

Solving a stochastic growth model using the Stacked-Newton method

Richard G. Pierse

University of Surrey, Guildford GU2 7XH, U.K.

December 1 2002

Abstract

In this paper we consider the solution of a simple stochastic growth model using stacked-Newton methods. The method is compared with a variety of other solution methods, all implemented in WinSolve, a general nonlinear model solution program.

1 Introduction

In 1990, ten authors were invited to solve a simple stochastic growth model proposed by Christopher Sims, using a variety of different numerical methods. Taylor and Uhlig (1990) presents a comparison of the various proposed methods and solutions without making any very clear recommendations for choosing between them. Comparison of the methods was complicated by the fact that authors had each used different computer software and hardware, and different random number generators to draw their stochastic shocks. In this paper, we revisit the stochastic growth model and suggest a new solution method based on the perfect foresight assumption, a stacked version of Newton's method implementing suggestions of Laffargue (1990), Boucekine (1995) and Juillard (1996). This so-called L-B-J algorithm is compared with other methods, focusing on the extended path method of Fair and Taylor (1983), implemented for the stochastic growth model by Gagnon (1990), and the parameterised expectations method of den Haan and Marcet (1990) and (1994). All these methods are implemented in *WinSolve*, a user-friendly computer package for solving general nonlinear models, described in Pierse (2002) and available for download from the internet.

The structure of the paper is as follows: Section 2 describes the stochastic growth model and different approaches to solving it, Section 3 discusses

the stacked-Newton method as applied to this model. Section 4 compares this method to the extended path and parameterised expectations methods. Finally, some conclusions are drawn.

2 The stochastic growth model

In this model agents are assumed to be infinitely lived and to maximise lifetime expected utility subject to a budget constraint. A constant relative risk aversion utility function

$$u(C_t) = (1 - \tau)^{-1} C_t^{1-\tau}$$

is assumed where C_t is consumption and τ is the coefficient of relative risk aversion $0 < \tau < 1$. Then, formally, agents solve the following problem:

$$\max E_0 \sum_{t=0}^{\infty} \beta^t (1 - \tau)^{-1} C_t^{1-\tau} \quad (1)$$

subject to the resource constraint

$$C_t + K_t = \theta_t K_{t-1} \alpha \quad (2)$$

where K_t is the end of period capital stock, and θ_t is technology. $1 - \mu$ is the rate of capital depreciation, $0 \leq \mu \leq 1$ and β is the rate of time discount, $0 < \beta < 1$.¹ We also impose the side-conditions that $C_t > 0$ and $K_t > 0$, for all t . Technology θ_t is assumed to be stochastic, following the autoregressive process

$$\ln \theta_t = \rho \ln \theta_{t-1} + \varepsilon_t \quad (3)$$

where ε_t is a serially uncorrelated normally distributed random variable with zero mean and constant variance σ^2 .

The first order Euler condition for capital in this model is given by

$$C_t^{-\tau} = E_t[\beta C_{t+1}^{-\tau} (\mu + \alpha \theta_{t+1} K_t^{\alpha-1})] \quad (4)$$

The solution to this model is a decision rule for consumption and one for capital stock given by $C_t = f(K_{t-1}, \theta_t)$ and $K_t = g(K_{t-1}, \theta_t)$ respectively. In general the exact forms of functions $f(\cdot)$ and $g(\cdot)$ are not known and solutions must be found by numerical solution of the equations (2), (3) and (4) over

¹To be precise, the original model proposed by Sims assumed no capital depreciation so that $\mu = 1$. However, allowing some depreciation does not materially complicate the model.

a finite time horizon $t = 1, \dots, T$. However, for the special case where the utility function is logarithmic ($\tau = 1$) and there is full depreciation ($\mu = 0$), there is a simple closed-form solution (see for example Sargent (1987) p122), which is given by

$$K_t = \alpha\beta K_{t-1}^a \theta_t$$

and

$$C_t = (1 - \alpha\beta)K_{t-1}\alpha\theta_t.$$

This special case of the model is described in Brock and Mirman (1972) and is known as the *Brock-Mirman* economy.

