Solving for the Global Nonlinear Saddle Path: Reverse Shooting vs. Approximation Methods

Manoj Atolia and Edward F. Buffie

August 2004

Intertemporal models grounded in optimizing behavior typically give rise to saddle point solutions. Saddle point equilibria have many desirable properties but are difficult to compute numerically. Some type of forward-shooting algorithm is usually employed to locate the path that lies on the stable manifold. Forward shooting suffers, however, from the drawback that small errors in the guesses for the initial values of the control variables grow exponentially over time as the path veers off in the direction of the unstable manifold. Consequently, even when the initial guess is quite good, the terminal state ends up far away from the true steady state. The instability problem is especially severe in models with multiple jump variables and a long time horizon, leading Judd (1999) to remark that "Shooting methods have only limited value in solving infinite-horizon optimal control problems . . . . To deal with these problems we must develop a better approach" (p.355).

The better approach recommended by Judd (1999) and Brunner and Strulik (2002) is to solve for the equilibrium path through reverse shooting or backward integration.[1] Reverse shooting exploits the fact that the stable manifold in the original system corresponds to the unstable manifold when the system is solved with time reversed. Since the unstable manifold is a strong attractor, reverse shooting is not plagued by extreme sensitivity of the terminal state to errors in the initial guess for the true solution. On the contrary, errors decrease exponentially on paths attracted to the unstable manifold. In the mathematics literature, this property is called "exponential tracking" (Casteneda and Rosa, 1996).

We have two objectives in this paper. The first is to explain exactly *how* to do reverse shooting in systems involving two state variables. Practitioners facing this task will not find the existing literature to be of much help. Judd and Brunner and Strulik work out examples for the case of one state variable but sidestep the difficulties posed by multiple state variables. The section on multidimensional reverse shooting in Judd's book does not contain an example and the solution procedure suggested there is distinctly problematic (more on this later). Brunner and Strulik set up an example with two state variables but then employ a change of variables to transform it into a problem with just one state variable. This paper goes beyond the treatment in Judd and Brunner and Strulik by describing in detail how to use reverse shooting to trap the global saddle path in systems with two state variables. Our algorithm is compact and efficient: when implemented on *Mathematica*, it occupies less than a page and returns solutions in 1-2 minutes in most cases. Moreover, the algorithm handles stiff systems with comparative ease. Even when the ratio of the two negative eigenvalues is on the order of twenty, the program solves for the saddle path in 5-10 minutes.

Our solution procedure exploits the geometry of the state space. Unfortunately, this makes it difficult to generalize the procedure to systems having three or more state variables. In practice therefore it will often be necessary to employ a method that approximates the true saddle path. The most common method, of course, is linearization of the model around the steady state. Recently, however, Novales et al. (1999) have proposed a more sophisticated method that recovers most of the nonlinear structure of the original model. Our second objective is to gain some insight into the conditions under which these approximation methods perform well or poorly. For what types of shocks and nonlinear model structures is approximation error acceptably small? Is it possible to rank alternative approximation methods?

We tackle the issue by comparing the solutions produced by linearization and two variants of the Novales et al. (NEA) procedure with the true solution in two of the workhorses of modern macroeconomics, the closed-economy Sidrauski model and an open economy model featuring currency substitution. The results are informative but also disconcerting. All of the approximation methods closely track the true solution in the standard Sidrauski model, but none scores well in the open economy model or in the Sidrauski model where capital accumulation incurs adjustment costs: in these two models, approximation error is significant for some system variables and large enough to be misleading for important endogenous variables such as the inflation rate and the real interest rate. Contrary to our prior expectations, the NEA method does not consistently outperform linearization. This may seem counterintuitive given that the NEA method incorporates much more of the nonlinear structure of the model than linearization. In a second-best setting that tolerates some approximation error, however, there is no guarantee that capturing more nonlinearities yields more accurate solutions. Our results suggest some guidelines, but, in the end, the analyst has to make a judgement call. In the murky world of the second best, the ranking of approximation methods depends not only on the nonlinear characteristics of the model but also on various factors related to the objectives of the analysis, most notably the time horizon of interest and whether it is important to accurately track the paths of all variables or just a particular subset of variables.

The rest of the paper is organized into five sections. Section 1 is devoted to a full description of our reverse shooting algorithm for the two state variable case. It is essentially a primer on how to do reverse shooting. Sections 2 and 3 use the algorithm to obtain the true solutions for the alternative formulations of the Sidrauski model and to quantify the degree of approximation error in the solutions generated by linearization and the NEA procedures. Section 4 does the same for the open economy currency substitution model. The final section contains concluding remarks.

1. The Reverse Shooting Algorithm

Dynamic general equilibrium models usually produce a system of the type

$$\begin{aligned} \dot{\mathbf{z}} &= f(\mathbf{z}, \mathbf{x}, \mathbf{w}), \\ \dot{\mathbf{x}} &= g(\mathbf{z}, \mathbf{x}, \mathbf{w}), \\ h(\mathbf{z}, \mathbf{x}, \mathbf{w}) &= 0, \end{aligned} \tag{*}$$

where $\mathbf{x}$ is a vector of control variables; $\mathbf{z}$ is a vector of state variables; and $\mathbf{w}$ is a vector containing other control variables and variables that are parametric to the agent but endogenous in general equilibrium.

Suppose the system in (*) has two state variables. We can then draw a ring with radius ϵ around the new stationary equilibrium in state space. This is illustrated in Figure 1, where E and F denote the initial and new steady state.

Projections of saddlepath trajectories onto the state space intersect the ring around F. The true solution is the trajectory that intersects the ring starting from $z_1(0) = z_{1,o}$ and $z_2(0) = z_{2,o}$. To locate this trajectory, set the initial values of the control variables at their final steady-state values and choose n consecutive points p_i, $i = 1, 2, ...n$, on the ring as initial conditions; then compute the trajectory for each point p_i by solving the system in (*) with time reversed.[2] Suppose the trajectory with initial conditions corresponding to p_j

comes closest to the initial steady state. Since trajectories do not cross in state space, the desired trajectory must intersect the ring somewhere on the arc $p_{j-1}p_jp_{j+1}$. This bisection method can be repeated to obtain successively better approximations to the point where the desired trajectory intersects the ϵ-ring.[3] For example, if q and r are the midpoints of the arcs $p_{j-1}p_j$ and p_jp_{j+1}, then one of the three trajectories starting at q, p_j and r will come nearest to the initial steady state. This narrows the search to arc $p_{j-1}p_j$, arc qr, or arc p_jp_{j+1}. The updating process is continued until the distance of the trajectory currently closest to the steady state is within the desired tolerance.

