SOLVING A SADDLEPATH UNSTABLE MODEL WITH COMPLEX-VALUED EIGENVALUES

by

Peter J. Stemp[*]
Department of Economics
The University of Melbourne
Melbourne, Victoria, 3010
Australia
email: pjstemp@unimelb.edu.au

and

Ric D. Herbert[#]
School of Design, Communication and Information Technology
The University of Newcastle
University Drive, Callaghan, NSW, 2308
Australia
email: ric.herbert@newcastle.edu.au

Paper to be presented to the 9th International Conference
Computing in Economics and Finance
Society of Computational Economics
University of Washington
Seattle, Washington, USA
July 11-13, 2003

[*] Peter Stemp is grateful to the Faculty of Economics and Commerce at The University of Melbourne for financial support through a Faculty Research Grant.
[#] Ric Herbert is grateful to The University of Newcastle for financial assistance through an Early Career Research Grant.

ABSTRACT

This paper investigates the success of the well-known reverse-shooting and forward-shooting algorithms in finding stable solutions for linear macroeconomic models that both possess the particular property known as saddle-path instability and also have highly cyclic dynamic properties. It is anticipated that assessing how well these algorithms cope with solving highly cyclic models will also provide insights into how well they are likely to cope with solving non-linear models.

In this paper, a perfect foresight version of the well-known Cagan (1956) model of the monetary dynamics of hyperinflation is augmented with a labour market. Additional eigenvalues are then generated through sluggish adjustment mechanisms for wages and for the supply of labour. This process provides the simplest model with stable complex-valued eigenvalues and a saddlepath: a model with two stable complex-valued eigenvalues and one unstable real-valued eigenvalue.

Using this model, it is possible to define an indexing parameter that, when varied, determines a range of values for the stable eigenvalues: from (i) real-valued, to (ii) complex-valued with small absolute imaginary part, to (iii) complex-valued with large absolute imaginary part. This leads to corresponding stable time-paths for the model, which are (i) either humped or monotonic, (ii) cyclic but with infrequent cycles, and (iii) cyclic but with frequent cycles.

This paper then compares the properties of solutions derived using the reverse-shooting and forward-shooting approaches as the magnitude of the indexing parameter (and hence of the cycles) is allowed to vary. In the highly oscillatory case, we show that the success of both approaches is crucially dependent on the choice of ODE solver.

JEL classification: C63; E17

Keywords: Macroeconomics; Complex-valued eigenvalues; Real-valued eigenvalues; Cyclic convergence; Monotonic convergence; Saddle-path instability; Computational techniques.

1. INTRODUCTION

Macroeconomic models play an important role in today's society. They are used for forecasting future values of a multitude of economic variables and for evaluating the effects of changes or proposed changes in government policy. Such models are, in general, both dynamic and non-linear. These characteristics mean that solving such models for the dynamic paths of jointly-determined economic variables, a procedure vital for forecasting and policy evaluation, is not always straightforward. These difficulties have led to the development of a variety of algorithms that are used to solve macroeconomic models with particular characteristics.

In this paper we consider a linear model with complex-valued eigenvalues, and hence with cyclic dynamics. The model is also characterised by the property known as saddle-path instability. By choosing alternative parameter configurations, we allow the frequency of cycles to increase and hope to gain insights into how well alternative solution algorithms are likely to cope with non-linear dynamics. Because the model is linear, we have the advantage that a true analytic solution exists and this true solution can be used as a benchmark to evaluate the success of the competing algorithms.

Judd (1998, Chapter 10) describes a variety of methods for solving systems of equations described by ordinary differential equations. Because of the saddle-path property we focus on solution methods for boundary-value problems. In particular we consider two well-known shooting methods: reverse-shooting and forward-shooting. Both methods require searching over a sub-space or manifold so as to find the "right" solution. However, it can be demonstrated that, in general, the reverse-shooting method involves searching over a smaller dimensional manifold than does the forward-shooting method. This means, that, as long as the reverse-shooting algorithm is successful, then it is likely to be more efficient than the forward-shooting approach. Our focus on a model with complex-valued eigenvalues (and hence with cyclic dynamics) allows us to evaluate how successfully the two competing algorithms are likely to cope with solving highly non-linear models.

The rest of the paper proceeds as follows: Section 2 introduces the basic model. Section 3 of the paper derives the model solution. Section 4 is the main results section of the paper, describing the approach and results for competing algorithms. Section 5 contains some concluding comments.

2. THE MODEL

The basic model

Consider the following model:

$$m - p = \alpha_1 y - \alpha_2 \dot{p} \tag{1}$$

$$y = \beta + (1-\gamma)n, 0 < \gamma < 1 \tag{2}$$

$$w - p = \delta - \gamma n \tag{3}$$

where all variables are functions of time, lower-case letters denote logarithms and

y = output;
n = employment;
p = price level;
m = nominal money stock, assumed to be constant; and
w = wage rate.

With y fixed and m exogenous, equation (1) is the perfect foresight extension of the Cagan (1956) model. Using corresponding upper case letters to denote levels, the other two equations can be derived from a Cobb-Douglas production function of the form:

$$Y = AN^{1-\gamma}, \tag{2'}$$

yielding the following first-order condition for profit maximisation:

$$\frac{W}{P} = \frac{dY}{dN} = (1-\gamma)AN^{-\gamma}. \qquad (3')$$

Equations (2) and (3) can then be derived from (2') and (3') by taking logarithms.