An analytic expression for the long-run steady state of the full model can be evaluated by setting $\varepsilon_t = 0$, $\theta_{t-1} = \theta_{t+1} = \theta_t$, $K_{t-1} = K_t$, $E_t C_{t+1} = C_t$, and solving equations (2), (3) and (4). The solution is:

$$\begin{aligned} \theta^* &= 1 \\ K^* &= \left(\frac{\alpha\beta}{1 - \beta\mu}\right)^{1/(1-\alpha)} \\ C^* &= \left(\frac{\alpha\beta}{1 - \beta\mu}\right)^{\alpha/(1-\alpha)} + (\mu - 1)\left(\frac{\alpha\beta}{1 - \beta\mu}\right)^{1/(1-\alpha)}. \end{aligned} \tag{5}$$

The basic problem in solving the stochastic growth model is in evaluating the nonlinear expectation in (4) and solution methods can be broadly divided into three approaches according to how they do this. In the first approach, the continuous shocks from the original problem are discretised so that the expectation can be evaluated numerically and the resulting discrete problem solved over a grid of points. By refining the grid, an arbitrary degree of accuracy can be achieved. Tauchen (1990) applies this approach to the stochastic growth model. A second approach involves approximation of the original problem by a simpler problem for which a closed form solution is readily available. The most obvious example of this approach involves the (log-)linearisation of the model. However, a different example of the same approach is the method of parameterised expectations of den Haan and Marcet (1990) in which the expectation in (4) is explicitly approximated by a known functional form, the parameters of which can be estimated from realisations of the model. A third approach involves replacing the expectation in (4) by a realised future value. This is equivalent to an assumption of perfect foresight and is sometimes known as the assumption that expectations are model consistent. The extended path method of Fair and Taylor (1983), implemented for the stochastic growth model by Gagnon (1990), is one example of a solution method that adopts this approach. The stacked Newton method proposed in the next section is another such method.

3.1 The L-B-J algorithm

Each iteration of Newton's method involves the solution of a set of nT equations. When either n or T is large, the Jacobian matrix \mathbf{J} will become very large and solution using standard methods, such as the LU decomposition (see for example Judd (1998)), will become very expensive. Even storage of the Jacobian matrix in computer memory will become a problem since, for example with $nT = 5000$, 191 megabytes of memory is required, and with $nT = 10000$, 763 megabytes.

It can be seen from (6) that the structure of the Jacobian matrix is very sparse with many zero elements. One possible approach, suggested by Armstrong *et al.* (1998) is to make use of the general sparse matrix solution methods explained in Duff *et al.* (1986) and implemented by AERE Harwell in the MA28 library (Duff (1977))². These methods take account of the sparsity pattern of a matrix to reduce the required storage by only storing non-zero elements, and to eliminate redundant calculations involving known zero elements.

An alternative approach is the L-B-J algorithm, suggested by Laffargue (1990) and refined by Boucekkine (1995) and Julliard (1996), which explicitly takes account of the special block-band structure of the Jacobian matrix \mathbf{J} to solve the equations efficiently. The method proceeds in two stages. In the first stage, the Jacobian matrix is transformed into an upper-triangular structure, by eliminating the blocks below the diagonal by the recursion

$$\text{subtract } \mathbf{B}_t^j \times \text{rows of block } t - j \text{ from rows of block } t$$

from $j = p^*, p^* - 1, \dots, 1$, where $p^* = \min(p, t - 1)$ and then replacing the block on the diagonal by the identity matrix by

$$\text{premultiply rows of block } t \text{ by the inverse of the diagonal block } J_t^*.$$

This transformation proceeds, period by period from $t = 1$ through to $t = T$. Note that the only blocks that need to be stored are those corresponding to the lead coefficients \mathbf{F}_t^i , $i = 1, \dots, k$. This means that storage is reduced from $nT \times nT$ to $nT \times nk$. This can be reduced further by dropping any columns in \mathbf{F}_t^i , corresponding to variables that never appear with a lead. Note also that the last step in this stage is the solution of an $n \times n$ system of equations in the transformed Jacobian block J_t^* . Since this matrix will itself generally be sparse, the general sparse solution methods of Duff (1977) can be applied to this step.

In the second stage of the procedure, the upper-triangular structure is solved recursively, going backwards in time from period T to period 1.

²The NAG library also includes similar versions of these routines.

