamma^b = (H_{\theta}^{b,*})^{-1} \begin{bmatrix} H_{-\tau}^{b,*} & \dots & H_{-1}^{b,*} \end{bmatrix}$$

It is possible to express x_{τ} in terms of leads of the x 's. Figure 6 provides a graphical representation of the normal form associated with computing

these expressions. The State Space Reduction(see Section 6.1) step of the algorithm exploits the constraints generated by constructing the non singular block in both the forward and backward direction.

2.2 Invariant Space Calculations

In order to compute solutions to equation 6 that converge, one must rule out explosive trajectories. Blanchard and Kahn(Blanchard and Kahn, 1980) used eigenvalue and eigenvector calculations to characterize the space in which the solutions must lie. The AIM algorithm exploits biorthogonality and the computation of left invariant space vectors to constrain the trajectories so that the solution cannot explode.

The asymptotic analysis exploits the fact that each left eigenvectors is orthogonal to each right eigenvector associated with roots of different value. Since vectors in the left invariant space associated with roots outside the unit circle are orthogonal to right eigenvectors associated with roots inside the unit circle, a given state vector that is part of a convergent trajectory must be orthogonal to each of these left invariant space vectors.

Theorem 2 Consider a left invariant space and

a right invariant space with no eigenvalues in common. Suppose V_1 spans the left invariant space and W_2 spans the right invariant space.

$$\begin{aligned} V_1 A &= T_1 V_1 \\ A W_2 &= W_2 T_2 \end{aligned}$$

With eigenvalues of $T_1 \neq T_2$. Then $V_1 W_2 = 0$

Proof A right eigenvector x_i and a left-eigenvector y_j corresponding to distinct eigenvalues λ_i and λ_j are orthogonal. (Noble, 1969) The left invariant space vectors are linear combination of the left eigenvectors and the right invariant space vectors are a linear combination of the right eigenvectors of the transition matrix raised to some finite power.

$$\begin{aligned} V_1 &= \beta_1 \begin{bmatrix} y_1 \\ \vdots \\ y_J \end{bmatrix} \\ W_2 &= [x_1 \quad \dots \quad x_K] \alpha_2 \\ V_1 W_2 &= \beta_1 \begin{bmatrix} y_1 \\ \vdots \\ y_J \end{bmatrix} [x_1 \quad \dots \quad x_K] \alpha_2 = 0 \end{aligned}$$

■

Theorem 3 Let $\{x_t^{conv}\}$, $t = -\tau, \dots, \infty$ be a non explosive solution satisfying equation 6. Let A be the state space transition matrix for equation 6 and V be a set of invariant space vectors spanning the invariant space associated with roots of A of magnitude bigger than 1. Then for $t = 0, \dots, \infty$

$$V \begin{bmatrix} x_{t-\tau}^{conv} \\ \vdots \\ x_{t+\theta-1}^{conv} \end{bmatrix} = 0$$

Proof Using W , the left generalized eigenvectors of A , one can employ the Jordan Canonical Form of A to write

$$W^H A = J W^H$$

so that

$$\begin{aligned} A^t &= (W^H)^{-1} J^t W^H \\ y_t &= A^t y_0 \\ W^H y_t &= J^t W^H y_0 \\ \lim_{t \rightarrow \infty} y_t = 0 &\Rightarrow \lim_{t \rightarrow \infty} W^H y_t = 0 \end{aligned}$$

Consequently,

$$W_i^H y_0 = 0 \forall i \ni |J_{ii}| > 1.$$

so that

$$V y_0 = \alpha W^H y_0 = 0$$

■

Thus, if A has roots with magnitude 1 then a path converges to a limit cycle (or fixed point) if and only if

$$V \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0$$

for some t .

Corollary 2 Let $\{x_t\}$, $t = -\tau, \dots, \infty$ be a solution satisfying equation 6. If A has no roots with magnitude 1 then the path converges to the unique steady state if and only if

$$V \begin{bmatrix} x_{t-\tau} \\ \vdots \\ x_{t+\theta-1} \end{bmatrix} = 0$$

for some t .

Proof

$$W_i^H y_0 = 0 \forall i \ni |J_{ii}| > 1.$$

means y_t

$$\begin{aligned} J_{ii} &\neq 1. \\ y_t &= A^t y_0 \end{aligned}$$

■

All convergent solutions converge to the same fixed point.

Combining V and Z^\sharp completely characterizes the space of stable solutions satisfying the linear system.

Theorem 4 Let

$$Q = \begin{bmatrix} Z^\sharp \\ V \end{bmatrix} = [Q_L \quad Q_R]$$

The existence of convergent solutions depends on the magnitude of the rank of the augmented matrix

$$r_1 = \text{rank} \left(\begin{bmatrix} I & 0 & x_{data} \\ Q_L & Q_R & 0 \end{bmatrix} \right)$$

and

$$r_2 = \text{rank} \left(\begin{bmatrix} I & 0 \\ Q_L & Q_R \end{bmatrix} \right)$$

and $L(\tau + \theta)$, the number of unknowns.

1. If $r_1 > r_2$ there is no nontrivial convergent solution
2. If $r_1 = r_2 = L(\tau + \theta)$ there is a unique convergent solution
3. If $r_1 = r_2 < L(\tau + \theta)$ the system has an infinity of convergent solutions

Proof The theorem applies well known results on existence and uniqueness of solutions to linear equation systems (Noble, 1969). If $\mathcal{M}_2 = \begin{bmatrix} x_{data} \\ 0 \end{bmatrix}$ does not lie in the column space of $\mathcal{M}_1 = \begin{bmatrix} I & 0 \\ Q_L & Q_R \end{bmatrix}$, then there is no solution. If \mathcal{M}_2 lies in the column space of \mathcal{M}_1 and the latter matrix is full rank, then there is a unique solution. If \mathcal{M}_2 lies in the column space of \mathcal{M}_1 and the latter matrix is not full rank, then there are multiple solutions. ■

The first set of equations come from the explicit initial conditions. The second set of equations come from the equations in equation system 6 which do not appear in the transformed system of Equation 4 but must nevertheless be satisfied. The last set of equations come from the constraint that the solutions are non-explosive. There are three cases.

Corollary 3 When Q has $L\theta$ rows, Q_R is square. If Q_R is non-singular, the system has a unique solution and

$$\begin{bmatrix} B \\ B_2 \\ \vdots \\ B_\theta \end{bmatrix} = Q_R^{-1} Q_L$$

If Q_R is singular, the system has an infinity of solutions.

Corollary 4 When Q has fewer than $L\theta$ rows, The system has an infinity of solutions.

