CESifo *Working Paper Series*

EXPLOITING MODEL STRUCTURE TO SOLVE THE DYNAMICS OF A MACRO MODEL

Ric D. Herbert
Peter J. Stemp*

Working Paper No. 266

March 2000

CESifo
Poschingerstr. 5
81679 Munich
Germany
Phone: +49 (89) 9224-1410/1425
Fax: +49 (89) 9224-1409
http://www.CESifo.de

* Peter Stemp is grateful to the Australian Research Council (ACR) and the Faculty of Economics and Commerce at the University of Melbourne for financial support through an ARC Small Grant and a Faculty Research Grant.

CESifo *Working Paper No. 266*
March 2000

EXPLOITING MODEL STRUCTURE TO SOLVE THE DYNAMICS OF A MACRO MODEL

Abstract

This paper considers alternative approaches to solving the time-path of a representative agent model following an exogenous shock. The model has a number of important dynamic properties that are both common to a wide range of economic models and have important computational implications for solving the model. The paper compares the alternative approaches on a computational basis and shows that the best approach to solving the model is obtained by fully exploiting the model structure in a modified reverse shooting algorithm.

Keywords: Representative agent models, shooting methods, MATLAB

JEL Classification: C6, E1

Ric D. Herbert
School of Business
The University of Newcastle
Ourimbah
New South Wales, 2258
Australia
e-mail: rherbert@mail.newcastle.edu.au

Peter J. Stemp
Department of Economics
The University of Melbourne
Melbourne
Victoria, 3010
Australia
e-mail: p.stemp@ecomfac.unimelb.edu.au

1. INTRODUCTION

In this paper we consider the computational aspects of solving a well-known representative agent model (Matsuyama, 1987). This model has a number of important dynamic properties. These properties have significant implications, common to a range of macroeconomic models, for computing the model solution. Firstly the model has a number of stable and unstable trajectories so that it is likely to be complicated to solve the model for a stable solution. The economy is initially at a stable steady-state equilibrium, and when shocked by, say, an exogenous change in world interest rates, then it moves to a stable trajectory leading to a new steady-state equilibrium. The movement to the new equilibrium is assumed to come about as a consequence of optimising behavior of the agents in the model. In the model, certain variables jump instantaneously after the shock, and force the economy onto the trajectory leading to the stable equilibrium.

A second property of the model is that it is nonlinear with nonlinearities arising as a direct consequence of optimising behavior by the representative agents. The usual approach is to linearise the model in the neighborhood of the steady-state and then solve the linearised model. This approach can be particularly unreliable if the jumps required to bring the economy back onto a stable path are particularly large.

These properties, especially the property of jumps to the stable path, introduce some interesting challenges to solving the model.

The essential issue is that of finding the stable manifold giving the dynamic solution. The jumps in the variables ensure that the model solution is on the stable manifold. This issue has been considered, especially in the case of rational expectations variables, by a number of authors including Anderson and Moore (1985), Blanchard and Kahn (1980), Boucekkine (1995), Fair and Taylor (1983), and Zadrozny (1998).

In this paper we use shooting methods to find the stable manifold. We show that, by exploiting the structure of the model, the computational effort in solving the dynamics of the model can be substantially reduced.

2.1 *Defining the model structure*

The chosen model is a real model of a small open economy in a one-product world. The economy is assumed to be so small in the international market for tradeable goods that it is a price-taker in the market for foreign exchange. Agents in the economy also face perfect capital markets and a given world interest rate, r.

There are four sectors in this economy: the corporate sector, the household sector, the government sector and the external sector. These four sectors can be aggregated to yield a model of a small open economy given by the following set of equations[1]:

$$\dot{q} = \left\{r - b\left[\Lambda(q)^2\right]\right\} q - F_K \tag{1}$$

$$\frac{\dot{K}}{K} = \Lambda(q)[1 - b\Lambda(q)] \tag{2}$$

$$\dot{C} = (r - \theta)C - p(p + \theta)[qK - D + B] \tag{3}$$

$$\dot{D} = C - F(K, 1) + G + K\Lambda(q) + rD \tag{4}$$

where

$F = F(K, L) = aK^{\alpha}L^{1-\alpha}$ represents the output of the firm;

K represents the real capital stock;

$L = 1$ This represents the demand and supply of labour;

q represents average (and marginal) Tobin's q;

$\Lambda(q) = \frac{q-1}{2bq}$;

C represents real aggregate consumption;

r represents the real world interest rate (assumed exogenous);

p represents the instantaneous probability of death per unit time for representative consumers;

[1] For convenience the dot notation is used for time derivatives, and the subscript notation is used for other derivatives. Time is not explicitly included except where necessary.

θ represents the consumer's rate of time preference;

D represents overseas debt;

B represents domestic holdings of government bonds;

G represents government expenditure (assumed exogenous and fixed).

There are four endogenous variables in the model, given by q, K, C and D. The other parameters and variables given by a, α, p, θ, r, b, G and B are exogenously fixed. All variables and parameters as well as the functional form of $\Lambda(q)$ have been defined above.

The model is autonomous and with the exogenously fixed parameters and variables, can be written in a state space form as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) \tag{5}$$

where

$\mathbf{x} = [q, K, C, D]^T$ are the states or vector of endogenous variable.

The model can be solved by using equation 5. But the model has a block recursive structure with C and D not appearing in equations 1 and 2. This allows the model to be broken into an investment sub-model (equations 1 and 2) and a consumption sub-model (equations 3 and 4). This paper demonstrates how model structure can be exploited in solving the model.

The objective of this paper is then to find a suitable solution approach that will define the trajectory of a stable No-Ponzi game solution with a plausible economic interpretation. This leads to a unique finite-valued steady-state, $\mathbf{x}^*$.

Throughout this paper it will be assumed that the model has been calibrated using plausible parameter values. We assume a balanced budget, giving $B = 0$. We also assume that government expenditure is a fixed proportion of steady-state output and is given by $G = \frac{1}{4}F(K^*, 1)$.

2.2 Dynamic properties of the linearised model

Linearising the model in the neighborhood of the steady-state yields the following fourth-order linear dynamic system with an asterisk indicating a corresponding steady-state value:

$$\dot{\mathbf{x}} = A(\mathbf{x} - \mathbf{x}^*) \tag{6}$$

where

$$A = \begin{bmatrix} r & -F_{KK} & 0 & 0 \\ \frac{K^*}{2b} & 0 & 0 & 0 \\ -p(p+\theta)K^* & -p(p+\theta) & r-\theta & p(p+\theta) \\ \frac{K^*}{2b} & -r & 1 & r \end{bmatrix}$$

Expressing the linearised model in this way clearly indicates that the model has a block-recursive structure, where the dynamics of q and K can be solved independently of the dynamics of C and D. This means that the dynamic model can be solved in two steps, first solving the investment sub-model, which defines a second-order system in q and K. The full model can then be solved by substituting solutions for q and K into the $\dot{C}$ and $\dot{D}$ equations and then solving for the second-order system in C and D (the consumption sub-model). This two-step solution approach can also be applied to the true (nonlinear) model given by equations 3 and 4.