Introducing sluggish adjustment

Next, by introducing sluggish adjustment for wages and labour, we can derive the following model:[1]

$$m - p = \alpha_1 y - \alpha_2 \dot{p} \qquad (4)$$

$$y = \beta + (1-\gamma)n, 0 < \gamma < 1 \qquad (5)$$

$$\dot{n} = \theta(\delta - \gamma n - w + p) \qquad (6)$$

$$\dot{w} = \eta(n - \overline{n}) \qquad (7)$$

where, in addition to the variables introduced earlier,

$\overline{n}$ = full employment expressed in logarithms.

The full model given by equations (4-7) can be reduced to the following set of equations:

$$\dot{p} = \frac{\alpha_1}{\alpha_2}\beta + \frac{\alpha_1}{\alpha_2}(1-\gamma)n - \frac{m}{\alpha_2} + \frac{p}{\alpha_2} \qquad (8)$$

$$\dot{n} = \theta(\delta - \gamma n - w + p) \qquad (9)$$

$$\dot{w} = \eta(n - \overline{n}) \qquad (10)$$

This can be expressed in matrix form as:

$$\begin{pmatrix} \dot{p} \\ \dot{n} \\ \dot{w} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_2} & \frac{\alpha_1(1-\gamma)}{\alpha_2} & 0 \\ \theta & -\theta\gamma & -\theta \\ 0 & \eta & 0 \end{pmatrix} \begin{pmatrix} p - p^* \\ n - n^* \\ w - w^* \end{pmatrix} \qquad (11)$$

[1] Various extensions to the Cagan (1956) model are considered in Chapter 3 of Turnovsky (2000). In Chapter 7, Section 7.2, he considers a similar model to that considered here but with sluggish adjustment only in wages. The model introduced here in equations (4-7) is an extension of Turnovsky's approach.

where an asterisk denotes the corresponding steady-state value and where these steady-state values are given by:

$$p^* = m - \alpha_1\beta - \alpha_1(1-\gamma)\overline{n} \tag{12}$$

$$n^* = \overline{n} \tag{13}$$

$$w^* = m - \alpha_1\beta - \alpha_1(1-\gamma)\overline{n} + \delta - \gamma\overline{n} \tag{14}$$

3. DERIVING THE ANALYTIC SOLUTION

Real-valued and complex-valued eigenvalues

In order to derive the full analytic solution to this model, we must first demonstrate that it is possible for the dynamical system defined by equation (11) to have complex-valued eigenvalues. Letting $\gamma \to 1$, the characteristic equation of this system satisfies:

$$c(\lambda) = \begin{vmatrix} \frac{1}{\alpha_2} - \lambda & 0 & 0 \\ \theta & -\theta - \lambda & -\theta \\ 0 & \eta & -\lambda \end{vmatrix}$$

$$= (\tfrac{1}{\alpha_2} - \lambda)\left[(\theta + \lambda)\lambda + \theta\eta\right]$$

$$= (\tfrac{1}{\alpha_2} - \lambda)\left[\lambda^2 + \theta\lambda + \theta\eta\right] \tag{15}$$

Hence the eigenvalues of the system are given by:

$$\lambda_1 = \frac{1}{\alpha_2} > 0\text{; and } \lambda_2, \lambda_3 = \frac{-\theta \pm \sqrt{\theta^2 - 4\theta\eta}}{2}. \tag{16}$$

Note that λ_2 and λ_3 both have negative real parts and that both λ_2 and λ_3 are complex-valued whenever $4\eta > \theta$. Accordingly, if the system has at least one complex-valued eigenvalue then it will have two complex-valued eigenvalues each with negative real parts. Furthermore, if all other parameters are fixed, the size of the imaginary part of the eigenvalues can be determined by changing the value of η. For low value of η, both λ_2 and λ_3 will be real-valued. For larger value of η, both λ_2 and λ_3 will have small imaginary part. For even larger value of η, both λ_2 and λ_3 will have large imaginary part.

(Table 1 about here)

Using continuity, it can be demonstrated that these general properties will also apply for γ less than 1 but still close to 1. In particular, this paper will calibrate the model given by Equations (4-7) using the following parameter values: $\alpha_1 = 1.0$, $\alpha_2 = 0.5$, $\beta = 0.0$, $\gamma = 0.5$, $\delta = \log(0.5)$, $\theta = 1.0$, and $\bar{n} = 1.0$. Subsequently, we will consider the following experiment: initially the model is in the steady-state associated with $m = 0.0$; then an unanticipated shock moves the nominal money supply to $m = 0.1$; this paper investigates the solutions to transitional dynamics as the economy adjusts towards its new steady-state.

Using the chosen parameter values as well as the four different choices of η give values for λ_1, λ_2 and λ_3 as described in Table 1.

General solution with real-valued eigenvalues

Letting λ_1, λ_2 and λ_3 denote the eigenvalues of the system given by equation (11), irrespective of whether those eigenvalues are real-valued or complex-valued. Then the eigenvectors of the system are given by:

$$\mathbf{v}(\lambda_i) = \begin{pmatrix} \alpha_1(1-\gamma)\lambda_i \\ (\alpha_2\lambda_i - 1)\lambda_i \\ \eta(\alpha_2\lambda_i - 1) \end{pmatrix} \qquad (17)$$

and the general solution to the system is given by:

$$\begin{pmatrix} p-p^* \\ n-n^* \\ w-w^* \end{pmatrix} = \left[\mathbf{v}(\lambda_1) \quad \mathbf{v}(\lambda_2) \quad \mathbf{v}(\lambda_3)\right] \begin{pmatrix} C_1 \exp(\lambda_1 t) \\ C_2 \exp(\lambda_2 t) \\ C_3 \exp(\lambda_3 t) \end{pmatrix} \tag{18}$$

where C_1, C_2 and C_3 are yet-to-be-determined constants.