Corollary 5 When Q has more than $L\theta$ rows, The system has a unique nontrivial solution only for specific values of x_{data}

Algorithm 2

```

1 Given  $V, Z^{\#, *}$ ,
2 funct  $\mathcal{F}_2(A, Z^{\#, *})$ 
3   Compute  $V$ , the vectors spanning the left
4   invariant space associated with eigenvalues
5   greater than one in magnitude
6    $Q := \begin{bmatrix} Z^{\#, *} \\ V \end{bmatrix}$ 
7   return  $Q$ 
8 .

```

Algorithm 3

```

1 Given  $Q$ ,
2 funct  $\mathcal{F}_3(Q)$ 
3   cnt = noRows( $Q$ )
4   return  $\begin{cases} \{Q, \infty\} & \text{cnt} < L\theta \\ \{Q, 0\} & \text{cnt} > L\theta \\ \{Q, \infty\} & (Q_R \text{ singular}) \\ \{B = -Q_R^{-1} Q_L, 1\} & \text{otherwise} \end{cases}$ 
5 .

```

THE EXAMPLE:

For the example model

$$Q = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 \\ 0 & 0 & -\theta & 0 & 0 & -1 & 0 & 1 & 0 & -\gamma \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \phi_4 & 0 & 0 & 0 & 2 & \phi_5 & \phi_6 & 1 \end{bmatrix}$$

where the ϕ 's are defined in Table 1

For the example model

$$B = \frac{1}{\phi_7} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \theta - \gamma\phi_4 + 2\theta\phi_6 & \gamma\phi_6 & 0 \\ 0 & 0 & -2(\phi_4 + \theta\phi_5) & -1 - \gamma\phi_5 & 0 \\ 0 & 0 & -\phi_4 - \theta\phi_5 & \phi_6 & 0 \end{bmatrix}$$

The formulae in Table 2 indicates that the number of floating operations depends on the size of A , the dimension of the dominant invariant space, $k-1$, and the relative sizes of the roots. Convergence is slower and the number of floating point operations larger if the magnitude of roots just inside the dominant space is close to the magnitude of roots just outside the dominant space.

For a given saddle point problem, the problem specification dictates the dimension of the dominant

$$\begin{aligned}
\phi_7 &= 1 + \gamma \phi_5 + 2 \phi_6 \phi_6 = \frac{-2 \left(18 \alpha \gamma + (-1 + \rho)^2 + (2 + \rho) \phi_2^{\frac{1}{3}} + \phi_2^{\frac{2}{3}} \right)}{\phi_3} \\
\phi_5 &= - \frac{-1 - 3 \alpha \gamma + 3 \rho + 21 \alpha \gamma \rho - 3 \rho^2 + \rho^3 + \phi_1 + \frac{108 \alpha^2 \gamma^2 + (-1 + \rho)^4 + 6 \alpha \gamma (-1 + \rho) (1 + 5 \rho) + 2 (-1 + \rho) \phi_1 + (12 \alpha \gamma + (-1 + \rho)^2) \phi_2^{\frac{1}{3}}}{\gamma \phi_3 \phi_2^{\frac{1}{3}}} \\
\phi_4 &= - \frac{3 \alpha \gamma \left(216 \alpha^2 \gamma^2 - 4 (-1 + \rho)^3 + 9 \alpha \gamma (1 + (-14 + \rho) \rho) \right) + \phi_1^2}{\gamma \phi_2^{\frac{2}{3}} \phi_3} \\
\phi_3 &= 18 \alpha \gamma + (-1 + \rho)^2 + (5 + \rho) \phi_2^{\frac{1}{3}} + \phi_2^{\frac{2}{3}} \\
\phi_2 &= (-1 + \rho)^3 + 27 \alpha \gamma (1 + \rho) + 3 \phi_1 \\
\phi_1 &= \sqrt{- \left(\alpha \gamma \left(216 \alpha^2 \gamma^2 - 4 (-1 + \rho)^3 + 9 \alpha \gamma (1 - 14 \rho + \rho^2) \right) \right)}
\end{aligned}$$

Table 1: Q Matrix ϕ Definitions

space of interest, and the model parameters determine the root configuration. One can, however, streamline the computations by computing a similar matrix and carrying out the eigenspace calculations on a submatrix of the similar matrix and reconstructing the invariant space vectors for the original matrix. The algorithms in Section 6.1 show how to construct this similarity transformation and how to recover the invariant space vectors for the original problem.

3 Inhomogeneous Solution

$$\sum_{i=-\tau}^{\theta} H_i x_{t+i} = 0, t \geq 0 \quad (6)$$

with initial conditions, if any, given by constraints of the form

$$x_t = x_t^{data}, t = -\tau, \dots, -1 \quad (7)$$

where both τ and θ are non-negative, and x_t is an L dimensional vector with

$$\lim_{t \rightarrow \infty} x_t = 0 \quad (8)$$

Modelers can augment the homogeneous linear perfect foresight solutions with particular solutions characterizing stochastic elements or exogenous fac-

tors. Suppose

$$\begin{aligned}
\sum_{i=-\tau}^{\theta} H_i x_{t+i} &= \Psi z_t, t \geq 0 \\
\lim_{t \rightarrow \infty} x_t &= x^* \\
\lim_{t \rightarrow \infty} z_t &= z^*
\end{aligned}$$

with

$$\sum_{i=-\tau}^{\theta} H_i x^* = \Psi z^*$$

The Q matrix plays a fundamental role in all subsequent calculations. This set of linear restrictions codifies all the constraints needed to guarantee the existence and uniqueness of saddle point solutions. These solutions provide a foundation for characterizing many other useful model solutions and properties.

Then one can use Q to write solutions in the form

$$(x_t - x^*) =$$

$$\begin{aligned}
& [B_L \quad B_R] \begin{bmatrix} (x_{t-\tau} - x^*) \\ \vdots \\ (x_{t-1} - x^*) \end{bmatrix} + \\
& [0 \quad \dots \quad 0 \quad I] \sum_{s=0}^{\infty} (F^s \begin{bmatrix} 0 \\ \Phi \Psi z_{t+s} \end{bmatrix})
\end{aligned}$$