3.2 Terminal conditions

One important issue is that of *terminal conditions*. Solution requires values of the variables $\mathbf{y}_{T+1}, \dots, \mathbf{y}_{T+k}$, which are outside the solution period. When these are set to fixed exogenous values, then they are analogous to initial conditions and do not affect the Newton algorithm. However, terminal values are often set according to an equation, either an automatic rule such as constant level $\mathbf{y}_{T+j} = \mathbf{y}_T$, $j = 1, \dots, k$ or constant growth rate $\mathbf{y}_{T+j} = \mathbf{y}_T^{j+1} \mathbf{y}_{T-1}^{-j}$, $j = 1, \dots, k$, or an equilibrium equation deriving for example from a steady state of the model. In general let us denote the terminal condition by the set of equations

$$\bar{\mathbf{f}}(\mathbf{y}_{T+j}) = \mathbf{0} \quad , \quad j = 1, \dots, k.$$

In this case, the Newton equations (7) needs to be supplemented by nk rows of the form

$$\bar{\mathbf{J}}_{T+j}(\mathbf{y}_{T+j}^s - \mathbf{y}_{T+j}^{s-1}) = -\bar{\mathbf{f}}(\mathbf{y}_{T+j}^{s-1}) \quad , \quad j = 1, \dots, k$$

where

$$\bar{\mathbf{J}}_{T+j} = \begin{bmatrix} \mathbf{0}_{n \times (T+j-q-1)n} & \bar{\mathbf{B}}_{T+j}^q & \cdots & \bar{\mathbf{B}}_{T+j}^1 & \mathbf{I}_n & \mathbf{0}_{n \times (k-j)n} \end{bmatrix}$$

is of dimension $n \times n(T+k)$ with

$$\bar{\mathbf{B}}_{T+j}^i = \frac{\partial \bar{\mathbf{f}}}{\partial \mathbf{y}_{T+j-i}'} \quad , \quad i = 1, \dots, q.$$

Note that the terminal condition equations will be predetermined so that only lagged variables will appear in $\bar{\mathbf{f}}$.

It is undesirable that the terminal conditions should play an important part in affecting the solution. In order to assess their impact, it is possible to extend the solution period in

4 Solving the Stochastic Growth Model

Imposing the assumption that expectations are model consistent, the three model equations (3), (2) and (4) can be written as

$$\ln \theta_t = \rho \ln \theta_{t-1} + \varepsilon_t$$

$$C_t + K_t = \theta_t K_{t-1} \alpha + \mu K_{t-1}$$

and

$$C_t^{-\tau} = \beta C_{t+1}^{-\tau} (\mu + \alpha \theta_{t+1} K_t^{\alpha-1}).$$

The first of these equations is predetermined so that θ can be solved recursively, independently of the other two equations. The remaining two equations can be written in different ways. One formulation is given by

$$K_t = \theta_t K_{t-1} \alpha + \mu K_{t-1} - C_t$$

$$C_t = [\beta C_{t+1}^{-\tau} (\mu + \alpha \theta_{t+1} K_t^{\alpha-1})]^{-\frac{1}{\tau}}$$

in which case the equations for C and K need to be solved simultaneously. Terminal values are needed for C_{T+1} and θ_{T+1} .

An alternative representation, used by Gagnon (1990), is given by

$$(\theta_t K_{t-1} \alpha + \mu K_{t-1} - K_t)^{-\tau} = \beta (\theta_{t+1} K_t \alpha + \mu K_t - K_{t+1})^{-\tau} (\mu + \alpha \theta_{t+1} K_t^{\alpha-1})$$

$$C_t = \theta_t K_{t-1} \alpha + \mu K_{t-1} - K_t.$$

In this second formulation, consumption does not appear in the first equation and so is a *post-recursive* variable that can be solved recursively once K is determined. Thus the only variable in the simultaneous block is K . Terminal values are required for K_{T+1} and θ_{T+1} .

5 Parameterised Expectations

The problem in solving the stochastic growth model is in finding the expectation

$$E_t[y_{t+1}] \tag{8}$$

in (4) where

$$y_t = C_t^{-\tau} (\alpha \theta_t K_{t-1}^{\alpha-1} + \mu).$$

This expectation is a function of the state variables $x_t = \{K_{t-1}, \theta_t\}$ but its form is unknown. Note that on the assumption of model consistent expectations,

$$E_t[y_{t+1}] = y_{t+1}.$$

den Haan and Marcet (1990) propose a general method for solving models by approximating expectations such as (8) using a functional form

$$\psi_t(\mathbf{x}_t; \boldsymbol{\delta})$$

where \mathbf{x}_t is a $p \times 1$ vector of state variables and $\boldsymbol{\delta}$ is a $k \times 1$ vector of parameters. These parameters are chosen such as to minimise the sum of squared residuals

$$\min_{\boldsymbol{\delta}} \sum_{t=1}^T (y_{t+1} - \psi_t(\mathbf{x}_t; \boldsymbol{\delta}))^2 .$$