For each trajectory it is necessary to calculate the minimum distance from the initial steady state. This is easily done in our *Mathematica* program. The nonlinear differential equation solver in *Mathematica* returns its solution as an interpolating function. For numerical computations, the interpolating functions may be treated as analytic functions; hence a standard minimization routine can be used to calculate the minimum distance from E.[4]

Judd (1999) has sketched an alternative algorithm that combines reverse shooting with the solution strategy of forward shooting. Choose a time horizon T, set $\mathbf{z} = \mathbf{z}^*$ (* signifies the new steady-state solution), and make an initial guess for $\mathbf{x}(\mathrm{T})$ consistent with $\mathbf{z}(0) = \mathbf{z}_o$. Solve the time-reversed system for $\mathbf{z}(0)$. After computing $\mathbf{z}(0)$ for multiple guesses of $\mathbf{x}(\mathrm{T})$, construct the approximating functions $\mathbf{z}(0) = \mathrm{F}[\mathbf{x}(\mathrm{T})]$. Solve $\mathrm{F}[\mathbf{x}(\mathrm{T})]$ - $\mathbf{z}_o = 0$ for $\mathbf{x}(\mathrm{T})$ and call the solution $\tilde{\mathbf{x}}(\mathrm{T})$. If the solution from the time-reversed system with $\mathbf{x}(\mathrm{T}) = \tilde{\mathbf{x}}(\mathrm{T})$ does not yield $\mathbf{z}(0)$ sufficiently close to $\mathbf{z}_o$, then make a new set of guesses for $\mathbf{x}(\mathrm{T})$ and construct a better set of approximating functions $\mathrm{F}[\mathbf{x}(\mathrm{T})]$. When $\mathrm{F}[\mathbf{x}(\mathrm{T})]$ - $\mathbf{z}_o$ produces a solution $\bar{\mathbf{x}}(\mathrm{T})$ that satisfies the tolerance criterion, check that $\bar{\mathbf{x}}(\mathrm{T})$ is sufficiently close to $\mathbf{x}^*$. If it is not, then increase T and repeat the entire procedure.

This algorithm may or may not work.[5] Since the right value of T and the right approximating functions $\mathrm{F}[\mathbf{x}(\mathrm{T})]$ are found through trial and error, a lot depends on the complexity of the model and the skill the user brings to the process. Even when the user is endowed with lots of human capital, the search may prove frustrating and time-consuming. And the laborious trial-and-error search process has to be repeated whenever new values are assigned to parameters or the model is altered slightly. By contrast, our algorithm is systematic, transparent, and completely reliable. Completely reliable means that the algorithm is *guaranteed* *to work*. Moreover, it works in the same way every time. The user does not need to adjust the algorithm when making runs for different parameter values or different models.

1.1 Refining the Search Process

There are a couple of ways to refine the search process that hones in on the intersection point of the desired trajectory with the ϵ-ring. The first refinement eliminates computation of irrelevant trajectories. Note in Figure 1 that the initial steady state is inside the corridor outlined by the trajectories associated with points p_j and p_{j-1}. Thus, after p_j emerges from the first round of computations as the trajectory that comes closest to E, the next round of search should be confined to the arc $p_{j-1}qp_j$ — there is no need to investigate trajectories that pass through arc p_jp_{j+1}. A smart program uses this information to cut the solution time in half.[6]

The other refinement requires information from the linearized system. In a system with two state variables, the saddle point solution for the linearized system is controlled by the system's two negative eigenvalues. The projection of the eigenvector associated with the largest negative eigenvalue (i.e., the smallest in absolute value) is called the dominant eigenvector ray.[7] If the eigenvalues are real, the relative weight of the term involving the smallest eigenvalue decreases exponentially and the various trajectories on the stable manifold gravitate toward the dominant eigenvector ray (DER). Thus the trajectories that span the state space cluster around points K and J on the ϵ-ring in Figure 2. The linearized solution indicates the direction of approach when the trajectory hits the ϵ-ring. If the approach is southwest to northeast, then the search can be confined to the quarter-circle arc NKR.

1.2 Practical Tips and Computational Issues

In systems with real eigenvalues, all trajectories on the stable manifold converge to the dominant eigenvector ray in the neighborhood of the new steady state. Consequently, many trajectories that start far away from the initial steady state cross the ϵ-ring *very* close to the point where the saddle path intersects the ring. Trajectories that are easy to distinguish early in the adjustment process often look much the same therefore by the time they hit the ϵ-ring. Although reverse shooting ensures convergence to the stable manifold of the original system — the unstable manifold in the time-reversed system — we are still dealing with an ill-conditioned problem.

What this means in practical terms is that often it will be necessary to slice the ϵ-ring very finely. Exactly how fine depends on the value of ϵ and the tolerance criterion that determines how close the acceptable trajectory has to get to the initial steady state. In the simulations undertaken for this paper, ϵ was 10^{-4} times the distance between the two steady states and the tolerance criterion did not accept a trajectory unless it came within .0001% of the initial steady state.[8] To achieve this degree of accuracy, we had to split the ϵ-ring into $10^8 - 10^{30}$ parts.[9] Because the ring had to be partitioned so finely and because numerical error accumulates in the differential equation solver, it was necessary to compute the trajectories with high precision: for some runs, 24-digit precision was sufficient; but stiff systems, where the ratio of the eigenvalues was eight or more, required 32-40 digit precision. (A few cases needed 40+ digits.) Since computation time increases exponentially with the precision of the calculations, the ϵ-ring should not be any smaller than is strictly necessary.[10] A smaller ring implies smaller errors in the solutions for the jump variables in the vicinity of the new steady state, but we have found in repeated applications that reducing ϵ below 10^{-4} does not yield any change in the solution (up to the sixth decimal place).

Computational complexity is also affected by the presence of endogenous variables in the differential equations for the core dynamic system. There are two ways to handle these variables. Going back to the system in (*), one option is to differentiate $h(\mathbf{z}, \mathbf{x}, \mathbf{w}) = 0$ with respect to time and then solve the expanded system with $\mathbf{w} + \mathbf{z}$ control variables. This method method introduces additional numerical error, however, when the $\dot{\mathbf{w}}$ equations are integrated backward to solve the system. The other option is to employ a nonlinear equation solver to compute $\mathbf{w} = n(\mathbf{z}, \mathbf{x})$ at each step when solving the core dynamic system. (*Mathematica 5.0* can compute mixed systems of algebraic and differential equations.[11]) The advantage of this method is that numerical error [associated with solving $\mathbf{w} = n(\mathbf{z}, \mathbf{x})$] does not accumulate in the differential equation solver. Its drawback is that computation time increases when the solution to a system of nonlinear equations is required as an input at *each* step for the solution of the system of differential equations. This is a potentially serious problem when computation entails high precision and a small step size.