Where

$$B = \begin{bmatrix} B_L & B_R \\ \vdots & \vdots \\ B_L^\theta & B_R^\theta \end{bmatrix} = Q_R^{-1} Q_L$$

$$\begin{aligned}
\Phi &= (H_0 + H_+ \begin{bmatrix} B_R \\ \vdots \\ B_R^\theta \end{bmatrix})^{-1} & H_- + ([0 \ H_0] + H_+ \begin{bmatrix} B \\ \vdots \\ B^\theta \end{bmatrix})\tilde{B} &= 0 \\
F &= \begin{bmatrix} 0 & I & & \\ \vdots & & \ddots & \\ 0 & & & I \\ -\Phi H_+ \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I \end{bmatrix} & -\Phi H_+ \begin{bmatrix} 0 \\ \vdots \\ I \\ B_R \end{bmatrix} & \dots & -\Phi H_+ \begin{bmatrix} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{bmatrix} \end{bmatrix} & H_- + ([0 \ H_0] + H_+ \begin{bmatrix} B_L & B_R \\ \vdots & \vdots \\ B_L^\theta & B_R^\theta \end{bmatrix})\tilde{B} &= 0 \\
& & \begin{bmatrix} B_L & B_R \\ \vdots & \vdots \\ B_L^\theta & B_R^\theta \end{bmatrix} \tilde{B} &= \begin{bmatrix} 0 & B_L \\ \vdots & \vdots \\ 0 & B_L^\theta \end{bmatrix} + \begin{bmatrix} B_R \\ \vdots \\ B_R^\theta \end{bmatrix} B
\end{aligned}$$

When

$$z_{t+1} = \Upsilon z_t$$

$$H_- + H_+ \begin{bmatrix} 0 & B_L \\ \vdots & \vdots \\ 0 & B_L^\theta \end{bmatrix} + (H_0 + H_+ \begin{bmatrix} B_R \\ \vdots \\ B_R^\theta \end{bmatrix})B = 0$$

the infinite sum simplifies to give

$$(x_t - x^*) = [B_L \ B_R] \begin{bmatrix} (x_{t-\tau} - x^*) \\ \vdots \\ (x_{t-1} - x^*) \end{bmatrix} + \vartheta z_t$$

$$\Phi = (H_0 + H_+ \begin{bmatrix} B_R \\ \vdots \\ B_R^\theta \end{bmatrix})^{-1}$$

where

$$\Phi H_- + \Phi H_+ \begin{bmatrix} 0 & B_L \\ \vdots & \vdots \\ 0 & B_L^\theta \end{bmatrix} + B = 0$$

$$vec \vartheta = [0 \ \dots 0 \ I] (I - \Upsilon^T \otimes F)^{-1} vec \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Phi \Psi \end{bmatrix}$$

$$\begin{aligned} \tilde{B} &= \begin{bmatrix} 0 & I \\ B \end{bmatrix} \\ B^{k+1} &= B^k \tilde{B} \end{aligned}$$

$$[H_- \ H_0 \ H_+] \begin{bmatrix} I & & 0 \\ & \ddots & \vdots \\ 0 & \dots & 0 \\ & & B \\ 0 & & \vdots \\ & & B^\theta \end{bmatrix} \begin{bmatrix} x_{t+s-\tau} \\ \vdots \\ x_{t+s} \end{bmatrix} = \Psi z_{t+s}$$

$$[H_- \ H_0 \ H_+] \begin{bmatrix} I \\ B \\ \vdots \\ B^{\theta+1} \end{bmatrix} = 0$$

$$\Phi \Psi z_{t+s} =$$

$$\Phi ([H_- \ 0] + [0 \ H_0] + \begin{bmatrix} 0 & B_L \\ \vdots & \vdots \\ 0 & B_L^\theta \end{bmatrix} H_+ \begin{bmatrix} B_R \\ \vdots \\ B_R^\theta \end{bmatrix}) \begin{bmatrix} x_{t+s-\tau} \\ \vdots \\ x_{t+s} \end{bmatrix}$$

$$[H_- \ H_0 \ H_+] \begin{bmatrix} I \\ \begin{bmatrix} 0 & I \\ B \\ \vdots \\ B^\theta \end{bmatrix} \tilde{B} \end{bmatrix} = 0$$

$$[-B \ I] \begin{bmatrix} x_{t+s-\tau} \\ \vdots \\ x_{t+s} \end{bmatrix} = \Phi \Psi z_{t+s}$$

$$x_{t+s} = B \begin{bmatrix} x_{t+s-\tau} \\ \vdots \\ x_{t+s-1} \end{bmatrix} + \Phi \Psi z_{t+s}$$

$$\begin{aligned}
& \Phi([H_- \ 0] + [0 \ H_0] + \\
& \left[\begin{array}{cc} 0 & B_L \\ H_+ & \vdots \\ 0 & B_L^\theta \end{array} \right] H_+ \left[\begin{array}{c} B_R \\ \vdots \\ B_R^\theta \end{array} \right] \left[\begin{array}{c} x_{t+s-\tau-1} \\ \vdots \\ x_{t+s-1} \end{array} \right] + \\
& \Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right] \Phi \Psi z_{t+s} = 0
\end{aligned}$$

$$[-B \ I] \left[\begin{array}{c} x_{t+s-\tau-1} \\ \vdots \\ x_{t+s-1} \end{array} \right] + \Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right] \Phi \Psi z_{t+s} = 0$$

$$x_{t+s-1} = B \left[\begin{array}{c} x_{t+s-\tau-1} \\ \vdots \\ x_{t+s-2} \end{array} \right] - \Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right] \Phi \Psi z_{t+s}$$

$$x_{t+s-i} = B \left[\begin{array}{c} x_{t+s-\tau-1} \\ \vdots \\ x_{t+s-2} \end{array} \right] + (-\Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right])^i \Phi \Psi z_{t+s}$$

And by superposition

$$x_t = B \left[\begin{array}{c} x_{t-\tau} \\ \vdots \\ x_{t-1} \end{array} \right] + \sum_{s=0}^{\infty} ((-\Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right])^s \Phi \Psi z_{t+s})$$

For constants (ie. inhomogeneous equation) require

$$\begin{aligned}
& \left(\sum_{i=-\tau}^{\theta} H_i \right) x^* = 0 \\
& (x_t - x^*) = B(x_{t-1} - x^*) + \sum_{s=0}^{\infty} F^s c \\
& \vartheta = (I - I \otimes F) \text{vec}(\phi) \\
& (I - B)x^* = \vartheta c
\end{aligned}$$

Algorithm 4

- 1 Given $H, \Psi,$
- 2 funct $\mathcal{F}_4(H, \Psi, B)$
- 3 $\Phi = (H_0 + H_+ \left[\begin{array}{c} B_R \\ \vdots \\ B_R^\theta \end{array} \right])^{-1}$

$$\begin{aligned}
& 4 \quad F = \left[\begin{array}{ccc} 0 & I & \\ \vdots & & \ddots \\ 0 & & \\ -\Phi H_+ \left[\begin{array}{c} 0 \\ \vdots \\ 0 \\ I \end{array} \right] & -\Phi H_+ \left[\begin{array}{c} 0 \\ \vdots \\ I \\ B_R \end{array} \right] & \dots \\ \dots & \dots & -\Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right] \end{array} \right] \\
& 5 \quad \text{return}(\Phi, F) \\
& 6 \quad .
\end{aligned}$$