Irrespective of whether the eigenvalues are real-valued or complex-valued, in the cases that we consider it is always the case that precisely one eigenvalue has positive real part (the unstable eigenvalue) and two eigenvalues have negative real part (the stable eigenvalues). Without loss of generality, assume that λ_1 has positive real part. Then the stable solution is given by setting $C_1 = 0$ so that:

$$\begin{pmatrix} p-p^* \\ n-n^* \\ w-w^* \end{pmatrix} = \left[\mathbf{v}(\lambda_2) \quad \mathbf{v}(\lambda_3)\right] \begin{pmatrix} C_2 \exp(\lambda_2 t) \\ C_3 \exp(\lambda_3 t) \end{pmatrix} \tag{19}$$

The standard approach to solving this model is to assume that one variable "jumps" to the stable solution, while other variables evolve continuously from their historically determined positions.[2] In our interpretation of this model, we assume that *p* is the jump variable. Then the constants, C_2 and C_3, are chosen consistent with the initial values of *n* and *w*. When the eigenvalues are all real-valued, C_2 and C_3 are real-valued constants and equation (19) fully determines the analytic solution of the model.

General solution with complex-valued eigenvalues

When some of the eigenvalues can be complex-valued, the stable eigenvalues, λ_2 and λ_3, form a complex conjugate pair. As a consequence, the corresponding eigenvectors, $\mathbf{v}(\lambda_2)$ and $\mathbf{v}(\lambda_3)$, are complex conjugates. Finally, in order for the solutions to *p, n* and *w* to be real-valued, it is necessary that C_2 and C_3 also form a

[2] See Blanchard and Kahn (1980).

complex conjugate pair. Hence, letting a bar above a parameter denote its complex conjugate, the general solution in the case of complex-valued eigenvalues can be written in the form:

$$\begin{pmatrix} p-p^* \\ n-n^* \\ w-w^* \end{pmatrix} = \begin{bmatrix} \mathbf{v}(\lambda_2) & \mathbf{v}(\bar{\lambda}_2) \end{bmatrix} \begin{pmatrix} C_2 \exp(\lambda_2 t) \\ \bar{C}_2 \exp(\bar{\lambda}_2 t) \end{pmatrix} \qquad (20)$$

Let $\lambda_2 = -\mu_a + i\mu_b$ and $C_2 = D_a + iD_b$ and equation (20) can be rewritten in the form:

$$\begin{pmatrix} p-p^* \\ n-n^* \\ w-w^* \end{pmatrix} = \begin{pmatrix} -\alpha_1(1-\gamma)\mu_a + i\alpha_1(1-\gamma)\mu_b & -\alpha_1(1-\gamma)\mu_a - i\alpha_1(1-\gamma)\mu_b \\ \alpha_2(\mu_a^2-\mu_b^2)+\mu_a - i(2\alpha_2\mu_a\mu_b+\mu_b) & \alpha_2(\mu_a^2-\mu_b^2)+\mu_a + i(2\alpha_2\mu_a\mu_b+\mu_b) \\ -\eta\alpha_2\mu_a - 1 + i(\eta\alpha_2\mu_b) & -\eta\alpha_2\mu_a - 1 - i(\eta\alpha_2\mu_b) \end{pmatrix}$$

$$\cdot\begin{pmatrix} (D_a+iD_b)\exp(-\mu_a t)[\cos(\mu_b t)+i\sin(\mu_b t)] \\ (D_a-iD_b)\exp(-\mu_a t)[\cos(\mu_b t)-i\sin(\mu_b t)] \end{pmatrix} \qquad (21)$$

If the constants, D_a and D_b, are chosen consistent with the initial values of *n* and *w*, then equation (21) fully determines the analytic solution of the model with complex-valued eigenvalues, providing real-valued solutions for *p, n* and *w.* It will be observed that these solutions include the terms, $\cos(\mu_b t)$ and $\sin(\mu_b t)$ thus demonstrating that the analytic solution has cyclic properties and, in particular, that the frequency of the cycles increases as the absolute value of μ_b increases.

Plotting the analytic solutions

(Figures 1 and 2 about here)

For the calibration parameters presented in Table 1, it will be observed that the absolute value of the imaginary part (μ_b) of the stable eigenvalues increases as η increases. A three-dimensional phase diagram for the four sets of eigenvalues is presented in Figure 1, while Figure 2 presents corresponding time plots of each of the three variables, *p, n* and *w.* As anticipated, these figures demonstrate that the frequency of the cycles increases as η (and hence the absolute value of μ_b) increases. Since our intention is to investigate the success of various algorithms in coping with highly oscillatory behaviour, throughout the rest of this paper we will focus on the outcome for wages when $\eta = 100$.

4. COMPARING ALTERNATIVE ALGORITHMS

Methodological approach

We can employ the above model to examine how well the reverse-shooting and forward-shooting algorithms cope with the type of cyclic behaviour generated by complex eigenvalues. A full analytic solution can be derived for the model and this analytic solution can be used as a benchmark for the solutions derived using the two algorithms. In this case, where an analytic solution is available, there is really no need to use the algorithms to solve the model at all. But the comparisons between the benchmark analytic solution and the solutions derived using the competing algorithms provide good indications of how well the reverse-shooting and forward-shooting algorithms are likely to cope when faced with a non-linear model that exhibits cyclic behaviour. Of course, in general, in the case of such a non-linear model, no analytic solution exists.