Three additional points merit brief comment. First, there is a large gain in computational speed from running the program on the most recent version of *Mathematica/Matlab/Maple*. In the case of *Mathematica*, we can vouch that version 5.0 solves for the saddle path 10-100 times faster than version 4.1. Second, when trying to debug the program, it is helpful at each stage of iteration to plot the closest trajectory and the initial and new steady states in a diagram like Figure 1. The pattern in the plots will usually supply hints about what is wrong in the model or the program. Third, there is a simple way to determine whether rounding error has distorted the solution: re-run the program with a higher value for working precision; if accumulated rounding error is a problem, then the solution will change noticeably.

1.3 Reverse Shooting Made Easy

To promote use of reverse shooting, we have placed two programs in the public domain at http://mailer.fsu.edu/~matolia/. Both programs are written in the spirit of "plug and play." The user sets the values for a few global parameters, plugs in his/her model, and then waits for a solution. It is not necessary to become conversant with the intricacies of the programs.

1.3.1 Slow But Very Easy

The program *Reverse Shoot Easy* does not require information from the user about the nature of the transition path in the neighborhood of the new steady state. Consequently, it is easy to run but slow to find a solution. Below we describe the general structure of the program and how to implement it.

The program has two blocks. In the first, the user assigns values to seven global parameters that control the updating algorithm. These are:

- *Tmax*. *Tmax* sets the maximum number of years for which the time-reversed system is solved. This parameter should be given a large value to ensure that [0, *Tmax*] encompasses the time at which the trajectory reaches its minimum distance from the initial steady state (*tmin*). Accordingly, the default value of *Tmax* is 200. The large value for *Tmax* does not reduce computational speed because, as explained in the next bullet point, most (usually all) trajectories are solved for much shorter time horizons.
- *Maxsteps* and *mxstinc*. *Maxsteps* specifies the maximum number of steps the nonlinear differential equation solver is allowed to take when solving the time-reversed system. The program assigns a "small" value to *Maxsteps* in order to enhance computational speed. Since *Tmax* is large, the constraint imposed by *Maxsteps* invariably determines *tsol*, the length of the solution period. If *Maxsteps* is too small, then *tmin* will equal *tsol* when

the true value of *tmin* occurs at a larger value of t (see Figure 3). The program handles this problem by bumping up *Maxsteps* in increments of *mxstinc* until an interior solution emerges (i.e., until $tmin < tsol$).

- ϵ and *tol.* ϵ and *tol* fix the size of the rings around the new steady state and the initial steady state. For most purposes, $\epsilon = 10^{-4}$ and $tol = 10^{-6}$ will be sufficient. Users who desire greater accuracy far out on the transition path — it usually takes decades to hit the ϵ-ring — should assign a smaller value to ϵ.[12]

  Distance is measured by the Euclidean norm. Units should be chosen therefore so that the changes in z_1 and z_2 across steady states are of similar magnitude.

- *WorkingPrecision.* *WorkingPrecision* sets the number of digits *Mathematica* uses in internal computations. The default value is machine precision. At present, this is sixteen digits. If more precision is needed to find the solution, the program prompts the user to increase *WorkingPrecision.* (See the section on trouble shooting.)

- *ntraj.* *ntraj* specifies the number of trajectories (paired with the points $p_1, \ . \ . \ . \ p_j \ . \ . \ . \ p_{ntraj}$) that will be evaluated on the first sweep over the whole ϵ-ring and when searching over successively smaller segments of the ring. The right value for *ntraj* is model-specific. Usually, however, $ntraj = 20$ will do the job. If this does not work, $ntraj = 50$ almost certainly will.

The second block in the program asks the user to enter the model. This is a straightforward business when machine precision computation is accurate enough to find the saddle path: the model can be entered in its natural form as a system of differential equations $\dot{\mathbf{z}} = f(\mathbf{z}, \mathbf{x}, \mathbf{w})$ and $\dot{\mathbf{x}} = g(\mathbf{z}, \mathbf{x}, \mathbf{w})$, with accompanying algebraic equations $h(\mathbf{z}, \mathbf{x}, \mathbf{w}) = 0$ that yield implicit solutions for relevant endogenous variables ($\mathbf{w}$). High-precision computation is less user-friendly in that it requires explicit solutions of the form $\mathbf{w} = n(\mathbf{z}, \mathbf{x})$. If the initial set of algebraic equations $h(\mathbf{z}, \mathbf{x}, \mathbf{w}) = 0$ is analytically intractable, then the user should artificially expand the number of control/jump variables in the system by differentiating one after another of the algebraic equations with respect to time until explicit solutions can be obtained for the remaining endogenous variables. These operations are simple, mechanical, and tedious; hopefully, future versions of *Mathematica* will eliminate the need for them by solving mixed systems of differential and algebraic equations at high precision.

Trouble Shooting

So far we have discovered only two reasons why the program may fail. First, if *Tmax* is too small, the constraint on the length of the solution period prevents the program from finding the time when the trajectory is closest to the initial steady state. When this happens, $tmin = tsol = Tmax$ (see Figure 3). The solution is to increase *Tmax*.

The other, more common problem is that the computations may not be executed with sufficient precision. Consider the situation in Figure 4. The updating algorithm has narrowed the search to the segment a-b on the ϵ-ring. One more bisection will locate the solution — the trajectory that intersects the ring at point c. But point c is inaccessible if a-c is smaller than working precision for internal computations: when the computer tries to assign a number to point c, rounding error throws it back to either point a or point b. The updating algorithm stops updating and mindlessly repeats the solution for the trajectory that passes through a or b. After the program repeats the same solution 50 times, it aborts and sends out the message "Need to increase working precision." The user should take the hint and re-run the program with a higher value for *WorkingPrecision*.[13] We recommend an increase of four digits per round until the problem disappears.

1.3.2 Much Faster But Less Easy

The program *Reverse Shooting Fast* is *much* faster but entails more work as the user must provide the solution from the linearized model. The linearized solution enables the program to ascertain the slope of the dominant eigenvector ray (DER) and the direction of approach at points close to the new steady state. Armed with this information, yyy limits the search for the global saddle path to a narrow segment of the ϵ-ring.

Figure 5 shows why a little information makes a big difference to the efficiency of the search process. The true solution converges to the DER as the path approaches the steady state F. The solution for $\lim_{t\to\infty} x(t) - x^*$ tells us, in addition, whether the direction of approach is from the southwest or northeast. Suppose it is from the southwest. The search should then start on the arc $(\phi - \theta/2, \phi + \theta/2)$, where $\phi \in (\pi, 3\pi/2)$ is the angle formed by the DER and the horizontal axis. The parameter θ controls the angular span of the arc. It is set in the first block of the program and is given the name *spanangle*.