This is simply a nonlinear least squares problem and can be solved using Newton's method by iterating on

$$\boldsymbol{\delta}^s = \boldsymbol{\delta}^{s-1} + (\boldsymbol{\Psi}'_{s-1} \boldsymbol{\Psi}_{s-1})^{-1} \boldsymbol{\Psi}'_{s-1} (y_{s+1} - \boldsymbol{\psi}(\mathbf{x}; \boldsymbol{\delta}^{s-1}))$$

where

$$\boldsymbol{\Psi}_{s-1} = \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\delta}}$$

is the $T \times k$ matrix of derivatives of $\boldsymbol{\psi}$ with respect to the parameters $\boldsymbol{\delta}$ evaluated at iteration $s - 1$.

The functional form of $\boldsymbol{\psi}$ should be chosen so as to be able to approximate the expectation as closely as possible. den Haan and Marcet suggest the class of *power functions*

$$\exp P_n(\ln(\mathbf{x}))$$

where P_n is a polynomial function of degree n . With large enough n , this class of functions can approximate any function $R_+^p \rightarrow R_+$ arbitrarily well. For the stochastic growth model they suggest

$$\psi_t(K_{t-1}, \theta_t; \boldsymbol{\delta}) = \delta_1 K_{t-1}^{\delta_2} \theta_t^{\delta_3} = \exp P_1(\ln K_{t-1}, \ln \theta_t)$$

but also consider higher order power functions.

den Haan and Marcet (1994) propose a test of solution accuracy that can be applied to the method of parameterised expectations. This is implemented by increasing the degree of the power function and testing the significance of the additional coefficients.

5.1 The stochastic growth model in WinSolve

In the *WinSolve* model definition language the stochastic growth model can be written as

$$\begin{aligned} ltheta &= rho * ltheta(-1) + norm(sigma^2) ; theta = exp(ltheta) ; \\ c &= (beta * cexp(1)) ^(-1 / tau) ; \\ k &= theta * k(-1) ^alpha + mu * k(-1) - c ; \\ cexp &= C ^(-tau) * (alpha * theta * k(-1) ^alpha + mu) ; \end{aligned} \tag{9}$$

where the model parameters ρ , α , β , μ , τ , and σ have been coded as *WinSolve* parameters *rho*, *alpha*, *beta*, *mu*, *tau* and *sigma* respectively. The equation on the second line corresponds to the Euler condition (4) where *cexp(1)* is the forward expectation of *cexp* which is defined by the fourth line.

This model has forward expectations and so requires a solution algorithm that can impose consistent expectations. *WinSolve* has three solution methods that can be applied to this model: the *Fair-Taylor* method (Fair and Taylor 1983), the *Stacked Newton* method (Newton 1687, Laffargue 1990, Boucekkine 1995) and the *parameterised expectations* method (den Haan and Marcet 1990, 1994). In practice, the *Fair-Taylor* method has problems with highly nonlinear models and will not solve this model.

5.1.1 Parameterising expectations in WinSolve

WinSolve implements the parameterised expectations algorithm of den Haan and Marcet (1990) through a function defined in the model definition language. For the case of the growth model, the expectation *cexp(1)* can be parameterised by replacing the second line in (9) with

$$c = (\text{beta} * \text{parexp}(\text{cexp}(1), k(-1), \text{theta}, 1, 2))^{(-1/\text{tau})}); \quad (10)$$

The *WinSolve* function *parexp()* takes arguments defined by

$$\text{parexp}(y, x_1, \dots, x_p [\delta_1, \dots, \delta_k], n, p)$$

where y is the expectation to be parameterised, x_1, \dots, x_p are the state variables, n is the order of the power function and p is the number of state variables. $\delta_1, \dots, \delta_k$ represent *optional* initial values for the parameters of the power function. Good initial values will improve the speed of convergence of the method. When a model has been solved once with parameterised expectations, *WinSolve* will save the solution values of the parameter vector δ and will use these as starting values in subsequent solutions. This will speed up convergence in these subsequent runs.

Note that parameterising expectations does not require a separate solution algorithm in *WinSolve*. The *Fair-Taylor* method should be selected but the parameterised expectations algorithm will be doing all the work since, apart from the function *parexp()*, the model is completely backward looking.