Illustrating the alternative approaches

We can illustrate the reverse-shooting and forward-shooting approaches by rewriting the general solution of the model given in equation (18) in the form:

$$\mathbf{x}(t) - \mathbf{x}^* = \begin{bmatrix} \mathbf{v}(\lambda_1) & \mathbf{v}(\lambda_2) & \mathbf{v}(\lambda_3) \end{bmatrix} \begin{pmatrix} C_1 \exp(\lambda_1 t) \\ C_2 \exp(\lambda_2 t) \\ C_3 \exp(\lambda_3 t) \end{pmatrix} \qquad (22)$$

The forward-shooting solution to this model is given by first fixing an initial value for t, given by t_0, then searching over the values for (C_1, C_2, C_3) until a solution is found that is consistent with the initial values of the non-jump variables and arrives within a suitably small neighbourhood of the steady-state, $\mathbf{x}^*$. In this sense, the forward-shooting solution is equivalent to searching over a three-dimensional space.

To find the reverse-shooting solution, it is first necessary to write the model in reverse time, so that $\mathbf{z}(t) = \mathbf{x}(-t)$ and:

$$\mathbf{z}(t) - \mathbf{z}^* = \begin{bmatrix} \mathbf{v}(\lambda_1) & \mathbf{v}(\lambda_2) & \mathbf{v}(\lambda_3) \end{bmatrix} \begin{pmatrix} C_1 \exp(-\lambda_1 t) \\ C_2 \exp(-\lambda_2 t) \\ C_3 \exp(-\lambda_3 t) \end{pmatrix} \tag{23}$$

Without loss of generality, start at $t_0 = -N$ where N is a large positive number and choose $\mathbf{z}(t_0)$ close to $\mathbf{z}^*$. Then $C_i \exp(\lambda_i N)$ is close to zero for $i = 1, 2, ..., m$. If λ_i is an unstable eigenvalue, then $\exp(\lambda_i N)$ is a large positive number; hence C_i must be close to zero. On the other hand, if λ_i is a stable eigenvalue, then $\exp(\lambda_i N)$ is close to zero; hence C_i can take any value.

Since λ_1 has positive real part, and both λ_2 and λ_3 have negative real parts, equation (23) reduces to:

$$\mathbf{z}(t) - \mathbf{z}^* = \begin{bmatrix} \mathbf{v}(\lambda_2) & \mathbf{v}(\lambda_3) \end{bmatrix} \begin{pmatrix} C_2 \exp(-\lambda_2 t) \\ C_3 \exp(-\lambda_3 t) \end{pmatrix} \tag{24}$$

Then a solution is found by searching over the values for (C_2, C_3) until the associated trajectory arrives within a suitably small neighbourhood of the history-determined values for the non-jump variables of $\mathbf{x}(t)$ and hence for the non-jump variables of $\mathbf{z}(t)$. In this sense, the reverse-shooting solution is equivalent to searching over a two-dimensional space. A similar story can be told for the case when λ_2 and λ_3 are complex-valued.

Thus the reverse-shooting algorithm is likely to be more efficient than the forward-shooting algorithm in the sense that the reverse-shooting algorithm requires searching over a two-dimensional space whereas the forward-shooting algorithm requires searching over a three dimensional space.

The computational problem

The associated computational problem requires the solution of the above model from a known meaningful steady state, $\mathbf{x}_0^*$, to a new known meaningful steady state, $\mathbf{x}^*$, after an exogenous shock in *m*. The problem is to find the unique trajectory (in *p, n* and *w*) from the initial steady state to the final steady state resulting from the shock.

The fundamental problem is to find the stable solutions for the following dynamical system:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{q}) \tag{25}$$

where the state vector is given by:

$$\mathbf{x} = \left[p, n, w\right]^T \tag{26}$$

and where $\mathbf{q}$ is a vector of parameters.

The shock in the money supply determines the boundary conditions for the model and gives rise to the specific exercise we solve. Before the shock the model is at $\mathbf{x_0^*} = \left[p_0^*, n_0^*, w_0^*\right]^T$ and evolves along a unique stable solution trajectory to $\mathbf{x}^* = \left[p^*, n^*, w^*\right]^T$ as given in equations (25-26). The problem is to find this stable trajectory. Hence, while the initial values of *n* and *w* are predetermined by history, the initial condition for *p* is not known. The basic problem of this computational exercise is to find the initial condition for *p,* which must be chosen so that it lies on the stable trajectory.

The exercise is a two-point boundary value problem where the aim is to find the stable trajectory of the model. The exercise is difficult due to the unstable nature of the model problem. Basically for the reverse-shooting algorithm we need to search around points "near enough" to the final steady state so that the solution to the model has transient dynamics in reverse time that are forced onto the stable manifold and that also satisfy the appropriate initial values for *n* and *w*. For the forward-shooting algorithm we search around points at the initial values of *n* and *w*, to find an initial value for the variable, *p*, so that the solution to the model has transient dynamics in forward time that pass "near enough" to the final steady-state. Both searches will determine a solution trajectory and initial conditions, ($p(0), n(0), w(0)$).

To program the exercise, software components are needed to solve differential equations and undertake searches for a range of parameter sets. We used Matlab (Mathworks, 2003) as it is ideally suited for this type of computational problem. The programming was written so as to make use of key Matlab features. Library routines (toolboxes) were used so that state-of-the art solvers and searches are included in the code. All results were generated using Version 6.1 of Matlab.

Both shooting algorithms considered here have two essential components: a differential equation solver to solve for each candidate path and a search routine that chooses among possible candidate paths and determines when an acceptable candidate path has been found. For the search method we use a Nelder-Meade direct simplex search (Lagarias, Reeds, Wright and Wright, 1998). We implement this search by the Matlab function *fminsearch*. There is a range of differential equation solvers available in Matlab (Shampine and Reichelt, 1997).