A general idea about the stability properties of the true (nonlinear) model can be obtained by examining the stability properties of the linearised system. The eigenvalues for this system are given by:

$$\lambda_1, \lambda_2 = \frac{r \pm \sqrt{r^2 - 2F_{KK}\frac{K^*}{b}}}{2} \tag{7}$$

$$\lambda_3 = r + p \tag{8}$$

$$\lambda_4 = r - \theta - p \tag{9}$$

Since $F_{KK} = a\alpha(\alpha - 1)(K^*)^{\alpha-2} < 0$, equation 7 defines two real-valued eigenvalues, one positive and one negative. Henceforth, it is assumed that $\lambda_1 > r > 0 > \lambda_2$. Also, if it is assumed that $\theta < r < \theta + p$ then $\lambda_3 > 0 > \lambda_4$. Then, assuming that $\lambda_2 \neq \lambda_4$, there are two real-valued positive eigenvalues given by λ_1 and λ_3, and two distinct real-valued negative eigenvalues, given by λ_2 and λ_4.

Thus both the investment sub-model and the C and D components of the full model will have one positive and one negative eigenvalue, thereby exhibiting the property of saddle-path instability. As a consequence, following an exogenous shock to the system it will be necessary for one of the K and q variables and one of the C and D variables to jump instantaneously so as to ensure stability of the solution. Since K and D are stock variables which cannot jump instantaneously in this model, it is appropriate that q and C should be the jump variables. These properties of the linearised model carry over to the true (nonlinear) model.

2.3 *Calibrating the model*

It is assumed that $a = 1$, $r = 0.05$, $\alpha = 0.3$ $b = 5$, $p = 0.05$ and $\theta = 0.045$. From this set of parameters the steady-state is given by:

$q^* = 1$,

$K^* = 12.9314$,

$C^* = 1.0237$, and

$D^* = 11.8538$.

The eigenvalues at the steady-state are:

$\lambda_1 = 0.0892$,

$\lambda_2 = -0.0392$,

$\lambda_3 = 0.1000$, and

$\lambda_4 = -0.0450$.

The eigenvalues are all of a similar order of magnitude. Further, they are small in

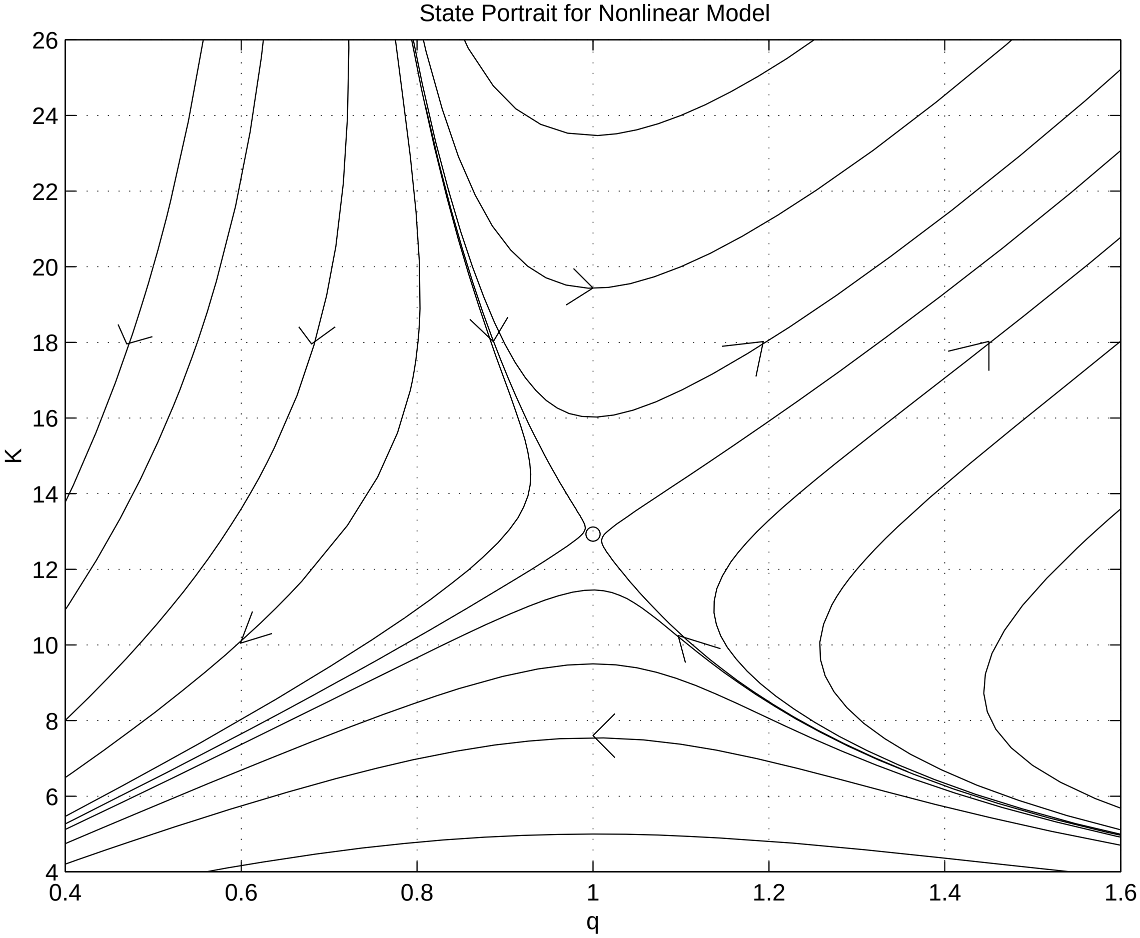

Figure 1. Phase Diagram of the Investment Sub-Model: True (Non-Linear) Model.

magnitude so the model evolves slowly with time. These properties mean that a standard numerical solution algorithm can be used to solve the model. The model solutions presented in this paper were generated using a variable step-size Runge-Kutta-Fehlberg algorithm (Burden and Faires, 1985) as implemented by the Matlab function `ode45` with default options (eg. tolerance values) (Mathworks, 1998). Other solvers have been implemented, ranging from fixed step-size Euler solvers to multi-step, predictor-corrector Adam-Bashforth-Moulton solvers, but little computational benefit was found over the standard Runge-Kutta solver.