The model parameters have been set to the values $\rho = .95$, $\alpha = .33$, $\beta = .95$, $\mu = 0.7$, $\tau = 1$, and $\sigma = .1$ corresponding to one of the high variance cases reported in den Haan and Marcet (1990). As can be seen, the two solutions are nearly identical. The final values for the power function parameters in the parameterised expectation are 1.44, $-.48$ and $-.74$ which

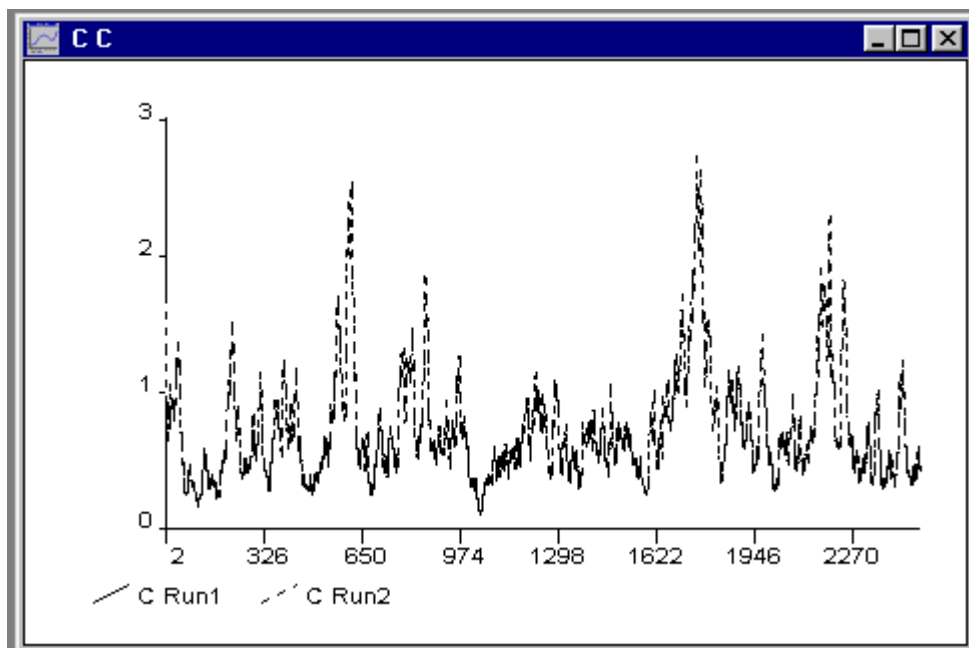


Figure 1: Dynamic simulation of consumption in stochastic growth model

are very close to the results reported by den Haan and Marcat for this case (1.44, $-.49$ and $-.72$).

5.2 Stochastic simulation

The dynamic solutions considered in the previous section are based on a single drawing of the random shock to technology. A new drawing will generate a different solution path. Consider taking repeated drawings of ε_t in (3), assuming a normal distribution with zero mean and variance σ^2 . The average of these drawings will be a consistent estimate of the expected value of the solution path. This procedure is known as *stochastic simulation* and increasing the number of drawings or *replications* increases its precision.

We will now do a stochastic simulation with the growth model, solving it using the parameterised expectations algorithm. The model is solved once without shocks to estimate the parameters of the expectations function. Then, in the stochastic replications, the expectations function is treated as fixed and not re-estimated. This is equivalent to the assumption that agents do not anticipate future shocks, so that the simulations are based on *rational* rather than *consistent* expectations. Computationally, it means that the simulations are much cheaper to perform.

The average of the stochastic simulations is plotted as the solid line in Figure (2) while the broken line represents a single draw dynamic solution.