Runge-Kutta solver

The usual solver of first choice is the Runge-Kutta algorithm. We initially use a variable time step size Runge-Kutta solver as implemented by calling the Matlab function *ode45*. This solver uses the explicit 4,5 pair of the Runge-Kutta formula (Dormand and Prince, 1980). It is a single-step solver so that only the solution at the previous time-step is used in determining the current solution. It is informative to compare solutions derived using the Runge-Kutta ODE solver with the correct (analytic) solution. In Figure 3, we compare the reverse-shooting and forward-shooting solutions for the most oscillatory case, which is the outcome for wages when $\eta = 100$.

(Figure 3 about here)

Using this ODE solver, both reverse-shooting and forward-shooting solutions have difficulty reproducing the correct solution. In particular, both solutions tend to produce cycles of larger amplitudes than the true solution as each derived solution gets further away from its starting point. Thus, for the reverse-shooting solution, which starts close to the final steady-state, the amplitude close to the initial point on the stable solution is much larger than the correct solution. Similarly, for the forward-shooting solution, which starts close to the initial point on the stable solution, the amplitude close to the final steady-state is much larger than the correct solution. The problem is clearly worse for reverse-shooting than it is for forward-shooting.

ABM solver

We next investigate the success rates of the two shooting algorithms using the Adams-Bashforth-Moulton predictor-corrector algorithm (Gear, 1971; Shampine and Gordon, 1975; Shampine and Reichelt, 1997). We will henceforth refer to this algorithm as the ABM solver. This is a multi-step algorithm, which usually needs several preceding time point solutions to compute the current time point solution. The ABM solver is regarded as better than the Runge-Kutta solver when the ODE function is expensive to calculate, as is the case with highly oscillatory solutions like those considered here, and has been implemented using the Matlab function *ode113*.

(Figures 4 and 5 about here)

We first implement the ABM solver using the default step-size parameter. Figure 4 shows that, with the default step-size, the overall solution time-path for both reverse and forward-shooting are close to the analytic solution. However, for both shooting approaches, the detail near the steady-state is less accurate. Figure 5 repeats this exercise using the same solver but with a smaller step-size. This approach is able to produce a near-perfect solution for both shooting approaches.

5. CONCLUDING COMMENTS

We have presented a model, which has a saddle-path property in the form of one unstable and two stable eigenvalues. With a simple change to the parameter configuration, it is possible to ensure that the two stable eigenvalues are either real-valued (with monotonic or hump-shaped dynamics) or complex-valued (with cyclic dynamics). Also, for the complex-valued eigenvalues, the number of cycles increases as the imaginary parts of the complex-valued eigenvalues increase in absolute size.

This paper then focuses on the properties of calibrated solutions to the model for the most oscillatory outcome, the case of wages when $\eta = 100$. Because the chosen model is linear, it has been possible to derive a correct (analytic) solution and then use this analytic solution as a benchmark when evaluating two alternative algorithms. The algorithms that are considered here are the well-known reverse-shooting and forward-shooting approaches. Our results have shown that the results are sensitive to the choice of ODE solver.

When the Runge-Kutta solver is used both types of algorithms have difficulty reproducing highly cyclic dynamics and, in particular, in reproducing the amplitude of models, which have complex-valued eigenvalues with large imaginary parts. The problem is clearly worse for reverse-shooting than it is for forward-shooting. However, when the ABM solver is used with a sufficiently small step-size, all these problems are eliminated and a near-perfect solution can be derived.

Remember, for this model, which can be reduced to a linear dynamical system, the correct solution exists and we have been able to benchmark our results from the shooting algorithms against the correct solution. We have shown that the choice of ODE solver has a significant impact on the likely success of the shooting algorithms. These results have broader ramifications, too. The varying levels of success experienced using different ODE solvers as they try to reproduce a highly cyclic model are likely to carry over to non-linear models, which also exhibit highly cyclic behaviour.

REFERENCES

Blanchard, O. J., and C. M. Kahn (1980), “The Solution of Linear Difference Models under Rational Expectations,” *Econometrica*, 48, pp. 1305-1311.

Cagan, P. (1956), “The Monetary Dynamics of Hyperinflation,” in M Friedman (editor), *Studies in the Quantity Theory of Money*, University of Chicago Press, Chicago.

Dormand, J. R. and P. J. Prince (1980, A Family of Embedded Runge-Kutta Formulae, *Journal of Computing and Applied Mathematics.*, 6, pp. 19-26.

Gear, C.W. (1971) *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA

Judd, K. L. (1998), *Numerical Methods in Economics*, MIT Press, Cambridge Massachusetts.

Lagarias, J. C., J. A. Reeds, M. H. Wright and P. E. Wright (1998), Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, *SIAM Journal of Optimization*, 9(1): p.112-147.

MathWorks (2003), website: www.mathworks.com,

Shampine, L. F. and M. K. Gordon (1975), *Computer Solution of Ordinary Differential Equations: The Initial Value Problem*, W. H. Freeman, San Francisco.

Shampine, L. F., and M. W. Reichelt (1997), The MATLAB ODE Suite, *SIAM Journal on Scientific Computing*, 18(1).

Turnovsky, S. J. (2000), *Methods of Macroeconomic Dynamics*, Second Edition, MIT Press, Cambridge Massachusetts and London England.