4 Observable Structure

4.1 “Aligned Information Set” Expectation Error Calculations

These saddle point calculations often arise in rational expectations models. To compute the error term for estimation of the coefficients of these models, one must commit to a particular information set. Two typical choices are t and $t-1$ period expectations.

$$\begin{aligned}
\epsilon_t &= [H_{-\tau} \dots H_0] \left[\begin{array}{c} x_{t-\tau}^{data} \\ \vdots \\ x_t^{data} \end{array} \right] + \\
& [H_1 \dots H_\theta] \left[\begin{array}{c} E[x_{t+1}|I_t] \\ \vdots \\ E[x_{t+\theta}|I_t] \end{array} \right] - \Psi z_t
\end{aligned}$$

Suppose $\exists K^* \ni x_{t-k} \in I_t, \forall k \geq k^*$
then set,

$$\left[\begin{array}{c} E[x_{t+1}|I_t] \\ \vdots \\ E[x_{t+\theta}|I_t] \end{array} \right] = \left[\begin{array}{c} B \\ \vdots \\ B^\theta \end{array} \right] \tilde{B}^{k^*} \left[\begin{array}{c} x_{t-\tau+1-k^*}^{data} \\ \vdots \\ x_{t-k^*}^{data} \end{array} \right] +$$

$$\left[\begin{array}{c} \sum_{s=0}^{\infty} ((-\Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right])^s \Phi \Psi z_{t+s+1}) \\ \vdots \\ \sum_{s=0}^{\infty} ((-\Phi H_+ \left[\begin{array}{c} I \\ B_R \\ \vdots \\ B_R^{\theta-1} \end{array} \right])^s \Phi \Psi z_{t+s+\theta}) \end{array} \right]$$

$$\epsilon_t = S \left[\begin{array}{c} x_{t-\tau+1-\max(1, k^*)}^{data} \\ \vdots \\ x_t^{data} \end{array} \right] - \Psi z_t$$

where

$$S = \begin{bmatrix} 0_{L \times L \max(1, k^*)} & H_{-\tau} & \dots & H_0 \end{bmatrix} + \begin{bmatrix} [H_1 \dots H_\theta] \begin{bmatrix} B \\ \vdots \\ B^\theta \end{bmatrix} \\ \tilde{B}^{k^*} & 0_{L \times L \max(1, k^*)} \end{bmatrix}$$

Note that for $k^* \geq 1$

$$\frac{\partial \epsilon_t}{\partial x_t^{data}} = H_0$$

and for $k^* = 0$

$$\frac{\partial \epsilon_t}{\partial x_t^{data}} = H_0 + [H_1 \dots H_\theta] \begin{bmatrix} B \\ \vdots \\ B^\theta \end{bmatrix} = \Phi^{-1}$$

Algorithm 5

```

1 Given  $H, \Psi, B$ 
2 funct  $\mathcal{F}_5(H, \Psi, B)$ 
3   return( $\varrho, \varphi$ )
4 .

```

5 Unit Roots or Constant Growth

Theorem 5 Suppose we have for some scalar valued variable y_t ,

$$\sum_{i=-\tau}^{\theta} H_i x_{t+i} - G y_t$$

$$y_t = g y_{t-1}$$

then solving the system

$$\sum_{i=-\tau}^{\theta} g^i H_i \bar{x}_{t+i}$$

determines the evolution of x_t

$$\bar{x}^* = \left(\sum_{i=-\tau}^{\theta} g^i H_i \right)^{-1} G$$

$$Q(\bar{x}_t - \bar{x}_t^*) = 0$$

$$x_{t+k} = g^{k+1} y_{t-1} \bar{x}_{t+k}$$

The scalar g may be bigger than one in magnitude.

6 Implementation

The following sections describe the symbolic algebra and numeric algebra implementations of the routines. The design of the numeric algebra routines must take care due to the computation with finite precision arithmetic. These routines rely on unitary matrices to limit rounding error. The symbolic algebra routines can often use infinite precision arithmetic and often work faster avoiding routines that renormalize linear equations.

6.1 State Space Reduction

One can use vectors associated with zero eigenvalue and the vectors associated with unit roots to reduce the size of the eigenspace calculation. State space reduction renders the transition matrix full rank and also typically reduces the dimension by 2/3. Consequently, the computational burden reduces to 1/9 that of the original problem. Thus, given A, Z one can obtain eigenvalues and invariant space vectors from \bar{A} and then recover invariant space vectors, V , for A .

Algorithm 6

```

1 Given  $\Gamma^{\sharp, *}, Z^{\sharp, *}, Z^{b, *}$ ,
2 compute vectors spanning the left invariant
3 space associated with large eigenvalues
4 funct  $\mathcal{F}_6(\Gamma^{\sharp, *}, Z^{\sharp, *}, Z^{b, *})$ 
5    $A := \begin{bmatrix} 0 & I \\ \Gamma^{\sharp} & \end{bmatrix}$ 
6    $\{\bar{A}, \Pi, J_0\} = \text{stateSpaceReducer}(A, Z^{\sharp, *}, Z^{b, *})$ 
7    $\{\bar{V}, M\} := \text{leftInvariantSpaceVectors}(\bar{A})$ 
8    $V = \text{stateSpaceExpander}(\bar{V}, M, \Pi, J_0)$ 
9 .

```

6.1.1 Zero Invariant Space

Theorem 6 The Z_*^{\sharp}, Z_*^b span the invariant space associated with zero eigenvalue.

$$\begin{bmatrix} Z_*^{\sharp} \\ Z_*^b \end{bmatrix} A^{L(\tau+\theta)} = 0$$

Proof

$$A^{\tau+\theta} = \begin{bmatrix} \Gamma_1 \\ \Gamma_1^2 \\ \vdots \\ \Gamma_1^{\tau+\theta-1} \end{bmatrix} \begin{bmatrix} \Gamma_{\tau+\theta} & \Gamma_1 \end{bmatrix}$$

■

Consider the Row Echelon form

$$Z = R_Z^T \begin{bmatrix} Z_*^\# \\ Z_*^\flat \end{bmatrix}$$

The rank deficiency of this matrix is equal to the rank of the minimal dimension transition matrix for calculating the eigenvalues and eigenvectors of the full problem.

To construct the state space reduction, extend Z to a basis for the whole space. (Noble, 1969, page 119)

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix}$$

We have then

$$ZA = J_0 Z$$

and $\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix}$ non-singular.