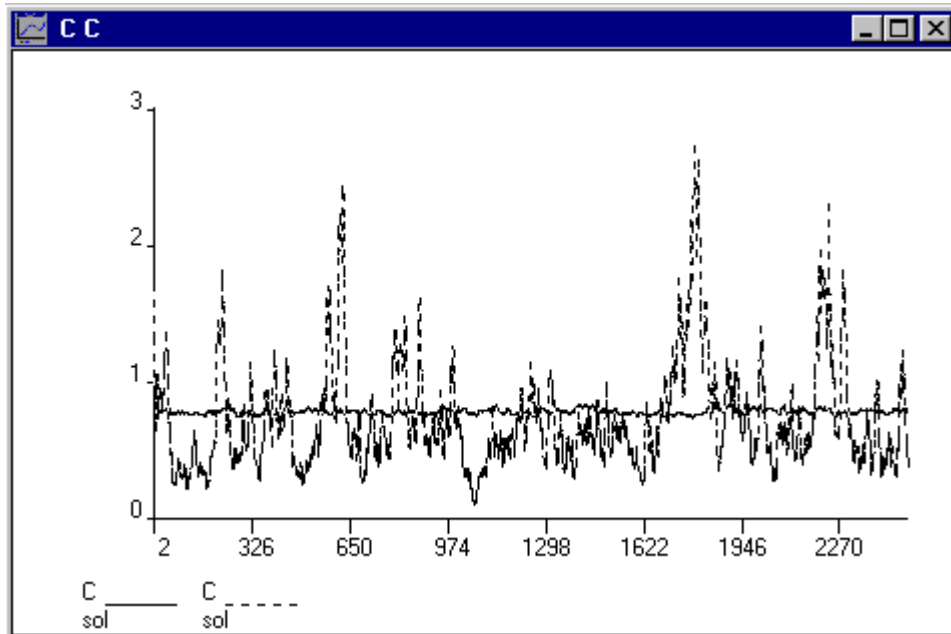


Figure 2: Consumption: stochastic simulation average versus one draw

References

- [1] Armstrong, J., Black, R., Laxton, D. and Rose, D. (1998), ‘A robust method for simulating forward-looking models’, *Journal of Economic Dynamics and Control*, 22, 489–501.
- [2] Boucekkine, R. (1995), ‘An alternative methodology for solving nonlinear forward-looking models’, *Journal of Economic Dynamics and Control*, 19, 711–734.
- [3] Brock, W.A. and L. Mirman (1972), ‘Optimal economic growth and uncertainty: the discounted case’, *Journal of Economic Theory*, 4, 479–513.
- [4] den Hann, W.J. and A. Marcet (1990), ‘Solving the stochastic growth model by parameterizing expectations’, *Journal of Business and Economic Statistics*, 8, 31–34.

- [5] den Haan, W.J. and A. Marcet (1994), ‘Accuracy in simulations’, *Review of Economic Studies*, 61, 3–17.
- [6] Duff, I.S. (1977), ‘MA28 – a set of Fortran subroutines for sparse unsymmetric linear equations’, Report AERE R8730, HMSO, London:UK.
- [7] Duff, I.S., Erisman, A.M. and Reid, J.K. (1986), *Direct Methods for Sparse Matrices*, Oxford University Press, Oxford: UK.
- [8] Fair, R. C. and J. B. Taylor (1983), ‘Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models’, *Econometrica*, 51, 1169–1186.
- [9] Gagnon, J.E. (1990), ‘Solving the stochastic growth model by deterministic extended path’, *Journal of Business and Economic Statistics*, 8, 35–36.
- [10] Judd, K.L. (1998), *Numerical Methods in Economics*, The MIT Press, Cambridge, MA, USA.
- [11] Juillard, M. (1996), ‘DYNARE: a program for the resolution and simulation of dynamic models with forward variables through the use of a relaxation algorithm’, CEPREMAP working paper No. 9602, Paris, France.
- [12] Juillard, M., Laxton, D., McAdam, P. and Pioro, H. (1998), ‘An algorithm competition: first-order iterations versus Newton-based techniques’, *Journal of Economic Dynamics and Control*, 22, 1291–1318.
- [13] Laffargue, J-P. (1990), ‘Résolution d’un modèle macroéconomique avec anticipations rationnelles’, *Annales d’Economie et de Statistique*, 17, 97–119.
- [14] Newton, I. (1686), *Philosophiae Naturalis Principia Mathematica*, translation by A. Motte (1729), revised by F. Cajori (1934), University of California Press, Berkeley, CA, USA.
- [15] Pauletto, G. (1997), *Computational Solution of Large-Scale Macroeconomic Models*, Kluwer Academic Publishers, Dordrecht, The Netherlands.
- [16] Pierse, R.G. (2002), ‘WinSolve: a users’ guide’, available at <http://www.econ.surrey.ac.uk/winsolve/>.

- [17] Rall, L.B. (1981), *Automatic Differentiation: Techniques and Applications*, Springer Verlag, Berlin, Germany.
- [18] Sargent, T.J. (1987), *Dynamic Macroeconomic Theory*, Harvard University Press, Cambridge MA, USA.
- [19] Tauchen, G. (1990), ‘Solving the stochastic growth model by using quadrature methods and value-function iterations’, *Journal of Business and Economic Statistics*, 8, 49–51.
- [20] Taylor, J.B. and H. Uhlig (1990), ‘Solving nonlinear stochastic growth models: a comparison of alternative solution methods’, *Journal of Business and Economic Statistics*, 8, 1–17.