What's critical to recognize is that the search can start on a very small arc because, exceptional cases aside, the saddle path is *very* close to the DER at the point where it intersects the ϵ-ring. A few rounds of manual search can exploit this insight to rapidly reduce the search space to an arc of microscopic dimensions. Playing it safe, start with a "large" value for θ such as $\pi/100$ and evaluate trajectories passing between two points equally spaced on either side of ϕ. If the true solution is the trajectory associated with point c, then the minimum distance from the initial steady state will first fall and then rise as the search proceeds from $\phi - \pi/200$ to $\phi + \pi/200$. This indicates that the search can be conducted on a smaller arc; ergo abort the current execution and reduce θ to $\pi/10,000$ or $\pi/100,000$. If this arc contains point c, the solutions for minimum distance will again exhibit a U-shaped pattern. Continue the manual search until the solutions for minimum distance rise monotonically, signalling that point c lies *outside* the selected arc. The "optimal" value of θ is then the value for the immediately preceding arc, the last arc where the solutions for minimum distance were U-shaped. Searching over this arc, the updating algorithm finds the solution very quickly.

Potential users should know one more thing: the tradeoff between speed and ease of use is quite pronounced. In the numerical simulations undertaken for this paper, *Reverse Shoot Easy* took 10-15 minutes to locate the saddle path. The faster but less-easy-to-use program *Reverse Shoot Fast* accomplished the task in just 20-40 seconds. Of course, the development of faster computers may soon reduce the difference in solution time to a minute or less. But right now *Reverse Shoot Fast* is definitely the program of choice for users who have to compute many solutions.

2. Example #1

The first example is the familiar Sidrauski (1967) model. Let C, m, b, and K denote, respectively, consumption, real money balances, the real stock of bonds, and the capital stock. The private sector is run by a representative agent who chooses consumption and asset holdings to maximize

$$U = \int_0^\infty \left(\frac{C^{1-1/\tau}}{1-1/\tau} + g\frac{m^{1-1/\tau}}{1-1/\tau} \right) e^{-\rho t} dt, \tag{1}$$

subject to the wealth constraint

$$A = m + b + K, \tag{2}$$

and the budget constraint

$$\dot{A} = Y(K, z) + T + rb - C - \delta K - \pi m, \tag{3}$$

where r is the real interest rate; T is real lump-sum transfers received from the government; π is the inflation rate; Y is real output; ρ is the pure time preference rate; g is a constant; τ is the intertemporal elasticity of substitution; and δ is the depreciation rate of the capital stock. The second term in the utility function reflects liquidity services generated by real money balances. In the budget constraint, $Y(K, z)$ is the CES production function

$$Y = z\left[a_o + a_1 K^{(\beta-1)/\beta}\right]^{\beta/(\beta-1)}, \tag{4}$$

where z is a Hicks-neutral productivity parameter; a_o and a_1 are constants; and β is the elasticity of substitution between capital and labor. (The constant labor input is suppressed.)

On an optimal path, consumption satisfies the Euler equation

$$\dot{C} = C\tau(r - \rho), \tag{5}$$

the real interest rate is tied to the marginal product of capital

$$r = Y_K - \delta = z\left[a_o + a_1 K^{(\beta-1)/\beta}\right]^{1/(\beta-1)} a_1 K^{-1/\beta} - \delta, \tag{6}$$

and the marginal rate of substitution between money and consumption equals the nominal interest rate

$$g\left(\frac{m}{C}\right)^{-1/\tau} = r + \pi. \tag{7}$$

The government finances transfer payments to the private sector and interest payments on its debt by printing money. This implies

$$\dot{m} = T + rb - \pi m. \tag{8}$$

Two more equations complete the model. Since the economy is closed, clearing of the goods market requires that consumption and gross investment add up to real output each period. The law of motion for the capital stock is thus

$$\dot{K} = Y(K,z) - \delta K - C. \tag{9}$$

Finally, the shift parameter in the production function evolves according to

$$\dot{z} = w(z^* - z), \quad z(0) = z_o = 1. \tag{10}$$

Starting from a stationary equilibrium, the economy enjoys a sequence of favorable productivity shocks. The parameter w determines how fast the productivity improvements arrive.

2.1 The Linearized Solution

Substituting for r and π in (5) and (8) produces

$$\begin{aligned} \dot{C} &= \tau C[Y_K(K,z) - \rho - \delta], && (5') \\ \dot{m} &= T + Y_K(K,z)b - [g(m/C)^{-1/\tau} - Y_K(K,z)]m. && (8') \end{aligned}$$

Equations (5′), (8′), (9) and (10) form a self-contained system of four differential equations in m, C, K, and z. The system has two state variables (K and z) and two negative eigenvalues. Thus, the steady state is a saddle point. The linear approximation of the saddle path is

$$\begin{bmatrix} C(t) - C^* \\ m(t) - m^* \\ K(t) - K^* \\ z(t) - z^* \end{bmatrix} = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \\ X_{31} & X_{32} \\ X_{41} & X_{42} \end{bmatrix} \begin{bmatrix} h_1 e^{\lambda_1 t} \\ h_2 e^{\lambda_2 t} \end{bmatrix}, \tag{11}$$

where a * denotes the steady-state value of a variable; λ_1 and λ_2 are negative eigenvalues; $(X_{1j}, X_{2j}, X_{3j}, X_{4j})'$ is the eigenvector associated with λ_j (see the appendix for the exact expressions); and h_1 and h_2 are constants determined by the initial conditions $K(0) = K_o$ and $z(0) = z_o = 1$.

The paths for the endogenous variables r, π, and Y are derived by taking first-order Taylor series approximations of (4), (6), and (7) around the new steady state and substituting in the solution paths for K, C, m, and z. This yields

$$r(t) = \rho + Y_{KK}[K(t) - K^*] + Y_{Kz}[z(t) - z^*], \tag{12}$$

$$
\begin{aligned}
Y(t) &= Y^* + (\rho+\delta)[K(t)-K^*] + Y_z[z(t)-z^*], && (13)\\
\pi(t) &= \pi^* - Y_{KK}[K(t)-K^*] - Y_{Kz}[z(t)-z^*] \\
&\quad -\frac{\rho+\pi}{\tau}\left[\frac{m(t)-m^*}{m^*} - \frac{C(t)-C^*}{C^*}\right]. && (14)
\end{aligned}
$$

2.2 The NEA (Novales et al.) Solution

The NEA procedure also computes the solution in (11). The linearized solution is used, however, only to impose a set of stability conditions on the relationship between the path of each jump variable and the paths of the state variables. Solving (11) for $m(t)$ and $C(t)$ as a function of $K(t)$ and $z(t)$ gives

$$
\begin{aligned}
C(t) &= C^* + \begin{bmatrix} X_{11} & X_{12} \end{bmatrix} \begin{bmatrix} X_{31} & X_{32} \\ X_{41} & X_{42} \end{bmatrix}^{-1} \begin{bmatrix} K(t)-K^* \\ z(t)-z^* \end{bmatrix}, && (15)\\
m(t) &= m^* + \begin{bmatrix} X_{21} & X_{22} \end{bmatrix} \begin{bmatrix} X_{31} & X_{32} \\ X_{41} & X_{42} \end{bmatrix}^{-1} \begin{bmatrix} K(t)-K^* \\ z(t)-z^* \end{bmatrix}. && (16)
\end{aligned}
$$

Once the stability conditions are in place, the NEA procedure returns to the original nonlinear model to derive the solution paths for the state variables and all endogenous variables. In the case at hand, the computer solves (4), (6), (7), (15), and (16) simultaneously with the two differential equations in (9) and (10) for the paths of m, C, r, Y, π, K, and z. This is the full NEA procedure. An alternative, simpler procedure generates the solution paths for r, Y, and π by substituting the linearized solutions for the system variables into (4), (6), and (7). We call this the semi-Novales procedure.