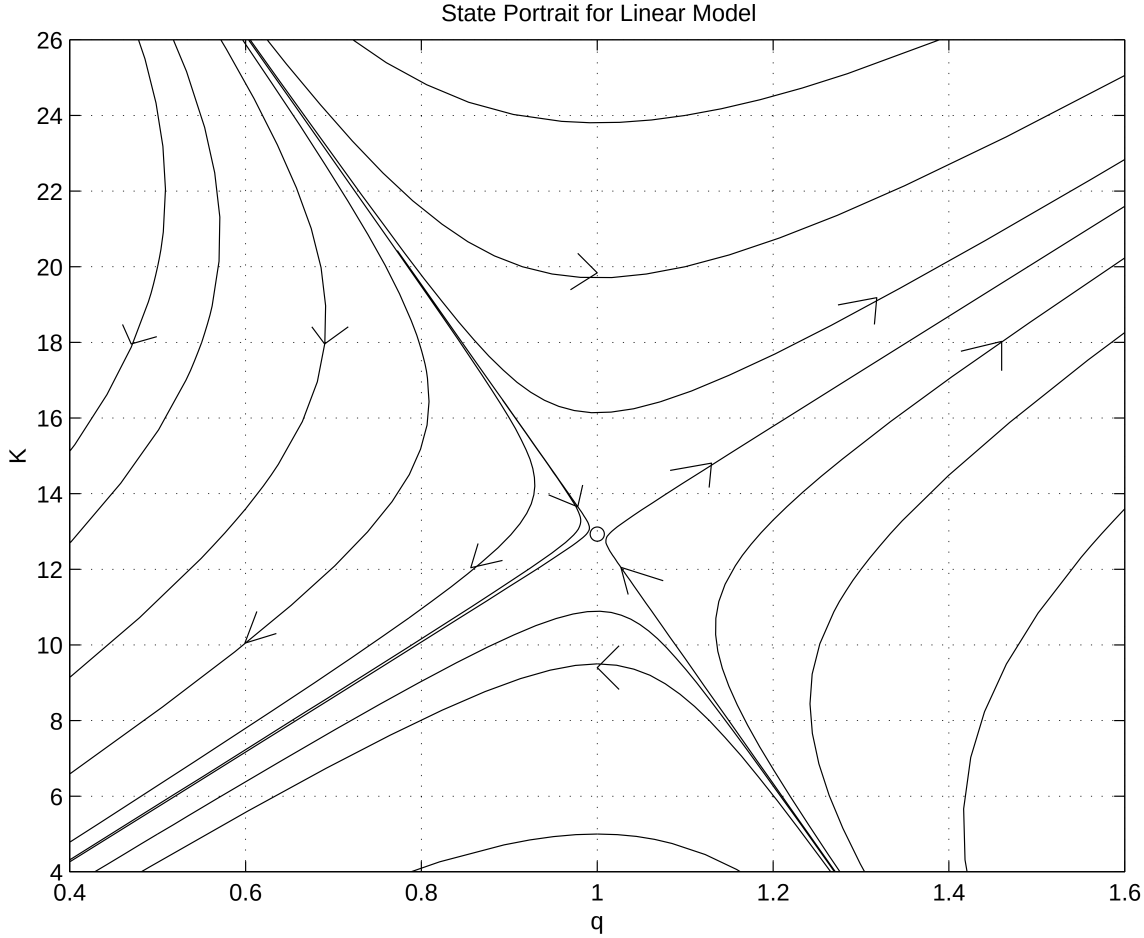

Figure 2. Phase Diagram of the Investment Sub-Model: Linearised Model.

3. SOLVING THE INVESTMENT SUB-MODEL

3.1 *Solving the dynamics*

The investment sub-model is given by equations 1 and 2. In state-space form it can be written as:

$$\dot{\mathbf{x}}_{\mathbf{I}} = \mathbf{f}_{\mathbf{I}}(\mathbf{x}_{\mathbf{I}}) \tag{10}$$

where

$\mathbf{x}_{\mathbf{I}} = [q, K]^T.$

The corresponding linearised model is given by:

$$\begin{bmatrix} \dot{q} \\ \dot{K} \end{bmatrix} = \begin{bmatrix} r & -F_{KK} \\ \frac{K^*}{2b} & 0 \end{bmatrix} \begin{bmatrix} q - q^* \\ K - K^* \end{bmatrix} \tag{11}$$

As demonstrated above, this second-order dynamic system has two eigenvalues, given by equation 7. Hence the linearised system has two real-valued eigenvalues given by $\lambda_1 > r > 0 > \lambda_2$ thereby exhibiting the property of saddle-path instability.

Solutions to the investment sub-model starting from a range of initial conditions can be used to derive a phase diagram for the dynamics of the investment sub-model. Phase diagrams for the true (nonlinear) and linearised investment sub-models are given in Figures 1 and 2 respectively. The same set of initial conditions are used in both cases, so the isoclines in the two figures are directly comparable.

The figures show saddle-path dynamics of the investment sub-model. They also show that there are substantial differences in the dynamics between the nonlinear and linearised sub-models.

The dynamics of the investment sub-model can be further considered by examining the sub-model's response to an exogenous interest rate shock. Assume the model is at steady-state and then is shocked by an interest rate increase from 3% to 5%. That is the parameters are those given above, but the model is instantaneously shocked from $r = 0.03$ to $r = 0.05$. As a consequence of the saddle-path property, for the investment sub-model K must initially remain at the old steady-state value, but q must jump so that the sub-model is on the stable arm as the sub-model evolves to the new steady-state. The new steady-state is that shown in Figures 1 and 2.

The numeric problem is to find the initial conditions for q, given that both the initial condition for K and the terminal conditions for both variables are known. This is a particular case of the two-point boundary value problem (Burden and Faires, 1985).

Solving the investment sub-model is then equivalent to solving the following problem.

Find $q(t_0)$ subject to:

$$\dot{q}(t) = f(q(t), K(t)) \tag{12}$$

$$\dot{K}(t) = g(q(t), K(t)) \tag{13}$$

$$K(t_0) = K_0 \tag{14}$$

$$q(t_f) = q^* + \epsilon_q \tag{15}$$

$$K(t_f) = K^* + \epsilon_K \tag{16}$$

$$t \in [t_0, t_f] \tag{17}$$

where t_0 is the initial time, t_f is some (exogenously given) large number representing the terminal point for time and ϵ_q and ϵ_K are small error terms that are 'close enough' to zero.

3.2 *Solving the model using reverse shooting*

This problem can be solved using a solution approach, which is referred to in this paper as reverse shooting. The aim of this approach is to find the stable trajectories of the model and generate the stable arms in (q, K) phase space. This approach makes use of the feature that time can be abstracted from the solution of the model. The stable arms forwards in time will become the unstable arms with time going backwards. The same will apply for the unstable arms, with reverse time making them the stable arms. This approach finds the forward-stable arms by finding the unstable arms in reverse time (backward-unstable arms). This motivates the word reverse in the name for the approach.

The approach also makes use of the separatrix property of saddles (Khalil, 1996). The stable trajectories from a saddle form a separatrix so that the phase plane of the model is divided into four separate regions. Solutions always remain in one and only one region. Choosing a solution close to the boundary of one of these regions will ensure that the solution will remain close to the boundary. Choosing a backward-unstable solution close to the boundary will provide a time-path for the forward-stable solution (stable arm).

Using this property and the fact that any solution that is close to the steady-state equilibrium is close to all four boundaries, linearisation can be helpful in the generation of the stable trajectories for a nonlinear model. From the linearisation of the investment sub-model at the steady-state, the eigenvalues are such that $\lambda_1 > 0 > \lambda_2$. The corresponding eigenvectors are denoted by $\mathbf{v}(\lambda_1)$ and $\mathbf{v}(\lambda_2)$. The forward-stable trajectories of the nonlinear model will be tangent to the forward-stable eigenvector, $\mathbf{v}(\lambda_2)$, as the trajectories approach the steady-state. Similarly, the forward-unstable trajectories will be tangent to the forward-unstable eigenvector, $\mathbf{v}(\lambda_1)$, as they approach the steady-state. These properties allow an approach for finding the forward-stable arms of the investment sub-model by using reverse time and choosing initial conditions so that ϵ_q and ϵ_K are close to zero and tangent to the forward-stable eigenvector.

Figure 3 shows a stable arm for the linearised and the true (nonlinear) model. The figure shows the dynamics of the model in response to the interest rate shock. These stable arms have been derived using the reverse shooting approach. Once the stable arm (or forward-stable trajectory) has been determined in this manner, the initial value for q can be obtained by reading the corresponding value of $q(t_0)$ along the stable arm for the initial condition $K(t_0)$.

3.3 *Solving the model using forward shooting*

Another approach is to use forward shooting (Burden and Faires, 1985; Judd, 1998). The general approach with forward shooting is to guess the unknown initial condition, solve the model as an initial value problem and see if the terminal conditions to the initial value problem are close enough to the steady-state equilibrium.