TABLES AND FIGURES

η	λ_1	λ_2	λ_3
0.2	2.3	-0.50	-0.34
0.5	2.3	-0.41+0.51i	-0.41-0.51i
10.0	2.1	-0.32+3.0i	-0.32-3.0i
100.0	2.0	-0.26+9.9i	-0.26-9.9i

Table 1
Values of Eigenvalues for Different Values of η

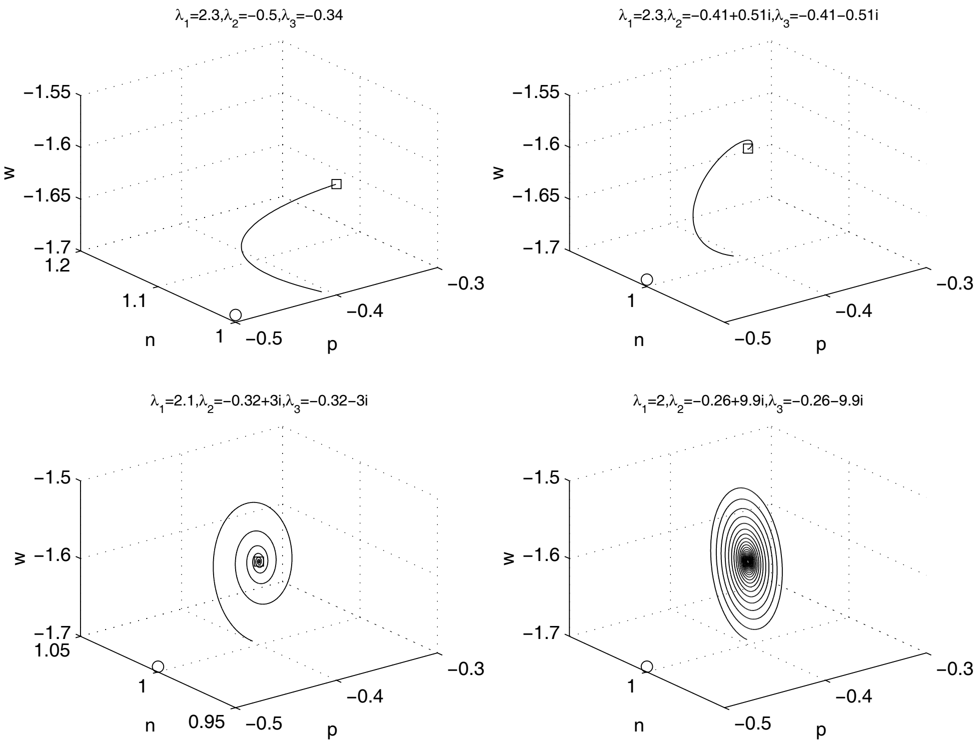
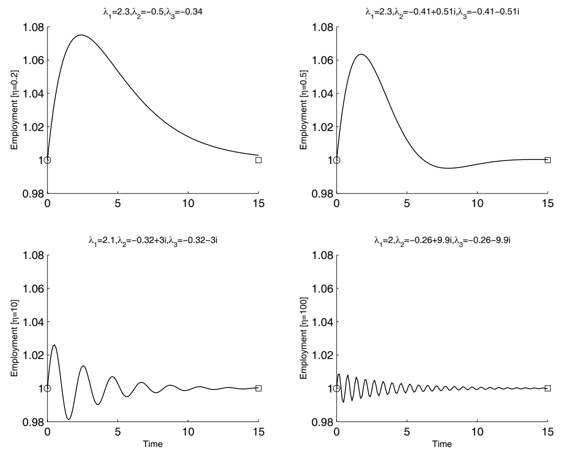
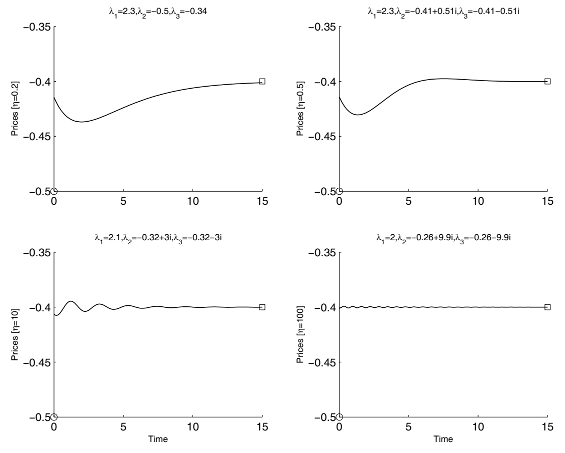


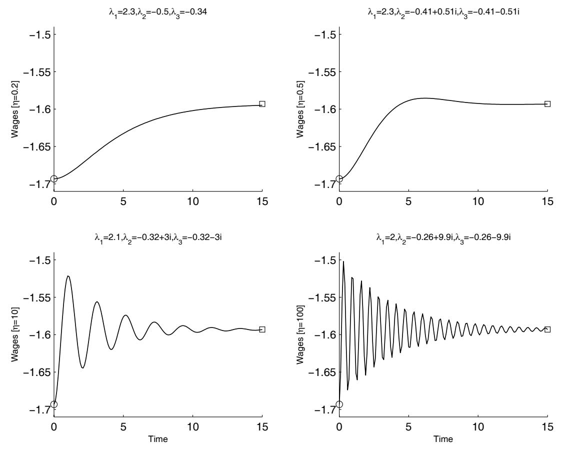
Figure 1: Phase Diagram of Analytic Solution