While the NEA procedure is attractive in some ways, it is not clear that it is a better approximation method than linearization. For all of its obvious shortcomings, linearization is at least an internally consistent solution procedure. The same cannot be said of the NEA procedure, or of any other procedure that incorporates some but not all of the nonlinear structure of the true model.

2.3 Numerical Solutions

To calibrate the model, we set

$$
\delta = .05, \quad \beta = .50, \quad m_o/Y_o = .075, \quad \tau = .25, .75
$$

$$\theta_o \;=\; .50, \quad \rho = .10, \quad \pi_o = .50.$$

Ordinary values are assigned to the depreciation rate (δ), the elasticity of substitution between capital and labor (β), and the ratio of high-powered money to GDP. Empirical estimates suggest that the intertemporal elasticity of substitution is less than unity and perhaps as small as .1-.2; since τ is a key parameter, it is allowed to vary from .25 to .75. For the time preference rate (ρ), the initial income share of capital (θ_o), and the initial inflation rate, we chose values appropriate for a less developed country experiencing macroeconomic trouble. We prefer this scenario simply because approximation error is of greater concern and easier to assess (in purely visual terms) if inflation falls from 50% to 30% on the transition path instead of from 4% to 2.6%.

Tables 1a-1b and 2a-2b show how the solutions from linearization and the NEA procedures compare with the true solution. The cumulative productivity shock is 5% in Tables 1a-1b and 20% in Tables 2a-2b; in all runs, the parameter w is set so that 80% of the productivity gains materialize within eight years ($w = .2$). The row labeled RS gives the correct reverse-shooting solution for the cumulative change in variable x at time t [i.e., $x(t) - x_o$]. The entries in the other two rows state the absolute approximation error for the linearized and NEA solutions [i.e., $x_a(t) - x_c(t)$, where x_c is the correct solution and x_a is the solution produced by the approximation method].

Although the economics of the Sidrauski model is not our main concern, it will prove helpful to first note a few features of the true solution. As expected, consumption, real money balances, the capital stock, and real output rise in the long run while inflation falls. (π falls because an increase in m widens the base for the inflation tax.) Since anticipation of higher future income spurs a large upward jump in consumption at $t = 0$, the capital stock decreases in the first phase of the adjustment process.[14] The initial phase of capital decumulation is deeper and more prolonged when $\tau = .25$ and the incentive to smooth the path of consumption is stronger.

2.3.1 Small Productivity Shocks

The different approximation methods do a fairly good job in the runs where the productivity shock is small. Inspection of the linearized solution reveals some nontrivial approximation error in the path for real output in the short run (t = 0-1), and in the paths for real money balances and inflation when $\tau = .75$. The approximation error for real money balances is about the same in the FN and SN solutions; a small amount of error also creeps into the FN solution for the capital stock at t = 10, 20. None of the errors, however, is large in absolute terms (see Figures 6a-6c). Overall, therefore, the approximation methods draw a reasonably accurate picture of the true transition path. If asked to rank the methods, we would put linearization behind FN and SN, citing the larger errors in its solution for the path of inflation.

2.3.2 Large Productivity Shocks

All of the approximation methods encounter problems when the cumulative productivity shock is 20%. Not only does approximation error affect more variables, it is also large enough in some cases to be a source of incorrect qualitative answers. Examine first the solutions for m and π in the run where $\tau = .75$. The approximation methods say that the price level rises .7% at $t = 0$ and that real money balances decrease slightly from 7.5% to 7.45% of GDP; in point of fact, the price level falls 8% on impact and real money balances jump immediately to 8.1% of GDP. The errors diminish steadily thereafter but are still significant at $t = 5$. And since the solution for real money balances is badly wrong in the first five years, so also is the solution for the inflation rate. Even though inflation decreases continuously, the FN and SN methods claim that inflation rises in the first year and stays above its pre-shock level for three years. Linearization errs in the opposite direction, predicting decreases in the inflation rate 7-10 percentage points larger than the actual decrease. For other variables, the discrepancy between the approximate and true solutions is less disturbing. At t = 0-1, however, the percentage errors in the linearized solution for real output and in the linearized/SN and FN solutions for consumption, are considerably larger than in the small shock case.

Approximation error has a different look but is no less serious a problem in the run for $\tau = .25$. Some solutions improve a great deal. The solutions for the path of real money balances, for example, are much more accurate than in Table 2a. On the other hand, there is more error in the solutions for consumption (after $t = 3$), and the FN method obviously has trouble tracking the path of the capital stock at the 10-20 year horizon. Furthermore, despite the much better solutions for m, the solutions for the path of inflation are again highly inaccurate. This is not a fluke: the elasticity of inflation with respect to the ratio of real money balances to consumption is three times larger when τ is .25 instead of .75; comparatively small errors in the solutions for m and C lead therefore to big errors in the solution for inflation.

Rankings

Given the variation in the results, any ranking of the approximation methods is partly subjective. Nevertheless, a strong case can be made for the view that SN is the best approximation method in Tables 2a-2b. None of the methods delivers solutions of acceptable accuracy for the path of inflation in either run, or for the path of real money balances in the run where $\tau = .75$; unlike its competitors, however, SN does not produce bad errors in the solution path for any other variable. Moreover, while the method that yields the smallest error differs across time periods and variables, the SN solution is never far away from the best solution. Maybe the FN solution is slightly better in Table 2a, but SN comes out on top when approximation error is evaluated for both runs. It is certainly the most robust approximation method.