To solve the investment sub-model, $K(t_0)$ is given and the shooting approach uses an initial guess for the initial condition of $q(t_0)$. This turns the problem into an initial value problem, which will generate terminal values, $q(t_f)$ and $K(t_f)$. The aim of the shooting approach is to find the particular $q(t_0)$ such that $q(t_f)$ and $K(t_f)$ are 'close enough' to q^* and K^*.

An advantage of this approach is that the initial time can be determined as the model is

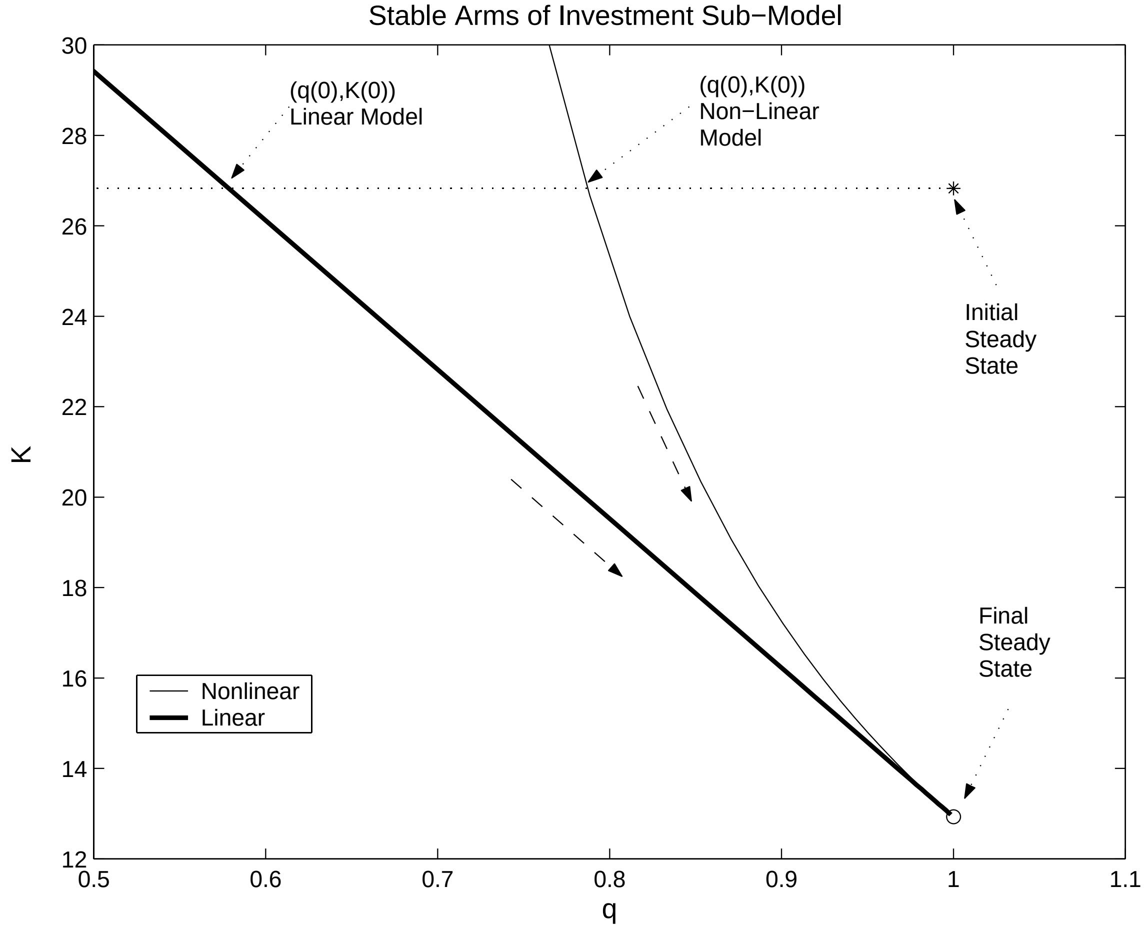

Figure 3. Stable Arms of the Investment Sub-Model.

autonomous. Thus t_0 can be set to zero.

In terms of Figure 3, $K(0)$ is kept constant at the pre-shock (initial) steady-state value and a search is made along the line of constant K (as indicated in the Figure by the dotted line) to find the q value that intersects with the stable arm. This gives $q(0)$, and the size of the jump in q due to the shock.

Numerically, the approach is implemented as a search for the iterate, $q(0)^{(n)}$, which minimises the objective function:

$$J(q(0)^{(n)}) = ||\mathbf{x_I}^* - \mathbf{x_I}(t_f)^{(n)}||_2 \tag{18}$$

Here the subscript, (n), refers to the nth iterate. The terminal values of the investment sub-model for the nth iterate, $\mathbf{x_I}(t_f)^{(n)}$, is generated from the initial value problem:

$$\dot{\mathbf{x}}_{\mathbf{I}}(t) = \mathbf{f}_{\mathbf{I}}(\mathbf{x}_{\mathbf{I}}(t)) \tag{19}$$

$$\mathbf{x}_{\mathbf{I}}(0) = [q(0)^{(n)}, K(0)]^T \tag{20}$$

$$t \in [0, t_f] \tag{21}$$

A limitation with this approach is that it is necessary to generate multiple solutions to the underlying differential equation. This is in stark contrast to the reverse shooting approach, which requires only one solution of the differential equation to provide a definitive solution for each stable arm.

Another problem with the forward shooting approach is that the sub-model solution $(q(t_f), K(t_f))$ is very sensitive to the initial conditions (and hence $q(0)$). This is the main concern of the issue of saddle-path instability. It is obvious from the phase diagrams (Figures 1 and 2) that finding the set of initial conditions which places the sub-model on the forward-stable arm is difficult.

3.4 *Numeric comparison of the approaches*

The two approaches were implemented in MATLAB. The comparison uses the scenario of an exogenous shock in interest rates from 3% to 5% as considered above, and uses the nonlinear model. The time horizon is 150 time-units ($t_f = 150$), which is adequate to reach the steady-state with forward shooting.

At the initial (pre-shock) steady-state is given by $(q, K) = (1.0000, 26.8270)$. The final (post-shock) steady-state is given by $(q^*, K^*) = (1.0000, 12.9314)$, Thus $K(0) = 26.8270$ and the aim is to find $q(0)$.

The reverse shooting approach finds $q(t_0) = 0.7872$ using a single solution to an initial value problem for the ordinary differential equation, and a single use of an interpolation function. The initial condition for K is found at time 18.55 ($t_0 = 18.55$). The computational effort is 7525 floating-point operations. The MATLAB *ode45* function is used as the differential equation solver; *interp1* function used to implement a cubic

Initial Value of q and Computational Effort for Different Solution Approaches. Investment Sub-Model.		
Solution Approach	Initial q	Computational Effort
Nonlinear Model		
Reverse Shooting	0.7872	7.5×10^3
Forward Shooting	0.7872	5.4×10^6
Linearised Model		
Reverse Shooting	0.5785	6.8×10^3
Forward Shooting	0.5785	5.3×10^5
Analytic	0.5785	3.2×10^3

Table 1. Comparison of Approaches for the Investment Sub-Model. Computational effort is measured by number of floating-point operations required to solve the model. The analytic solution is included as a benchmark. The computational effort for the analytic solution is the effort to generate a set of equally spaced points (with a step size of 1.25) over the time horizon (0,150). It is for the full model rather than the investment sub-model.

spline to interpolate for $q(t_0)$ given $K(t_0)$; and, *flops* function to calculate the number of floating-point operations.