(a) Employment

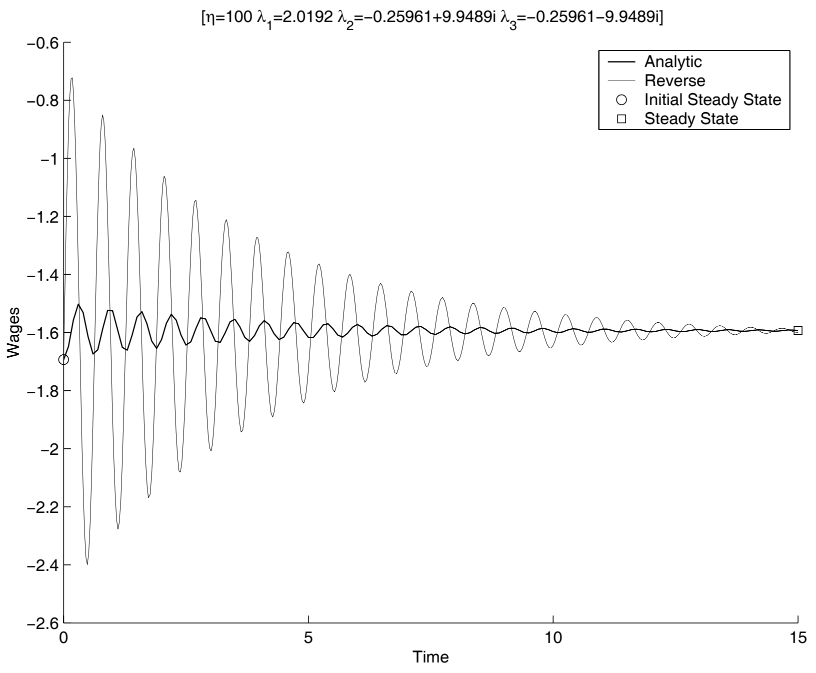


(b) Prices

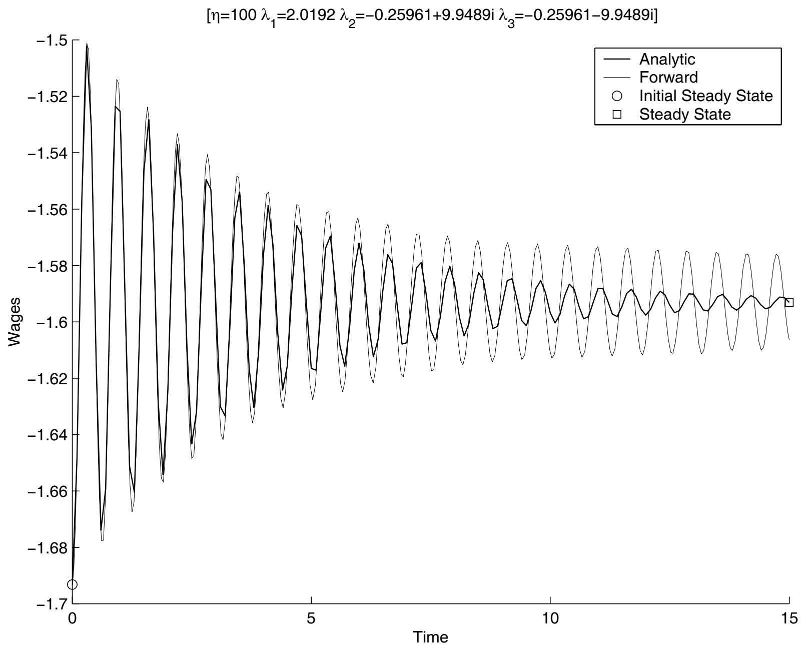


(c) Wages

Figure 2: Time Plots of Analytic Solution

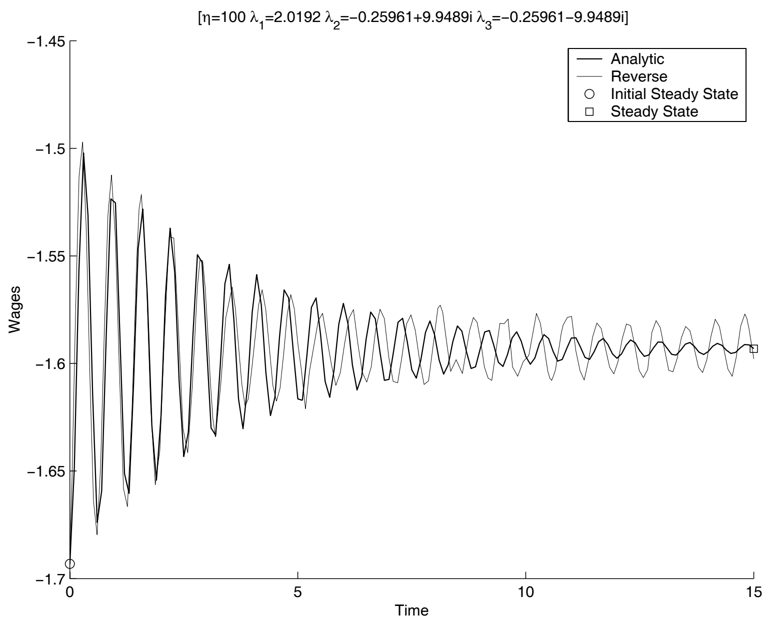


(a) Reverse-Shooting Solution

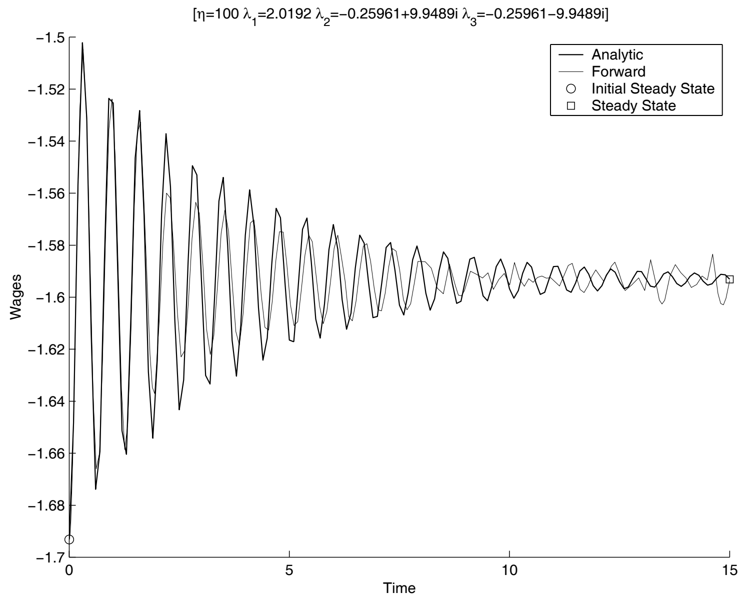


(b) Forward-Shooting Solution

Figure 3: Time Plots of Wages with Frequent Cycles ($\eta = 100$)
Comparison of Reverse-Shooting and Forward-Shooting Solutions Using Runge-Kutta ODE Solver.

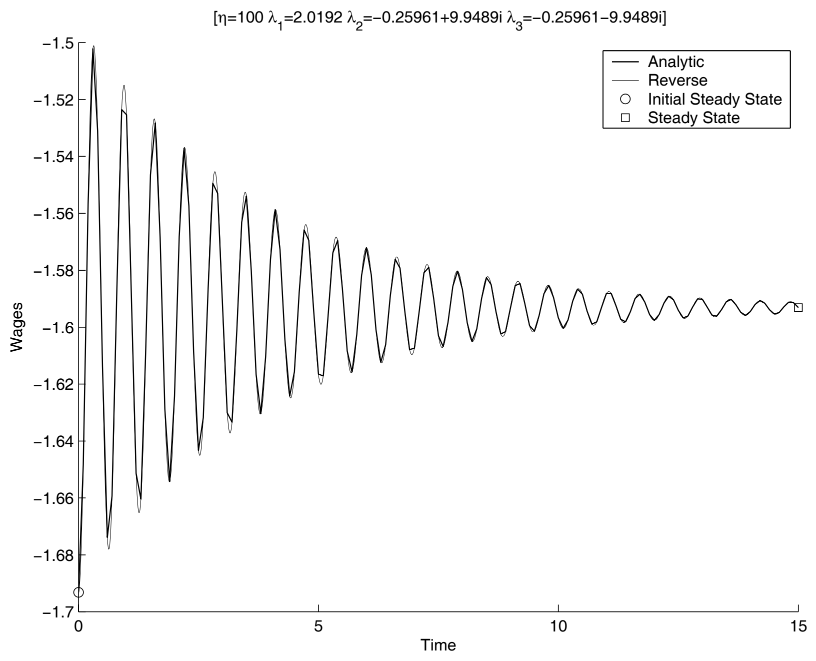