3. Example #2

Our second example differs from the first in only one detail: firms now incur adjustment costs $\psi(\dot{K}/K)$ when changing the capital stock. Following standard practice in the literature, adjustment costs are assumed to be increasing, symmetric, and strictly convex: $\psi(0) = 0$, $\psi' \gtreqless 0$ as $\dot{K} \gtreqless 0$, and $\psi'' > 0$. In the numerical results to be presented shortly, $\psi = n(I/K - \delta)^2/2$, where I is gross investment and the parameter n is set so that the q-elasticity of investment spending equals two at the initial steady state.[15]

Adjustment costs alter private behavior and the way markets interact. Two changes are important. First, the real interest rate fluctuates much more because it is no longer tied to the marginal product of capital. Equation (6) is replaced by the more complicated solution (see the appendix)[16]

$$\begin{aligned} r &= \rho + \frac{n}{K^2\Delta\tau}\left[\{Y_K + n(I/K-\delta)I/K^2 - [1+n(I/K-\delta)](2I/K-\delta)\}(I-\delta K) + w(z^*-z)\right] \\ &\quad + \frac{1+n(I/K-\delta)/K}{\Delta\tau}\left\{Y_K + n(I/K-\delta)I/K^2 - [1+n(I/K-\delta)/K](\rho+\delta)\right\}, \end{aligned} \tag{17}$$

where

$$\Delta \equiv \frac{Cn}{K^2} + \frac{[1+n(I/K-\delta)/K]^2}{\tau}.$$

The second important change concerns the impact of the productivity shocks on the path of real money balances. Recall that anticipation of higher income in the future provokes a sharp upward jump in consumption at $t = 0$. Since higher consumption increases the need for liquidity services, the private agent also wants to hold more real money balances. In the model without adjustment costs, the desire for additional liquidity causes investment to decrease more, *ex ante*, than consumption rises. When the price level drops to clear the goods market, the real money supply increases.

Temporary capital decumulation is more expensive in the presence of adjustment costs. The private agent has an incentive therefore to finance higher consumption spending in the short run partly by cutting investment and partly by reducing real money balances; using both assets to smooth the path of consumption saves on adjustment costs. It turns out that when the q-elasticity of investment spending equals two the saving-of-adjustment-costs effect dominates the desire-for-more-liquidity effect. Because investment decreases less, *ex ante*, than consumption rises, the price level jumps upward and real money balances fall at $t = 0$.

The point we wish to make with this example is that seemingly small changes in a model may cause dramatic changes in the pattern and magnitude of approximation error. Table 3 shows the results for the paths of m, r, and π in the case where the productivity shock is large.[17] What immediately catches the eye are the huge errors in the short-run solutions for m and π. The damage to the solution for the run $\tau = .25$ is especially severe. In the period t = 0-5, approximation error in the solution for m is 1.5-5 times larger than in the model without adjustment costs. Since the path for π is highly sensitive to variations in the path of m when τ is small, approximation error in the solution for inflation is absurdly large for the first three years; even at $t = 7$, the approximate solutions miss the true solution by 4-4.5 percentage points.

4. Example #3

In the last example, the economy is small and open, the exchange rate floats, and foreign currency F competes with domestic currency in the provision of liquidity services. World prices are constant and equal to unity, so the price level is one and the same as the exchange rate. The private agent solves the problem

$$\underset{\{C,m,b,F\}}{Max} \quad \int_0^\infty \left[\frac{C^{1-1/\tau}}{1-1/\tau} + \phi(m,F)\right] e^{-\rho t} dt, \tag{18}$$

subject to

$$\begin{aligned} A &= m + b + F, & (19) \\ \dot{A} &= Y(z) + T + rb - C - \pi m. & (20) \end{aligned}$$

To keep the number of state variables at two, we have dropped capital from the model. Liquidity services enter the utility function in the form

$$\phi(m,F) = g_o \frac{\left[g_1 m^{1-1/\sigma} + (1-g_1)F^{1-1/\sigma}\right]^{(\tau-1)\sigma/\tau(\sigma-1)}}{1-1/\tau},$$

where g_o and g_1 are constants and σ is the elasticity of substitution between domestic and foreign currency.

The necessary conditions for an optimum require that the path of consumption conform to the Euler equation (5) and that the marginal rate of substitution between consumption and m or F equal the opportunity cost of holding that type of currency:

$$\begin{aligned} \frac{\phi_m}{C^{1/\tau}} &= r + \pi, & (21) \\ \frac{\phi_F}{C^{1/\tau}} &= r. & (22) \end{aligned}$$

The government budget constraint (8) is unchanged but equation (9) disappears. Saving is now associated with a current account surplus and accumulation of foreign assets:

$$\dot{F} = Y(z) - C. \tag{23}$$

Equations (5), (8), (10), and (23) define the dynamic system, along with solutions from (21) and (22) that link the paths of r and π to the paths of m, C, F, and z. The system is similar to the systems analyzed in sections 2 and 3: F takes the place of K as one of the two state variables; as before, m and C are jump variables.

4.1 Numerical Results

We calibrate the model to a highly dollarized economy in which the private sector holds twice as much foreign currency as domestic currency. The initial values of m, C, Y, z, r, and π are the same as in the Sidrauski model, while the currency substitution parameter takes either the low value .5 or the high value 2. In keeping with the empirical evidence, all runs assume $\sigma > \tau$. This implies that domestic and foreign currency are Edgeworth substitutes ($\phi_{mF} < 0$).

The bottom line for small shocks is essentially the same as in the Sidrauski model. The FN and SN solutions are a little better than the linearized solution, but all of the approximation methods provide satisfactory results.

In the case of large shocks, the bottom line is *not* the same. Some of the patterns seen in the Sidrauski model recur. There are also, however, some important differences. Three results stand out in Tables 4a-4c:

- *The degree of approximation error is highly sensitive to the parameterization of the model.* There is tremendous variation in the accuracy of the results depending on the value assigned to τ. Approximation error is a serious problem in the two runs for $\tau = .25$, but not in the run where $\tau = .75$. In the Sidrauski model, by contrast, approximation error was substantial in all of the runs for large productivity shocks.
- *None of the approximation methods tracks the path of the real money supply or the path of inflation successfully in the runs for $\tau = .25$.* Approximation error in the solution for π is too large in both runs but much greater in the run where $\sigma = .50$. The results here parallel those in the runs for $\tau = .25$ vs. $\tau = .75$ in the Sidrauski model with adjustment costs. The explanation is also parallel: approximation error is much greater when $\sigma = .50$ vs. $\sigma = 2$ because small values for τ *and* σ make the solution for π sensitive to errors in the solution for m.

- *Approximation error is significant in the solutions for the real interest rate and the stock of foreign currency.* In both the Sidrauski model and the open economy model, approximation error in the solution for the endogenous state variable is large relative to the change in the variable. Foreign currency holdings, however, are much smaller than the capital stock (15% vs. 333% of GDP). Significant approximation error in tracking the *change* in F translates therefore into significant error in tracking the *level* of F. The large errors in the solution for F distort, in turn, the FN and SN solutions for the path of r. The linearized solution is fairly accurate after t = 1-2 because errors in the solution for F partly offset errors introduced by ignoring nonlinearities in the asset demand equations — another case where it is better to be lucky than smart.