For the forward shooting approach, the initial guess for $q(0)$ is 1. This is the pre-shock steady-state value. It is also the (post-shock) steady-state value, q^*. A Nelder-Mead simplex direct search search algorithm was used to minimise the objective function of equation 18. The MATLAB *fminsearch* function from the *Optimization Toolbox* (Branch and Grace, 1996) was used to implement the search.

To find $q(0)$ to the same level of accuracy as in the reverse shooting approach took 84 solutions to the ordinary differential equation, and 5423467 floating-point operations with forward shooting. This is over 720 times more computational effort than it took to solve the problem using reverse shooting.

A comparison of the solution approaches can be seen in Table 1. The table shows that the reverse shooting approach requires less computational effort for both the linear and nonlinear sub-models and that the nonlinear model requires more computational effort to solve than its linearised counterpart.

The table also shows that the linear model gives significantly different estimates of the jump in q following the interest rate shock when compared with the true (nonlinear) model. Of course the solution derived from the true model is an estimator of the true value of q following an initial jump. This indicates that the solutions derived from the linearised model are generally unreliable.

4. SOLVING THE FULL LINEARISED MODEL

The linearised model can be solved analytically. Indeed this is the usual way the model is solved. Here we use the analytic solution as a comparison to the numeric approaches for the linear model. The accuracy of the approaches can be obtained by comparing the numeric and analytic solutions for the linearised model.

The linearised model is given by equation 6. The eigenvalues of the driving matrix, A, are given by equations 7 to 9. For each eigenvalue, λ_i, the corresponding eigenvector is given by:

$$\mathbf{v}(\lambda_i) = \begin{bmatrix} \Delta(\lambda_i) \\ \Delta(\lambda_i)\frac{K^*}{2b\lambda_i} \\ (r-\lambda_i)p(p+\theta)K^* \\ \Delta(\lambda_i)\frac{K^*}{2b\lambda_i} - p(p+\theta)K^* \end{bmatrix} \tag{22}$$

where

$$\Delta(\lambda_i) = (r-\lambda_i)(r-\theta-\lambda_i) - p(p+\theta)$$

and, in particular, $\Delta(\lambda_3) = \Delta(\lambda_4) = 0$.

Hence, the general solution to the linear model (equation 6) is given by:

$$\mathbf{x}(t) - \mathbf{x}^* = V(\lambda)\mathcal{A}(\lambda, t) \tag{23}$$

where

$$\mathbf{x}(t) - \mathbf{x}^* = \begin{bmatrix} q(t) - q^* \\ K(t) - K^* \\ C(t) - C^* \\ D(t) - D^* \end{bmatrix}$$

$$V(\lambda) = [\,\mathbf{v}(\lambda_1) \quad \mathbf{v}(\lambda_2) \quad \mathbf{v}(\lambda_3) \quad \mathbf{v}(\lambda_1)\,]$$

$$\mathcal{A}(\lambda, t) = \begin{bmatrix} A_1 e^{\lambda_1 t} \\ A_2 e^{\lambda_2 t} \\ A_3 e^{\lambda_3 t} \\ A_4 e^{\lambda_4 t} \end{bmatrix}$$

Since λ_1 and λ_3 are real-valued positive eigenvalues, application of the appropriate transervality condition allows us to set $A_1 = A_3 = 0$, so that equation 23 reduces to the following:

$$\begin{bmatrix} q(t) - q^* \\ K(t) - K^* \\ C(t) - C^* \\ D(t) - D^* \end{bmatrix} = [\,\mathbf{v}(\lambda_2) \quad \mathbf{v}(\lambda_4)\,] \begin{bmatrix} A_2 e^{\lambda_2 t} \\ A_4 e^{\lambda_4 t} \end{bmatrix} \tag{24}$$

Equation 24 can be rewritten as:

$$\begin{bmatrix} q(t) - q^* \\ K(t) - K^* \\ C(t) - C^* \\ D(t) - D^* \end{bmatrix} = \begin{bmatrix} \Delta(\lambda_2) & 0 \\ \Delta(\lambda_2)\frac{K^*}{2b\lambda_2} & 0 \\ (r - \lambda_2)p(p+\theta)K^* & p+\theta \\ \Delta(\lambda_2)\frac{K^*}{2b\lambda_2} - p(p+\theta)K^* & -1 \end{bmatrix} \begin{bmatrix} A_2 e^{\lambda_2 t} \\ A_4' e^{\lambda_4 t} \end{bmatrix} \tag{25}$$

All that is need to solve the model is to use the initial conditions for K and D to determine the values of A_2 and A'_4. If at $t = 0$, $K = K_0$ and $D = D_0$ then

$$A_2 = \Delta(\lambda_2)\frac{K^*}{2b\lambda_2}(K_0 - K^*) \tag{26}$$

$$A'_4 = (\Delta(\lambda_2)\frac{K^*}{2b\lambda_2} - p(p+\theta)K^*)A_2 - (D_0 - D^*) \tag{27}$$

For solving the model the same scenario as above is used. That is the model is in steady-state with interest rates at 3%, and then is shocked by interest rates rising to 5%. In this case $r_0 = 0.03$, $K_0 = 26.8269$ and $D_0 = 30.3093$. The initial value of q is $q(0) = 0.5785$. (This is the same as the linear investment sub-model as can be seen from the Figure 3.) The initial value of C is $C(0) = 0.4046$. The constant $A_2 = 524.3516$, and the constant $A'_4 = -36.7677$.

5. SOLVING THE FULL NONLINEAR MODEL

Unlike the linearised model, the full nonlinear model cannot be solved analytically. Any solution approach must resort to numerical techniques.

Unfortunately, the reverse shooting approach used to solve the investment sub-model cannot be employed directly to derive a solution to the full model.

Under another approach, which makes use of the reverse shooting solution to the investment sub-model, the block recursive structure of the model may be exploited to solve the full model. This is achieved by generating the solution for the investment sub-model and then treating the solution to the investment sub-model as exogenous in the solution of the C-D component of the full model. The consumption sub-model can then be solved separately from the investment sub-model. This other approach is described below as modified reverse shooting.

The forward shooting approach can also be extended. These two alternative approaches to extending the forward shooting approach are described below as double forward shooting and as full forward shooting.

The consumption sub-model is given by equations 3 and 4. The sub-model is linear (in C and D) and may be written in state space form as:

$$\dot{\mathbf{x}}_{\mathbf{C}}(t) = A\mathbf{x}_{\mathbf{C}}(t) + \mathbf{b}(t) \tag{28}$$

with the states:

$$\mathbf{x}_{\mathbf{C}}(t) = \begin{bmatrix} C(t) - C^* \\ D(t) - D^* \end{bmatrix}$$

driving matrix:

$$A = \begin{bmatrix} r - \theta & p(p+\theta) \\ 1 & r \end{bmatrix}$$

and forcing function from the investment sub-model:

$$\mathbf{b}(t) = \begin{bmatrix} -p(p+\theta)[\tilde{q}(t)\tilde{K}(t) - q^*K^*] \\ -a[\tilde{K}(t)]^\alpha + \tilde{K}(t)\Lambda(\tilde{q}(t)) + a[K^*]^\alpha \end{bmatrix}$$

In the forcing function $\tilde{K}(t)$ and $\tilde{q}(t)$ are the solutions derived from the investment sub-model. Here we generate these solutions through the reverse shooting approach due to its superior computational performance.