(a) Reverse-Shooting Solution

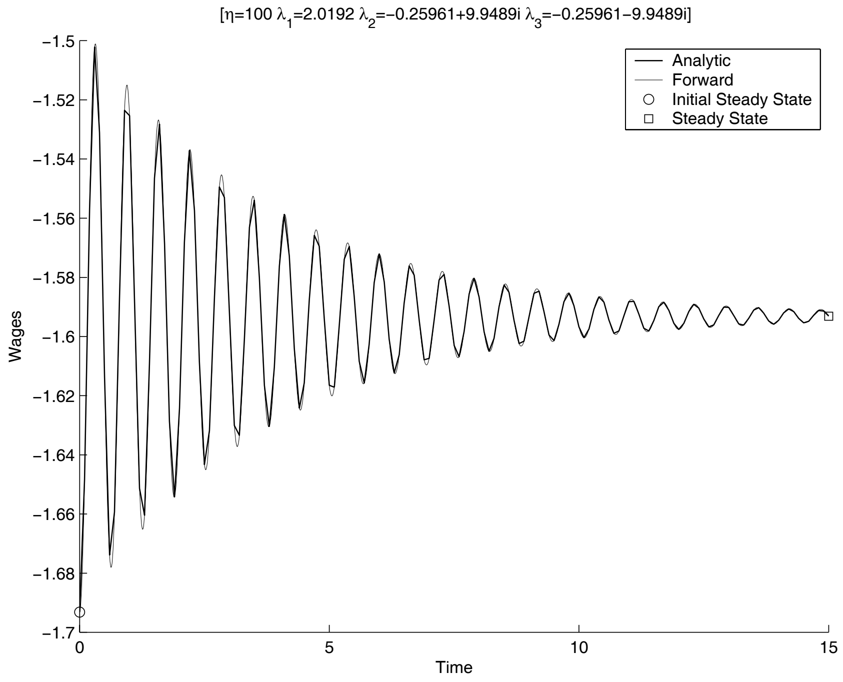


(b) Forward-Shooting Solution

Figure 4: Time Plots of Wages with Frequent Cycles ($\eta = 100$)
Comparison of Reverse-Shooting and Forward-Shooting Solutions Using ABM ODE Solver with Default (Large) Step-size.



(a) Reverse-Shooting Solution



(b) Forward-Shooting Solution

Figure 5: Time Plots of Wages with Frequent Cycles ($\eta = 100$)
Comparison of Reverse-Shooting and Forward-Shooting Solutions Using ABM ODE Solver with Small Step-size.