Consequently, after defining $Z_l, Z_r, \bar{Z}_l, \bar{Z}_r$ from

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix}^{-1} = \begin{bmatrix} Z_l & Z_r \\ \bar{Z}_l & \bar{Z}_r \end{bmatrix}$$

we have

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix} A \begin{bmatrix} Z_l & Z_r \\ \bar{Z}_l & \bar{Z}_r \end{bmatrix} = \begin{bmatrix} J_0 & 0 \\ \Pi & \bar{A} \end{bmatrix}$$

with

$$\begin{aligned} \Pi &= (\bar{Z} A \begin{bmatrix} Z_l \\ \bar{Z}_l \end{bmatrix}) \\ \bar{A} &= (\bar{Z} A \begin{bmatrix} Z_r \\ \bar{Z}_r \end{bmatrix}) \end{aligned}$$

Theorem 7 Suppose

$$Y \bar{A} = M Y$$

so that Y spans the invariant space associated with the eigenvalues of M , one can compute X with

$$\begin{bmatrix} X & Y \end{bmatrix}$$

spans the dominant invariant space of A . From

$$vec(X) = ((I \otimes M) - (J_0^T \otimes I))^{-1} (\Pi^T \otimes I) vec(Y)$$

Proof

Consequently, if we compute

$$Y \bar{A} = M Y$$

so that Y spans the invariant space associated with the eigenvalues of M , one can compute X with

$$X J_0 + Y \Pi = M X$$

So that

$$\begin{bmatrix} X & Y \end{bmatrix}$$

spans the dominant invariant space of A . Solving for X we have

$$vec(X) = ((I \otimes M) - (J_0^T \otimes I))^{-1} (\Pi^T \otimes I) vec(Y)$$

since

$$vec(ABC) = (C^T \otimes A) vec(B)$$

$$V = \begin{bmatrix} X & Y \end{bmatrix}$$

■

Algorithm 7

- 1 Given h, H ,
- 2 asymptotic stability constraints
- 3 **func** $\mathcal{F}_7(V, Z^{\#, *})$
- 4 •

THE EXAMPLE:

Collecting the auxiliary constraints generated by the auto regression phase of the algorithm for the example model one has:

$$\begin{bmatrix} Z_*^\# \\ Z_*^\flat \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} & 1 \\ 0 & 0 & -\theta & 0 & 0 & -1 & 0 & 1 & 0 & -\gamma \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & \alpha & 1 & -\frac{1}{2} & 0 & 0 & 0 & 0 & -\frac{1}{2} \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

One can extend the basis to get a non singular matrix.²

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -\frac{1}{\theta} & 0 & \frac{\gamma}{\theta} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \frac{-2\alpha}{\theta} & 2 & \frac{2\alpha\gamma-3\theta}{\theta} \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A = \begin{bmatrix} \rho & 2\gamma & -\gamma \\ 4\alpha & 3 & -2 \\ 2\alpha & 2 & -1 \end{bmatrix}$$

For this model, we expect one large root. Consequently, $M = [\lambda_L]$, a 1×1 matrix.

$$\Pi = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -2\gamma \\ 0 & 0 & 0 & 0 & 0 & 0 & -4 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2 \end{bmatrix}$$

$$J_0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -8 + \frac{4\alpha\gamma}{\theta} - \frac{2(2\alpha\gamma-3\theta)}{\theta} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$((I \otimes M) - (J_0^T \otimes I))^{-1}(\Pi^T \otimes I) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-2\gamma}{\lambda_L} & \frac{-4}{\lambda_L} & \frac{-2}{\lambda_L} \end{bmatrix}$$

6.2 Implementation of State Space Transition Matrix Computation

One can calculate these matrix transformation on a single copy of the $H^{\#,k}$ saving on unnecessary repetition of the coefficients present in the full tableau. Furthermore, for many models, leading rows of specific equations may be zero. One should immediately fill in initial rows of $Z^{\#}$ with these obvious shift right equations.

²The top rows are just the row-echelon form of the $Z^{\#,*}$, $Z^{b,*}$ vectors.

6.2.1 Symbolic Algebra Implementation

In symbolic algebra packages using exact arithmetic, there is no need to worry about roundoff error and one can compute the row-echelon form of an augmented $H^{\#,k}$ to determine rank and to obtain matrices for annihilating its rows (Noble, 1969, page 119). By augmenting the matrix with columns of the identity matrix, one obtains a full rank matrix that either inverts $H^{\#,k}$ or zeroes some of its rows.

$$\text{rowEchelon}([H^{\#,k} \ I]) = [R \ U^{\#,k}]$$

6.2.2 Numeric Implementation

For numerical computations one should use the numerically stable unitary matrices generated by QR Decomposition with column pivoting to annihilate rows. Note that one can facilitate subsequent applications of the routines for QR Decomposition by putting the shifted rows corresponding to the annihilated rows at the top of the matrix. Then, $H^{\#,k+1}$ will be closer to Hessenberg form. QR Decomposition works faster on Hessenberg matrices. Indeed if only one row was annihilated, the new matrix will be in Hessenberg form.

With the final QR decomposition in hand, it will not be necessary to invert $H_{\theta}^{\#,*} = Q^{\#,*}R^{\#,*}$. Since $R^{\#,*}$ will be upper triangular and $(Q^{\#,*})^{-1} = (Q^{\#,*})^T$, we can back solve and multiply to get the columns of Γ .

The construction of the companion matrix leads to a very sparse matrix. Note the sparsity of the transition matrix for the example problem in Equation 5. Section 6.1 shows how to construct a reduced dimensions non-singular matrix for studying the asymptotic dynamics of the linear system.

Floating point operation counts in Section 6.1 demonstrate the utility of computing these auxiliary constraints in both directions. Section 6.1 shows how to use the vectors, that arise from computing the autoregressive representations, $Z^{\#,*}$, $Z^{b,*}$, to construct a similarity transformation which makes it possible to compute the non zero eigenvalues and eigenvectors of the transition matrix with much smaller matrices. This can be very important for large sparse models since, as Section 2.2 shows, the computational burden of the asymptotic analysis increases with the cube of the problem size.

6.3 Implementation of the Invariant Space Computation

Numerical routines for computing the invariant space order the eigenvalues large to small and compute invariant space vectors for the dominant invariant space