The solution this state model (equation 28) is:

$$\mathbf{x}_{\mathbf{C}}(t) = Pe^{\Phi t}\left(\begin{bmatrix} 0 \\ A_5 \end{bmatrix} - \int_t^\infty e^{-\Phi\tau}P^{-1}\mathbf{b}(\tau)d\tau\right) \tag{29}$$

where

$$\Phi = P^{-1}AP = \begin{bmatrix} \lambda_3 & 0 \\ 0 & \lambda_4 \end{bmatrix}$$

$$e^{\Phi t} = \begin{bmatrix} e^{\lambda_3 t} & 0 \\ 0 & e^{\lambda_4 t} \end{bmatrix}$$

$$P = \begin{bmatrix} p & -\theta - p \\ 1 & 1 \end{bmatrix}$$

Equation 29 gives the solution for the consumption sub-model. It requires that an integral be calculated, with the integral terms given by:

$$\begin{bmatrix} Z_C(t) \\ Z_D(t) \end{bmatrix} = Pe^{\Phi t} \int_t^\infty e^{-\Phi\tau} P^{-1} \mathbf{b}(\tau) d\tau \tag{30}$$

Both the initial jump in consumption and the stable manifolds for the consumption sub-model (and hence the full model) can be generated from equation 29.

Firstly, consider the initial jump in consumption. If initially (at $t = t_0$) the overseas debt (D) is given by D_0, then from equation 29 the initial consumption is given by:

$$C_0 = Z_C(t_0) + C^* - (\theta + p)(D_0 - D^* - Z_D(t_0)) \tag{31}$$

To solve the dynamics of the stable manifold for the model, then it is necessary that the constant, A_5, be chosen to be consistent with the initial jump in consumption. If reverse shooting is used to solve the investment sub-model and the appropriate jump in q was found to occur at time t_0, then, from equation 29, the constant A_5 is given by:

$$A_5 = e^{-\lambda_4 t_0}(D_0 - D^* - Z_D(t_0)) \tag{32}$$

Equation 29 can then be solved to obtain the stable manifold for C and D, and hence the entire model.

Nonlinearities only enter the consumption sub-model through the forcing function $\mathbf{b}(t)$. To solve the full linear model the linear investment sub-model be used to generate this forcing function. In this case $\mathbf{b}(t)$ is replaced by:

$$\mathbf{b_L}(t) = \begin{bmatrix} -p(\theta + p)[\tilde{K}_L(t) - K^*] - p(\theta + p)K^*(\tilde{q}_L(t) - q^*) \\ -r[\tilde{K}_L(t) - K^*] + \frac{K^*}{2b}(\tilde{q}_L(t) - q^*) \end{bmatrix} \tag{33}$$

Figure 4 plots in $C - D$ space the stable manifold for the full model. In this plot q and K are not constant but remain on the stable manifold[2]. The figure uses the same change

[2]Thus the nonlinearities in the plot.

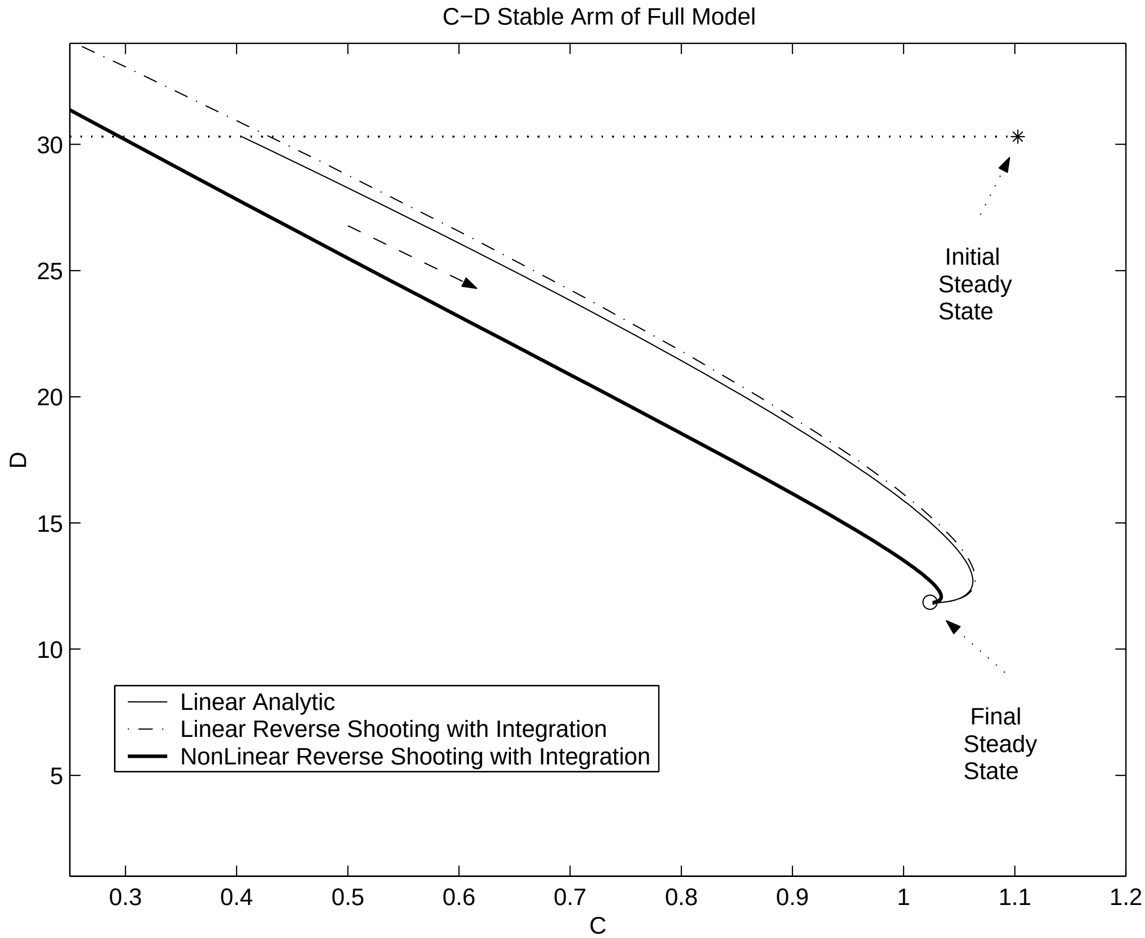

Figure 4. C-D Space Stable Arms of the Model.

in interest rate scenario as considered above. It contains the plot for the true (nonlinear) model and the linear model using this modified reverse shooting approach. The figure shows that despite the consumption sub-model being linear (in C and D), there are substantial differences between the true (nonlinear) model and its linear counterpart.

Figure 4 also plots the results from the analytic solution to the linear model (equation 25). It shows that the results generated through the modified reverse shooting approach for the linear model are similar to those generated by the analytic solution.

The differences between the numeric modified reverse shooting approach and the analytic solution of the linearised model give an indication of the computational accuracy of the numeric approach. Errors enter the numeric approach primarily through the truncation errors generated by the numeric integration method. The modified reverse shooting approach implemented here calculated initial consumption for the linear model of $C(t_0) =$

0.4302. This compares to the analytic solution of 0.4046. The relative error between the two solutions is 0.06 indicating that the approach has reasonable accuracy.