5. Concluding Remarks

This paper has demonstrated how reverse shooting can be used to pick out the global saddle path in dynamic systems with two state variables. We applied our algorithm to two models, the closed economy Sidrauski model and an open economy currency substitution model, and compared the true solution for the saddle path with the solutions generated by several popular approximation methods. Numerical runs were carried out for sequences of small and large productivity shocks and for alternative parameterizations of the models.

The comparison of approximation methods yielded mixed results. Both linearization and the Novales et al. procedure provided accurate solutions in the case of small productivity shocks. For large shocks, the results were model-specific. The solutions for some variables were highly inaccurate in the Sidrauski model, but in the currency substitution model this was true only in the case where the intertemporal elasticity of substitution was small. It was not possible to rank the competing approximation methods. The best method depends on the model under investigation and the variables the analyst is most interested in tracking. While this is not a particularly informative conclusion, future research that investigates different shocks (e.g., one-off shocks) and different model structures (especially models where the stable manifold is highly nonlinear) may uncover robust rules.

Forward-shooting algorithms and the reverse-shooting algorithm proposed in Judd (1998) are alternative methods of finding the global saddle path. These algorithms are not, however, as transparent and reliable as the reverse-shooting algorithm we have developed. Our algorithm exploits the insight that in systems with two state variables the search can be directed by a simple updating rule that guarantees convergence to the correct solution. After starting the program, the user does not have to engage in any intermediate tinkering to help it find the solution. The programs we have placed in the public domain are very user friendly; with “barriers to entry” all but eliminated, use of reverse shooting should become widespread.

The strength of our algorithm is also its weakness. When the state space is two dimensional, it is easy to find a rule that steers the search progressively closer to the solution. It is far more difficult to find a rule to direct the search in systems with three or more state variables. This is the major priority for future research.

Appendix

I. The Linearized Solution in Example #1

Linearizing (5′), (8′), (9) and (10) around the new steady state (m^*, C^*, K^*, z^*) yields

$$\begin{bmatrix} \dot{C} \\ \dot{m} \\ \dot{K} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \tau C Y_{KK} & \tau C Y_{Kz} \\ -\pi m / C\eta & \pi(1-\eta)/\eta & (b+m)Y_{KK} & (b+m)Y_{Kz} \\ -1 & 0 & \rho & Y_z \\ 0 & 0 & 0 & -w \end{bmatrix} \begin{bmatrix} C - C^* \\ m - m^* \\ K - K^* \\ z - z^* \end{bmatrix}, \tag{A1}$$

where $\eta \equiv \pi\tau/(\rho + \pi)$ is the elasticity of money demand with respect to inflation (defined to be positive).

Under the assumption that the economy operates on the upward-sloping portion of the seigniorage Laffer curve (where $\eta < 1$), the system in (A1) has two negative eigenvalues: $\lambda_1 = -w$ and $\lambda_2 = (\rho - \sqrt{\rho^2 - 4\tau C Y_{KK}})/2$. The associated eigenvectors are obtained from

$$\begin{bmatrix} 0 & 0 & \tau C Y_{KK} & \tau C Y_{Kz} \\ -\pi m / C\eta & \pi(1-\eta)/\eta & (b+m)Y_{KK} & (b+m)Y_{Kz} \\ -1 & 0 & \rho & Y_z \\ 0 & 0 & 0 & -w \end{bmatrix} \begin{bmatrix} X_{1j} \\ X_{2j} \\ X_{3j} \\ X_{4j} \end{bmatrix} = \begin{bmatrix} \lambda_j X_{1j} \\ \lambda_j X_{2j} \\ \lambda_j X_{3j} \\ \lambda_j X_{4j} \end{bmatrix}, \quad j = 1, 2. \tag{A2}$$

Straightforward algebra produces

$$\begin{aligned} X_{11} &= \tau C[Y_{KK}Y_z - (\rho + w)Y_{Kz}]/J, \\ X_{21} &= \frac{(\pi m/C)X_{11} - \eta(b+m)(Y_{KK}X_{31} + Y_{Kz})}{\pi(1-\eta) + w\eta}, \\ X_{31} &= -(wY_z + \tau C Y_{Kz})/J, \qquad X_{41} = 1, \end{aligned}$$

where

$$J \equiv w(\rho + w) + \tau C Y_{KK},$$

and

$$\begin{aligned} X_{12} &= \rho - \lambda_2, \\ X_{22} &= \frac{(b+m)\eta Y_{KK} - (\pi m/C)(\rho - \lambda_2)}{\eta\lambda_2 - \pi(1-\eta)}, \\ X_{32} &= 1, \qquad X_{42} = 0. \end{aligned}$$

II. The Sidrauski Model With Adjustment Costs

In the model with adjustment costs, the private agent solves the problem

$$U = \int_0^\infty \left(\frac{C^{1-1/\tau}}{1-1/\tau} + g\frac{m^{1-1/\tau}}{1-1/\tau} \right) e^{-\rho t} dt, \tag{A3}$$

subject to

$$\begin{aligned} A &= m + b, & \text{(A4)}\\ \dot{A} &= Y(K,z) + T + rb - C - I - n\frac{(I/K-\delta)^2}{2} - \pi m, & \text{(A5)}\\ \dot{K} &= I - \delta K. & \text{(A6)} \end{aligned}$$

The wealth constraint includes only m and b because there is no equities market in which money can be swapped for claims to the capital stock. (Recall that the model is formulated with a less developed country in mind.)

The necessary conditions for an optimum consist of

$$\begin{aligned} C^{-1/\tau} &= \beta_1, & \text{(A7)}\\ g\left(\frac{m}{C}\right)^{-1/\tau} &= r + \pi, & \text{(A8)}\\ 1 + n(I/K-\delta)/K &= \beta_2/\beta_1, & \text{(A9)}\\ \dot{\beta}_1/\beta_1 &= \rho - r, & \text{(A10)}\\ \dot{\beta}_2/\beta_2 &= \rho + \delta - \frac{\beta_1}{\beta_2}\left[Y_K + n(I/K-\delta)\frac{I}{K^2} \right], & \text{(A11)} \end{aligned}$$

where β_1 and β_2 are the multipliers attached to the constraints (A5) and (A6).