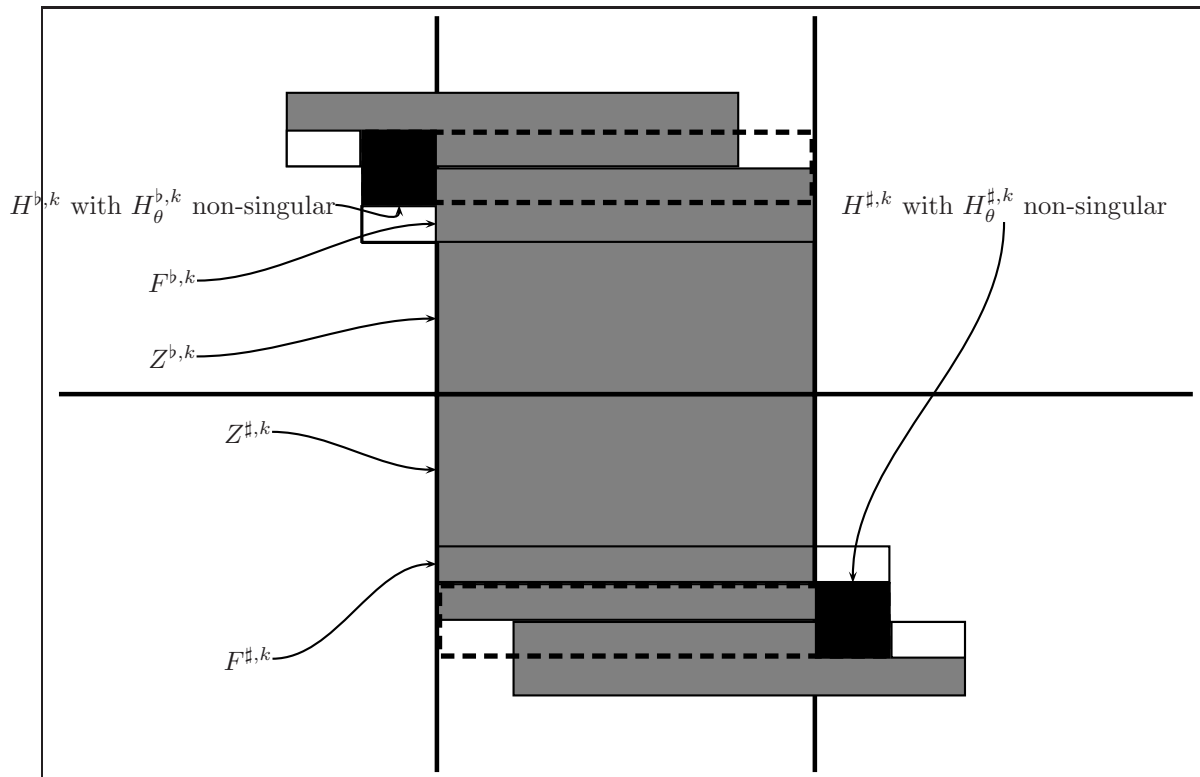


Figure 6: Matrix Tableau Characterization of Algorithm: Full Rank Trailing Block and Leading Block

(ie the invariant space associated with the largest eigenvalues). Invariant space vectors spanning the invariant space are generally easier to compute than eigenvectors. In subsequent calculations, one can choose the invariant space calculation dimension to be one more than the dimension of the required invariant space. Although root counting alone cannot discover all model anomalies, often a modeler knows enough about the problem to infer from the root count whether the model seems appropriate for further analysis. If this is the case, one can intervene after the Schur decomposition of the transition matrix to abort further calculations.

6.3.1 Symbolic Algebra Implementation

All is not lost when symbolic algebra packages fail to compute the entire set of eigenvectors in a reasonable amount of time. Symbolic algebra packages can often quickly obtain expressions for the eigenvalues of the matrices. Having identified which of these corresponds to the large eigenvalues, one can compute the relevant eigenvectors by computing vectors for the null space of $(A^T - \lambda I)$.

6.3.2 Numeric Implementation

The LAPACK routine DGEES computes the real-Schur form and the matrix of Schur vectors forming an orthonormal basis for the dominant invariant subspace of a real matrix A. Given an integer m, the algorithm computes a matrix V with m orthonormal columns and a real quasi triangular matrix T of order m such that

$$VA = TV$$

The eigenvalues of T are approximations to the m largest eigenvalues of A. The columns of V span the invariant subspace corresponding to those eigenvalues of A. Alternatively, the routine dnaupd in ARPACK applies Krylov methods for computing the invariant space vectors.

6.4 Implementation of the State Space Reduction

For symbolic computation, one can compute the row-echelon form and use the technique outlined in Noble to extend the rows to a basis. The most computationally intensive step will involve symbolic computation of the inverse.

For numerical calculations one can avoid a costly matrix inversion by computing the QR Decomposition

$$Q_Z R_Z = [(Z^{\sharp,*})^T \quad (Z^{b,*})^T]$$

and to use the orthonormal columns of Q^T for Z . Compute

$$\begin{aligned} Z^T &= Q_Z R_Z \\ \bar{Z}^T &= Q_Z Q_Z^T - I \\ \bar{Z}^T &= Q_{\bar{Z}} R_{\bar{Z}} \\ \bar{Z} &= Q_{\bar{Z}}^T \end{aligned}$$

Thus, the transpose provides the inverse of the matrix.

Now

$$\begin{bmatrix} Z \\ \bar{Z} \end{bmatrix}^{-1} = \begin{bmatrix} Z \\ \bar{Z} \end{bmatrix}^T = \begin{bmatrix} Z_l & Z_r \\ \bar{Z}_l & \bar{Z}_r \end{bmatrix}$$

7 Numerical Experiments

7.1 Total Floating Point Operation Counts

The number of floating point operations provides one measure of the computational cost of the calculations:

Computing the state space transition matrix requires:

$$TMFLOPS = L^3 \left(\frac{4}{3} + 26 (\tau + \theta) + 3 (\tau + \theta)^2 \right)$$

The eigenvalue calculations require:

$$\begin{aligned} ESFLOPS &= \frac{14(-l + L(\tau + \theta))^3}{3} + \\ &12k(-l + L(\tau + \theta))(-k - l + L(\tau + \theta)) + \\ &- \left(k \left(2k^2 + 6k(l - L(\tau + \theta)) - 3(l - L(\tau + \theta))^2 \right) f\left(\frac{\lambda_k}{\lambda_{k-1}}\right) \right) \end{aligned}$$

The algorithms for reducing the eigenvalue calculations require:

$$\begin{aligned} REDFLOPS &= \\ &\frac{-2L(\tau + \theta) \left(3l^2 + 3lL(\tau + \theta) - 8L^2(\tau + \theta)^2 \right)}{3} \end{aligned}$$

The gain from applying the reductions are:

$$\begin{aligned} DELTAFLOPS &= \frac{1}{3} \left(\right. \\ &2(-18k^2l + 7l^3 - 18l^2L(\tau + \theta) + \\ &24lL^2(\tau + \theta)^2 - 8L^3(\tau + \theta)^3 - 18kl(l - 2L(\tau + \theta))) + \\ &\left. 3kl(2k - l + 2L(\tau + \theta)) f\left(\frac{\lambda_k}{\lambda_{-1+k}}\right) \right) \end{aligned}$$

The example model produces $4(N - 1)$ auxiliary initial conditions for the forward transition matrix, $4(N - 1) - 1 \forall N > 1$ auxiliary initial conditions for the reverse transition matrix so that the zero invariant subspace has dimensions $8(N - 1) - 1$. Consequently, the minimal dimension transition matrix has dimension $2N - 1$. We can carry out eigenvalue calculations on a matrix of size $2N - 1$ instead of $8(N - 1)$. This can significantly reduce computation time since the floating point operations increase with the square of the matrix dimension.³

With $\alpha = 2, \gamma = \frac{1}{10}, \theta = -\frac{1}{5}$, there are $(N - 1)$ roots larger than 1 in magnitude. Consequently, the algorithm determines that the model has a unique solution converging to the steady state from arbitrary initial conditions so long as Q_R^{-1} is full rank. For example, with $N = 2$ there are 1 roots larger than 1 in magnitude. 4 auxiliary initial conditions for the forward transition matrix, 3 auxiliary initial conditions for the reverse transition matrix for a total of 7 dimension for the zero invariant subspace. Consequently, the minimal dimension transition matrix has dimension 3.