For the true (nonlinear) model the initial consumption is $C(t_0) = 0.2943$. The jump in consumption occurs at $t = 18.54$ (ie. $D(t) = D_0$ at $t = 18.54$ giving $t_0 = 18.54$). This is very close to the expected 18.55, which is the time of the jump ($\tilde{K} = K_0$) in the investment sub-model.

The jump in consumption from the true (nonlinear) model is substantially different from that calculated through the linear model. This is despite the fact that the consumption sub-model is the same in C and D in both cases and occurs as a direct consequence of nonlinearities in the investment sub-model.

For computing the stable manifold, the approach was implemented with the reverse shooting in the investment sub-model being the same as above in Section 3.2. While the differential equation solver used a variable step-size, results were obtained for fixed time intervals, with the time interval being the smallest time step generated by the solver in solving the investment sub-model. This increases the computational effort involved in solving the differential equation, but as results are only known at grid points and the numeric integration (quadrature) must be based on these grid points, it provides a fine enough grid for the integration. Integration was implemented using the Trapezoidal rule based on the fixed grid intervals.

Numerically solving equation 29 requires an integration at each time step. For a time horizon of 150 the computation effort required was 90856 flops for the true (nonlinear) model. This includes solving the investment sub-model.

5.2 *Double forward shooting*

The consumption sub-model (equation 28) can also be solved by other means. One approach is to use forward shooting to find the initial jump in consumption. In this case the problem is analogous to that of forward shooting in the investment sub-model as given in Section 3.3. That is, search for the initial consumption, $C(0)$, that minimises the error between the terminal solution to the initial value ordinary differential equation

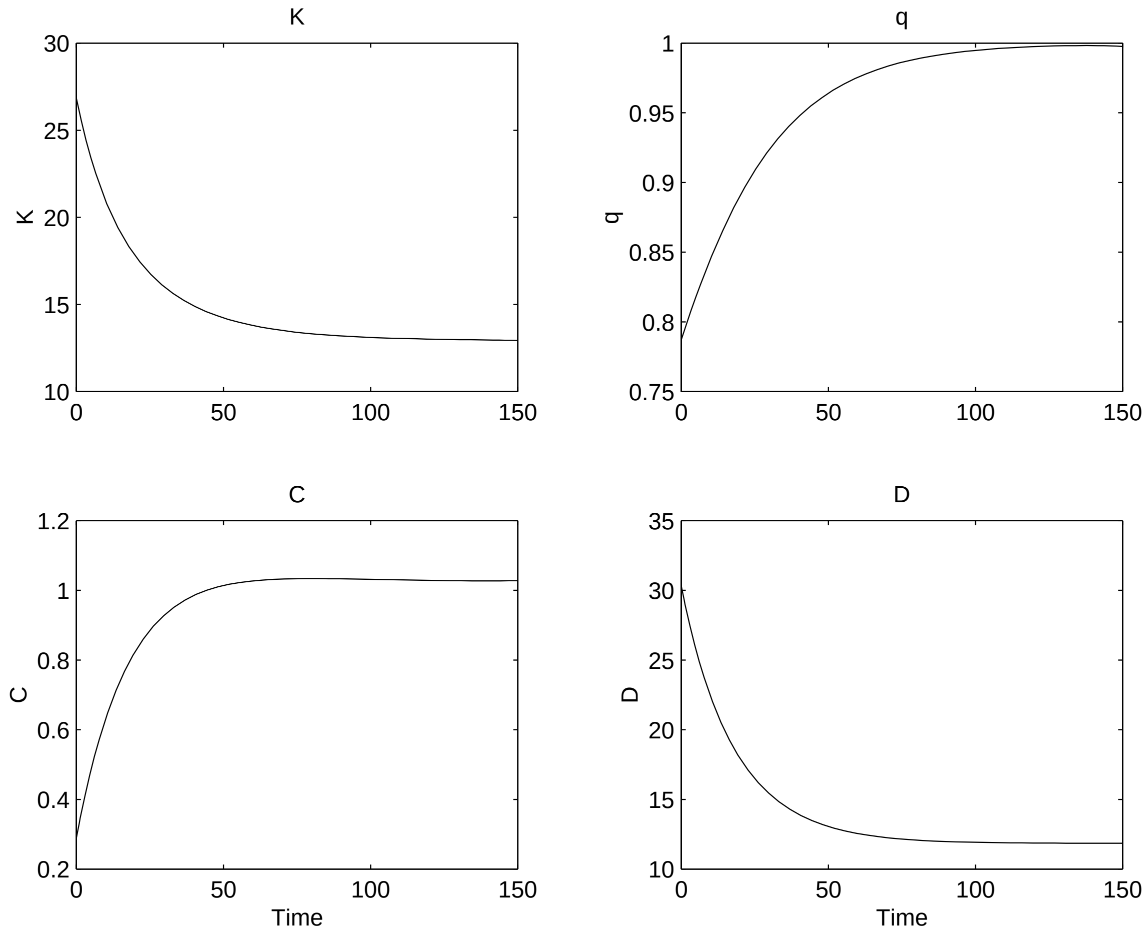

Figure 5. Trajectories of States after Exogenous Shock to Interest Rates: Double Forward Shooting with Nonlinear Model.

problem and the steady-state solution. This can be implemented in an analogous manner to equations 18 and 21, but with the model given by equation 28.

Forward shooting for the consumption sub-model also suffers from the same limitations as forward shooting for the investment sub-model. One limitation is that, because the consumption sub-model has the saddle-path instability property, the terminal solution to the model will be highly sensitive to the choice of initial consumption.

The solution of the consumption sub-model (equation 28) requires a solution to the investment sub-model for the forcing function, $\mathbf{b}(t)$. Implementing the forward shooting approach in the investment sub-model and using this solution in a forward shooting approach for the consumption sub-model leads to the approach we refer to as double forward shooting.

We implement the second (consumption) phase of the double forward shooting approach using the same techniques as for the investment sub-model. That is, a simplex search algorithm (*fminsearch*) to find the initial consumption that minimises the 2-norm between the terminal solution and the steady-state solution. Again a variable step-size solver (*ode45*) was used to solve the initial value problem. One issue is that the solution to the investment sub-model is only known at (time) grid points. Again interpolation using a cubic spline was used to determine solutions between grid points.

Implementing the double forward shooting approach generates the results presented in Figure 5. The figure plots the trajectory of each state with same exogenous shock to interest rates scenario considered above. The trajectories are for the true (nonlinear) model.

For this model the initial consumption is given by $C(0) = 0.2898$. This compares to the value of 0.2943 found by the modified reverse shooting approach. For the linear model, $C(0) = 0.4046$ which is the same as that given by the analytic solution. This latter result indicates that the double forward shooting approach is more accurate than the modified reverse shooting approach.

Double forward shooting requires considerable computational effort. For the nonlinear model 80 solutions of the differential equation were necessary to solve the investment sub-model and a further 74 where necessary to solve the consumption sub-model. In total 31324796 floating point operations where needed to solve the model. This is 345 times more flops than was necessary to solve the model using the modified reverse shooting approach.

5.3 *Full forward shooting*

The full model can also be solved by the forward shooting approach. In this situation, the problem becomes that of finding the initial jumps in both q and C simultaneously. The approach compares to double forward shooting where the model structure was exploited to find the jump in q and then the jump in C.