Equations (4), (8) and (10) carry over from the model without adjustment costs. There is a slight change, however, in the market-clearing condition, which now reads

$$C + I + n\frac{(I/K-\delta)^2}{2} = Y(K,z). \tag{A12}$$

The solution procedure is basically the same in the models with and without adjustment costs. The main difference is that the solution for the real interest rate is much more complicated in the presence of adjustment costs. To solve for r, note first from (A7) and

(A10) that

$$r = \rho + \dot{C}/C\tau. \tag{A13}$$

From (A11) and (A12),

$$\begin{aligned} C(\dot{C}/C) + [1 + n(I/K - \delta)/K]\dot{I} &= s_1, & \text{(A14)} \\ -\frac{1 + n(I/K - \delta)/K}{\tau}\frac{\dot{C}}{C} + \frac{n}{K^2}\dot{I} &= s_2, & \text{(A15)} \end{aligned}$$

where

$$\begin{aligned} s_1 &\equiv [Y_K + n(I/K - \delta)I/K^2](I - \delta K) + Y_z w(z^* - z), \\ s_2 &\equiv [1 + n(I/K - \delta)/K](\rho + \delta) + \frac{n}{K^2}(2I/K - \delta)(I - \delta K) - Y_K - n(I/K - \delta)\frac{I}{K^2}. \end{aligned}$$

These two equations can be solved for $\dot{C}/C$ and $\dot{I}$. The soluton for $\dot{C}/C$ is

$$\begin{aligned} \frac{\dot{C}}{C} &= \frac{n}{K^2\Delta}\left[\{Y_K + n(I/K - \delta)I/K^2 - [1 + n(I/K - \delta)](2I/K - \delta)\}(I - \delta K) + w(z^* - z)\right] \\ &\quad + \frac{1 + n(I/K - \delta)/K}{\Delta}\left\{Y_K + n(I/K - \delta)I/K^2 - [1 + n(I/K - \delta)/K](\rho + \delta)\right\}, \quad \text{(A16)} \end{aligned}$$

where

$$\Delta \equiv \frac{Cn}{K^2} + \frac{[1 + n(I/K - \delta)/K]^2}{\tau}.$$

Substituting the above solution for $\dot{C}/C$ into (A13) gives equation (17) in the main text.

NOTES

1. Reverse shooting and backward integration are similar methods. The principal difference, as emphasized by Brunner and Strulik, is that the time required for the path to reach a specified distance from the initial steady state is endogenously determined in backward integration.

2. The time reversed system replaces t with -t. This results in negative signs appearing in front of $f(\cdot)$ and $g(\cdot)$ in $(*)$.

3. Brunner and Strulik allude briefly to elements of this solution strategy on p.746 of their paper. They do not, however, discuss the strategy in detail or explain how to deal with the many issues that arise in its practical implementation.

4. In other programs, the solution provided by the nonlinear differential equation solver is a sequence of points $\{\mathbf{z}(t_i), t_i\}_{i=1}^{N}$. Computing the minimum distance of a trajectory from the initial steady state then requires implementation of an interpolation routine or of a routine that generates an approximate analytical representation of $\{\mathbf{z}(t_i), t_i\}_{i=1}^{N}$.

5. Judd is suitably cautious. When comparing the algorithm with the forward-shooting method, he claims only that "This is *more likely* to work, since small changes in λ^T will, *hopefully*, generate only small changes in the implied value of $k(0)$" (p.361, emphasis is ours).

6. Usually, a simple geometric argument tells whether the search should move to the left or the right from p_j.

7. The linearized solutions for the state variables are $z_1(t) - z_1^* = X_{11}h_1e^{\lambda_1 t} + X_{12}h_2e^{\lambda_2 t}$ and $z_2(t) - z_2^* = X_{21}h_1e^{\lambda_1 t} + X_{22}h_2e^{\lambda_2 t}$, where $\lambda_1 < \lambda_2 < 0$ are the system's two negative eigenvalues, X_{ij} is the $\mathrm{i_{th}}$ component of the eigenvector associated with λ_j, and h_j are constants determined by initial conditions. The slope of the dominant eigenvector ray is simply X_{22}/X_{12}.

8. The value given to ϵ assumes that units have been chosen so that the changes in z_1 and z_2 across steady states are similar in magnitude. (See the later discussion in section 1.3.1.)

9. This refers to how close comparison trajectories have to be to locate a solution with the desired accuracy, not to the number of trajectories computed. When information about the dominant eigenvector ray is not used to guide the search, the maximum number of trajectories that has to be computed is $2Log_2 10^8 - 2Log_2 10^{30} = 53 - 199$.

10. Loosening the tolerance criterion (the size of the ring around the initial steady state) does not reduce computation time significantly.

11. *Mathematica 5.0* solves mixed systems only for machine precision, not at higher levels of precision. Users should also know that the program does not really solve a mixed system; it differentiates the algebraic equations and then solves an expanded system of differential equations.

12. Since it is not *the exact* saddle path, the solution path will eventually move outside the ϵ-ring. The smaller is ϵ, the longer it takes for this to happen.

13. The requisite amount of precision depends on ϵ. An alternative therefore to raising working precision is to increase ϵ. A higher value for ϵ results in larger errors when the solution path (with time running forward) moves outside the ϵ-ring. But this will not matter if the path hits the ring at a large value of t and accurate solutions are not needed far out on the transition path (e.g., where t $>$ 50).

14. This result depends on the intertemporal elasticity of substitution beingh smaller than unity. For $\tau > 1$, the capital stock increases monotonically.

15. The first-order condition associated with investment is $1 + n(I/K - \delta)/K = q$, where q is the ratio of the shadow price of capital to the shadow price of wealth, commonly known as "Tobin's q" (the ratio of the demand price to the supply price of capital). Define $\Omega = (q/I)dI/dq$ to be the q-elasticity of investment spending. Evaluated at a steady state, the first-order condition then implies $n = K/\delta\Omega$.

16. Note that when $n = 0$ equation (17) gives $r = Y_K - \delta$, the solution in the model without adjustment costs.

17. The approximation methods still give accurate results for small shocks. In the large shock case, we present results only for m, r, and π. The magnitude and the pattern of approximation error for C, K, and Y is similar to that in Tables 2a-2c.

References

- Brunner, M. and H. Strulik, 2002, “Solution of perfect foresight saddle point problems: a simple method and applications.” Journal of Economic Dynamics and Control 26, 737-753.

  Casteneda, N. and R. Rosa, 1996, “Optimal estimates for the uncoupling of differential equations.” Journa of Dynamics and Differential Equations 8, 103-139.

  Judd, K., 1999, Numerical Methods in Economics (Cambridge, MA; MIT Press).

  Novales, A., Dominguez, E., Perez, J. and J. Ruiz, 1999, “Solving nonlinear rational expectations models by eigenvalue-eigenvector decompositions.” In R. Marimon and A. Scott, eds., Computational Methods for the Study of Dynamic Economies (New York, Oxford University Press).

  Sidrauski, M., 1967, “Rational choice and patterns of growth in a monetary economy.” American Economic Review 57 (May), 534-544.