The graph in Figure7 demonstrates that the benefits from problem reduction dominate the costs for all values of N and for all values of $\frac{\lambda_k}{\lambda_{k-1}}$. Increasing N increases the computational burden much more dramatically than changes in the rate of convergence.

References

Anderson, Gary (1997a). Continuous time application of the anderson-moore(aim) algorithm for imposing the saddle point property in dynamic models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://www.bog.frb.fed.us/pubs/oss/oss4/aimindex.html>.

³For $N = 1$, the larger matrix is of dimension N and one can use the auxiliary initial conditions to demonstrate that all the eigenvalues are less one in magnitude without eigenvalue calculations.

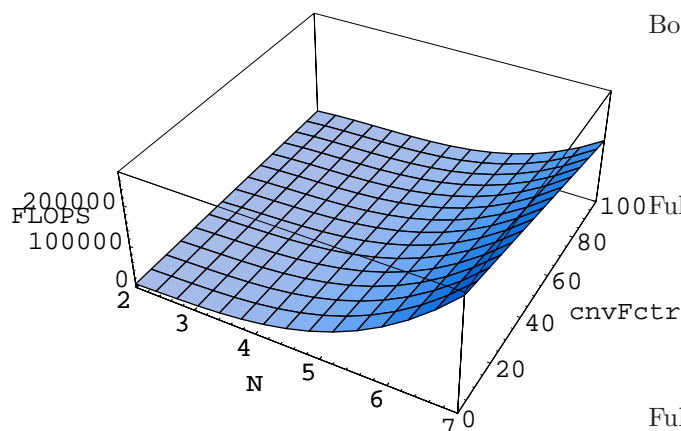


Figure 7: Reduction in Floating Point Operations as a Function of N

- Anderson, Gary (1997b). A parallel programming implementation of the anderson-moore(aim) algorithm. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://www.bog.frb.fed.us/pubs/oss/oss4/aimindex.html>.
- Anderson, Gary and George Moore (1983). An efficient procedure for solving linear perfect foresight models. Unpublished Manuscript, Board of Governors of the Federal Reserve System. Downloadable copies of this and other related papers at <http://www.bog.frb.fed.us/pubs/oss/oss4/aimindex.html>.
- Anderson, Gary and George Moore (1985). A linear algebraic procedure for solving linear perfect foresight models. *Economics Letters*.
- Andrew, Levin, John Williams and Volker Wieland (1998). Are simple rules robust under model uncertainty?. Seminar Paper.
- Binder, Michael and M. Hashem Peseran (1995). *Handbook of Applied Econometrics*. Chap. Multivariate Rational Expectations Models and Macroeconometric Modelling: A Review and Some New Results. Unknown.
- Blanchard, Olivier Jean and C. Kahn (1980). The solution of linear difference models under rational expectations. *Econometrica*.
- Bomfim, Antulio N. (1996). Forecasting the forecasts of others: Expectational heterogeneity and aggregate dynamics. Technical Report 1996-31. Federal Reserve Board, Finance and Economics Discussion Series.
- Fuhrer, Jeffrey C. (1994). Optimal Monetary Policy and the Sacrifice Ratio. In: *Goals, Guidelines, and Constraints Facing Monetary Policymakers*, (Jeffrey C. Fuhrer, Ed.). pp. 43–69. Federal Reserve Bank of Boston Conference Series No. 38.
- Fuhrer, Jeffrey C. (1996). Monetary Policy Shifts and Long-term Interest Rates. *Quarterly Journal of Economics* **111**, 1183–1209.
- Fuhrer, Jeffrey C. (1997a). Inflation/Output Variance Trade-offs and Optimal Monetary Policy. *Journal of Money Credit and Banking* **29**, No. 2, 214–234.
- Fuhrer, Jeffrey C. (1997b). The (Un)Importance of Forward-Looking Behavior in Price Specifications. *Journal of Money Credit and Banking* **29**, No. 3, 338–350.
- Fuhrer, Jeffrey C. (1997c). Towards a Compact, Empirically-Verified Rational Expectations Model for Monetary Policy Analysis. *Carnegie-Rochester Conference Series on Public Policy* p. forthcoming.
- Fuhrer, Jeffrey C. and Brian Madigan (1997). Monetary Policy When Interest Rates are Bounded at Zero. *Review of Economics and Statistics* **79**, No. 4, 573–585.
- Fuhrer, Jeffrey C. and George R. Moore (1995). Forward-Looking Behavior and the Stability of a Conventional Monetary Policy Rule. *Journal of Money Credit and Banking* **27**, No. 4, 1060–70.
- Fuhrer, Jeffrey C. and George R. Moore (1995a). Inflation Persistence. *Quarterly Journal of Economics* **110**, 127–159.
- Fuhrer, Jeffrey C. and George R. Moore (1995b). Monetary Policy Trade-Offs and the Correlation Between Nominal Interest Rates and Real Output. *American Economic Review* **85**, 219–239.
- Fuhrer, Jeffrey C. and Mark W. Hooker (1993). Learning About Monetary Regime Shifts in an

Overlapping Wage Contract Economy. *Journal of Economic Dynamics and Control* **17**, 531–553.

Fuhrer, Jeffrey C., George Moore and Scott Schuh (1995). A Comparison of GMM and Maximum-Likelihood Estimates of Quadratic Cost Inventory Models. *Journal of Monetary Economics* **35**, 115–157.

Golub, Gene H. and Charles F. van Loan (1989). *Matrix Computations*. Johns Hopkins.

Luenberger, David G. (1978). Time-invariant descriptor systems. *Automatica* **14**, 473–480.

Noble, Ben (1969). *Applied Linear Algebra*. Prentice-Hall, Inc.

Orphanides, Athanasios (1998). Monetary policy evaluation with noisy information. Technical Report 98-50. Finance and Economics Discussion Series.

Orphanides, Athanasios, David Small, David Wilcox and Volker Wieland (1997). A quantitative exploration of the opportunistic approach to disinflation. Technical Report 97-36. Finance and Economics Discussion Series, Board of Governors, Federal Reserve.