Numerically, the full forward shooting approach is implemented by finding the iterate,

$(q(0)^{(n)}, C(0)^{(n)})$, which minimises the objective function:

$$J(q(0)^{(n)}, C(0)^{(n)}) = ||\mathbf{x}^* - \mathbf{x}(t_f)^{(n)}||_2 \tag{34}$$

Where the subscript, (n), refers to the nth iterate. The terminal value of the model for the nth iterate, $\mathbf{x}(t_f)^{(n)}$, is generated from the initial value problem:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t)) \tag{35}$$

$$\mathbf{x}(0) = [q(0)^{(n)}, K(0), C(0)^{(n)}, D(0)]^T \tag{36}$$

$$t \in [0, t_f] \tag{37}$$

This numeric problem is directly analogous to the forward shooting approaches considered above.

For the true (nonlinear) model, the full forward shooting approach when implemented in the same manner as above on the exogenous shock to interest rates example, produces solutions that are (to an accuracy of 10^{-3}) the same as that from the modified reverse shooting approach. That is, the model solutions in Figures 4 and 5. The value of q is 0.7873, and the value of consumption is 0.2899.

To obtain the model solutions to the exogenous shock required 400 solutions to the differential equation. The solution used 26018550 floating point operations.

5.4 *Numeric comparison of the approaches*

A comparison of the approaches for solving the true (nonlinear) model to the interest rate shock is given in Table 2 (on page 29). To two decimal places all three approaches give the same value of the jumps. But the computational effort in solving the model differs significantly between the approaches. The modified reverse shooting approach requires the least number of floating point operations making it the most computationally efficient.

The full forward shooting approach requires less computational effort than the double shooting approach. This is despite the fact that it requires more than twice the number of solutions to a larger differential equation. It may result from the extra interpolation step required in solving the consumption sub-model in the double forward shooting approach.

Table 2 also compares the approached for the linearised model. As with the true (nonlinear) model the modified reverse shooting approach requires less computational effort than either of the forward shooting approaches. Again there is a computational disadvantage in implementing a double forward shooting approach compared to the full forward shooting approach.

The table also presents the computational effort required to generate a set of points giving the time-path for each endogenous variable following the shock from the analytic solution (equation 24). This can be used as a benchmark for a comparison of the numeric approaches. The table shows that the modified reverse shooting approach is closer to this benchmark than the forward shooting approaches.

For the linear model the initial consumption calculated by the double shooting approach and the full forward shooting approach are the same as that calculated using the analytic solution, indicating that they are more accurate then the modified reverse shooting approach.

Table 2 also shows that for exogenous shock to interest rates considered in this paper, the linearised model does not produce accurate results for the jumps in q and C.

6. CONCLUSION

In this paper we have considered solutions to a well known representative agent model. This model has a number of properties in common with a wide range of economic models. Firstly, the model is nonlinear so that numerical solutions are necessary. Secondly, the model solution must lie on a stable manifold, and numerically finding this manifold is the crux of the problem. Solutions will easily ‘fall off’ the stable manifold so that the solution will be highly sensitive to initial conditions and to round-off and truncation errors introduced by the solution algorithms. Thirdly, the model has variables that jump after a shock to ensure the solution remains on a stable manifold.

The paper has considered approaches to generating the solution of the model when it has been subjected to an exogenous shock.

In the paper it has been shown that the common practice of linearising the model and generating linear solutions will lead to results that are substantially different from those generated by using the true (nonlinear) model.

The paper has also compared a number of solution approaches. The task of computing the model solution can be converted to a two-point boundary value problem and to solve this problem the paper has exploited shooting techniques. Forwards in time shooting was used to solve the model. But the difficulty for forward shooting is that the model solution is highly sensitive to initial conditions and numeric errors. The paper has shown that a considerable computational effort is necessary to solve the model using this approach.

The computational effort of forward shooting is not reduced by exploiting the block recursive structure of the model and implementing forward shooting on each block. The paper has shown that this double forward shooting approach requires even more computational effort than solving the model as a single forward shooting problem.

The most computationally efficient approach is to fully exploit the model structure. The model has two blocks or sub-models, and each sub-model has the saddle-path instability property. Exploiting the saddle-path instability property on the first (investment) sub-model by using reverse (in time) shooting and the seperatrix property of saddles generates the stable arm in a single solution of the sub-model. Then, by exploiting the linearity of the second block (consumption sub-model), through the use of linear techniques a more computationally efficient solution approach is obtained. The paper has shown that this modified reverse shooting approach generates a superior solution methodology for the model.

The modelling framework chosen here has much in common with many large-scale macro models. Accordingly, computational savings, like those generated in this paper, are likely to be even more significant for larger macro models. Such computational savings are also likely to be important if the model needs to be solved many times. This is going to be the case where robustness over a large number of parameter values is being investigated or where a large number of graphical solutions need to be examined.

7. ACKNOWLEDGMENTS

Peter Stemp is grateful to the Australian Research Council (ARC) and the Faculty of Economics and Commerce at the University of Melbourne for financial support through an ARC Small Grant and a Faculty Research Grant.

8. REFERENCES

Anderson, Gary and George Moore (1985). A linear algebraic procedure for solving linear perfect foresight models. *Economics Letters* **17**, 247–252.

Blanchard, Oliver Jean and Charles M. Kahn (1980). The solution of linear difference models under rational expectations. *Econometrica* **48**(5), 1305–1311.

Boucekkine, Raouf (1995). An alternative methodology for solving nonlinear forward-looking models. *Journal of Economic Dynamics and Control* **19**, 711–734.

Branch, Mary Ann and Andrew Grace (1996). *MATLAB: Optimization Toolbox*. The Mathworks Inc. Natick, Massachusetts, USA.

Burden, Richard L. and J. Douglas Faires (1985). *Numerical Analysis*. fifth ed.. PWS Publishing Company. Boston, USA.

Fair, Ray C. and John B. Taylor (1983). Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models. *Econometrica* **51**(4), 1169–1185.

Judd, Kenneth L. (1998). *Numerical Methods in Economics*. The MIT Press. Cambridge, U.S.A.

Khalil, H. K. (1996). *Nonlinear Systems*. 2 ed.. Prentice Hall. Upper Saddle River, U.S.A.

Mathworks (1998). *Using MATLAB Version 5*. The Mathworks Inc. Natick, Massachusetts, USA.

Matsuyama, Kiminori (1987). Current account dynamics in a finite horizon model. *Journal of International Economics* **23**, 299–313.

Zadrozny, Peter A. (1998). An eigenvalue method of undetermined coefficients for solving linear rational expectations models. *Journal of Economic Dynamics and Control* **22**(8–9), 1353–1373.

Initial Values of q and C and Computational Effort for Different Solution Approaches. Full Model.			
Solution Approach	Initial q	Initial C	Computational Effort
Nonlinear			
MRS	0.7872	0.2943	9.1×10^4
DFS	0.7872	0.2898	3.1×10^7
FFS	0.7873	0.2899	2.6×10^7
Linearised			
MRS	0.5785	0.4302	9.1×10^4
DFS	0.5785	0.4046	2.5×10^7
FFS	0.5785	0.4046	3.7×10^6
Anl	0.5785	0.4046	3.2×10^3

Table 2. Comparison of Approaches for the Nonlinear and Linearised Models. Computational effort is measured by number of floating-point operations required to solve the model. MRS is the Modified Reverse Shooting approach; DFS is the Double Forward Shooting approach; FFS is the Full Forward Shooting approach, and Anl is the analytic solution. The analytic solution is included as a benchmark. The computational effort in the analytic solution is the effort is that required to generate points over the time horizon for equation 24.