Sims, Christopher A. (1996). Solving linear rational expectations models. Seminar paper.

Zadrozny, Peter A. (1996). An eigenvalue method of undetermined coefficients for solving linear rational expectations models. Seminar Paper.

floating point operations to produce the autoregressive representation, Γ . Table 3 reports the floating point counts by type of operation. The rank of $H_\theta^{\sharp,0}$ can be at most L . For typical problems, the rank, after shifting obvious equations forward, is nearly L . There can be at most $L(\tau + \theta)$ shifted equations but this number is typically about equal to $L\theta$.

A.3 Eigenspace Calculation Floating Point Operation Counts

When there are l rows in $\begin{bmatrix} Z^{\sharp,*} \\ Z^{\flat,*} \end{bmatrix}$, and the problem requires a dominant invariant space of dimension k the DRRIT algorithm requires:

$$\begin{aligned} & \frac{14 L^3 (\tau + \theta)^3}{3} + \\ & L^2 \left(12 k (\tau + \theta)^2 - 14 l (\tau + \theta)^2 \right) + \\ & L \left(-12 k^2 (\tau + \theta) - 24 k l (\tau + \theta) + 14 l^2 (\tau + \theta) \right) + \\ & 12 k^2 l + 12 k l^2 - \frac{14 l^3}{3} + \\ & f\left(\frac{\lambda_k}{\lambda_{-1+k}}\right) \left(\frac{-2 k^3}{3} - 2 k^2 l + k l^2 + \right. \\ & \left. k + L^2 + (\tau + \theta)^2 + \right. \\ & \left. L + 2 k^2 (\tau + \theta) - 2 k l (\tau + \theta) \right) \end{aligned}$$

Table 4 presents details for components of the algorithm.

A Floating Point Operation Counts

A.1 General Floating Point Operation Counts

The following sections use the total number of floating point operations as a measure of the amount of computation that various algorithms require. Table 2 summarizes a number of results from Golub (Golub and van Loan, 1989) that appear in this paper.

A.2 Unconstrained Autoregression Floating Point Operation Counts

The AIM algorithm requires at most

$$L^3 \left(\frac{4}{3} + 26 (\tau + \theta) + 3 (\tau + \theta)^2 \right)$$

A.4 Space Reduction Floating Point Operation Counts

If l is the number of rows in Z , the number of floating point operations in the reduction is:

$$-2 l^2 L (\tau + \theta) - 2 l L^2 (\tau + \theta)^2 + \frac{16 L^3 (\tau + \theta)^3}{3}$$

B Proofs

Theorem 1 If \mathcal{H} is full rank, then there exists a sequence of elementary row operations that

Floating Point Operations	Auxiliary Routine
Matrix Multiplication	
$2 m n p$	DGEMM(BLAS))
LU Decomposition	
$\frac{2 n^3}{3}$	DGETRF(LAPACK))
LU Decomposition Back-solution	
n^2	DGETRS(LAPACK))
QR Decomposition(with Pivoting)	
$\frac{4 r (3 n^2 - 3 n r + r^2)}{3}$	DGEQPF(LAPACK)
QR Decomposition(rank 1 update)	
$26 n^2$	DLARTG(LAPACK)
Compute Dominant Invariant Subspace	
$\frac{14 n^3}{3} + 12 k n (-k + n) - \frac{k (2 k^2 - 6 k n - 3 n^2) f(\frac{\lambda_k}{\lambda_{-1+k}})}{3}$	DRRIT

Table 2: Floating Point Operation Counts

Operation	Floating Point Operations
Initial QR Decomposition ^a	$\frac{4 r (3 L^2 - 3 L r + r^2)}{3}$
QR Rank One Update	$26 L^2$
Multiplication to Annihilate Rows	$L^2 (\tau + \theta)$
Compute Γ	$2 L^2 (\tau + \theta)$
Max Total	$L^3 \left(\frac{4}{3} + 26 (\tau + \theta) + 3 (\tau + \theta)^2 \right)$

Table 3: Floating Point Operations to Compute Autoregressive Representation

^a r matrix rank

Operation	Floating Point Operations
Hessenberg Reduction Step	$\frac{14 n^3}{3}$
Eigenvalue Swaps	$12 k n (-k + n)$
Orthogonal Iteration Step	$\frac{-2 k^2 (k - 3 n)}{3} + n^2 \min(k, n)$
Total	$\frac{14 L^3 (\tau + \theta)^3}{3} +$ $L^2 (12 k (\tau + \theta)^2 - 14 l (\tau + \theta)^2) +$ $L (-12 k^2 (\tau + \theta) - 24 k l (\tau + \theta) + 14 l^2 (\tau + \theta)) +$ $12 k^2 l + 12 k l^2 - \frac{14 l^3}{3} +$ $\left(\frac{-2 k^3}{3} - 2 k^2 l + k l^2 + k L^2 (\tau + \theta)^2 + L (2 k^2 (\tau + \theta) - 2 k l (\tau + \theta)) \right) f\left(\frac{\lambda_k}{\lambda_{-1+k}}\right)$

Table 4: Invariant Space Calculations Floating Point Operation Count

Operation	Floating Point Operations
Z QR Decomposition	$\frac{4l(l^2 - 3lL(\tau + \theta) + 3L^2(\tau + \theta)^2)}{3}$
\bar{Z} QR Decomposition	$\frac{4(-l+L(\tau + \theta))(3L^2(\tau + \theta)^2 - 3L(\tau + \theta)(-l+L(\tau + \theta)) + (-l+L(\tau + \theta))^2)}{3}$
Matrix Multiplications	$2L^2(\tau + \theta)^2(-l+L(\tau + \theta)) + 2L(\tau + \theta)(-l+L(\tau + \theta))^2$
Total	$-2l^2L(\tau + \theta) - 2lL^2(\tau + \theta)^2 + \frac{16L^3(\tau + \theta)^3}{3}$

Table 5: State Space Reduction Floating Point Operation Count

transforms \mathcal{H} into \mathcal{H}^*

$$\mathcal{H}^* = S\mathcal{H} = \begin{bmatrix} Z_{-\tau}^{\sharp*} & \cdots & Z_{\theta-1}^{\sharp*} & & & \\ H_{-\tau}^{\sharp*} & \cdots & & H_{\theta}^{\sharp*} & & \\ & \ddots & & & & \\ & & H_{-\tau}^{\sharp*} & \cdots & & H_{\theta}^{\sharp*} \\ & & [I \ 0] H_{-\tau}^{\sharp*} & \cdots & \cdots & [I \ 0] H_{\theta}^{\sharp*} \end{bmatrix}$$

with non-singular $H_{\theta}^{\sharp*}$.

Proof Under construction. ■