Building and Solving Macroeconomic Models using WinSolve: Solving DSGE models[*]

Richard G. Pierse

April 2006

1 Introduction

This paper discusses the solution of *Dynamic Stochastic General Equilibrium* (*DSGE*) models using *WinSolve* (Pierse 2000). *DSGE* models present a number of challenges since in general they are non-linear stochastic rational expectations models. Traditionally, these models have been solved either by approximating the original problem by an equivalent linear quadratic (*LQ*) optimisation problem, or by linearising the first order conditions of the problem around the deterministic steady state and then solving the resulting linear rational expectations problem.

Recently, there has been some criticism in the literature of these popular approaches. Firstly, Judd (1996, 1998) has pointed out that the standard *LQ* approximation (first used by Kydland and Prescott (1982) and then copied by numerous other researchers) is not valid when the model is non-linear. A correct approximation, (first appearing in the economics literature in Magill (1977)), involves the computation of extra terms involving second order model derivatives. Levine, Pearlman and Pierse (2006) show that an alternative approach, due to Benigno and Woodford (2003) is equivalent to Magill. Secondly, there has been some question about the validity of first order approximations. These imply linear decision rules and certainty equivalence so that uncertainty does not affect decisions. Kim and Kim (2003) show that conventional first order approximation can lead to spurious welfare reversals where an incomplete-markets economy produces a higher

[*] This paper was prepared for a HKIMR Workshop held at HKMA, Hong Kong, April 11-12 2006.

level of welfare than the complete-markets economy. An alternative approach is to use perturbation methods. Widely used in the physical sciences for many years, these can be used to produce high order Taylor series approximations to non-linear decision rules that can incorporate the effects of uncertainty.

An alternative approach is to solve the non-linear problem directly, without using linear approximation. Non-linear solution methods (stacked-Newton or Fair-Taylor) produce time-paths of model variables (over a finite horizon) but do not produce decison rules explicitly. The assumption of perfect foresight (model-consistent expectations) sets errors on forward expectations to zero so that the problem becomes deterministic. However, the mean and higher moments of the distribution of the variables of the full stochastic model can be calculated through stochastic simulation. One non-linear solution method, the parameterised expectations method of den Haan and Marcet (1990), involves the approximation of forward expectations by general non-linear functions. This is similar to the perturbation method and does produce non-linear approximations to decision rules.

The curent version of *WinSolve* allows *DSGE* models to be solved either using non-linear methods (including the parameterised expectations method) or by linear (or log-linear) approximation. The next release of WinSolve will add perturbation methods (first and second order) and also correct *LQ* approximation using the Magill approach. Both these new methods will make use of automatic methods to compute high order derivatives using analytic formulae.

Section 2 of this paper discusses the alternative solution methods in general and Section 3 applies them to a simple stochastic growth model. The tutorial illustrates solving *DSGE* models in *WinSolve* using a *stochastic growth* model.

2 DSGE models

Dynamic Stochastic General Equilibrium (*DSGE*) models are consistently micro-founded economic models in which all agents are explicitly optimising and are forming *rational expectations* of the future. Technically, the models are *non-linear stochastic rational expectations* models.

Formally, a *DGSE* model can be defined by

$$\max E_t \sum_{t=0}^{T} w_t(\mathbf{y}_t, \mathbf{u}_t, \varepsilon_t) \text{ s.t. } \mathbf{f}_t(\mathbf{y}_t, \mathbf{y}_{t-1}, \mathbf{y}_{t+1}^e, \mathbf{u}_t, \varepsilon_t) = \mathbf{0} \tag{2.1}$$

where $\mathbf{y}_t$ is a vector of endogenous variables, $\mathbf{u}_t$ is a vector of exogenous (control) variables and ε_t is a vector of stochastic disturbances. w_t is a scalarvalued welfare function and $\mathbf{f}_t$ is a vector-valued set of functions defining the model. Note that the model equations depend on lagged values of $\mathbf{y}_t$ and rational expectations of future values of $\mathbf{y}_t$ as well as exogenous variables and stochastic shocks.

The first order conditions of the maximisation problem (2.1) are given by

$$\mathbf{g}_t(\mathbf{y}_t, \mathbf{y}_{t-1}, \mathbf{y}^e_{t+1}, \mathbf{u}_t, \varepsilon_t) = \mathbf{0}. \tag{2.2}$$

These will include Euler equations corresponding to the first order conditions with respect to $\mathbf{y}_t$, $\mathbf{u}_t$ and ε_t as well as the equations $\mathbf{f}_t() = 0$ which are the first order conditions with respect to the Lagrange multipliers.

The solution to the problem will be a decision rule for the endogenous variables of the form

$$\mathbf{y}^e_{t+1} = \mathbf{h}(\mathbf{y}_t, \mathbf{u}_t, \varepsilon_t). \tag{2.3}$$

2.1 Linear Quadratic Approximation

The traditional approach to solving *DSGE* models is to approximate the original problem (2.1) by a linear-quadratic (*LQ*) problem

$$\min_{\mathbf{u}_t} \sum_{t=0}^{T} \frac{1}{2} \left(\mathbf{y}_t'\mathbf{Q}\mathbf{y}_t + 2\mathbf{y}_t'\mathbf{S}\mathbf{u}_t + \mathbf{u}_t'\mathbf{R}\mathbf{u}_t\right) \tag{2.4}$$

subject to the *linear* model equations

$$\mathbf{A}\mathbf{y}_t = \mathbf{B}\mathbf{y}_{t-1} + \mathbf{C}\mathbf{u}_t$$

where $\mathbf{Q}$, $\mathbf{S}$ and $\mathbf{R}$ are matrices of weights. Note that in the approximate *LQ* problem, the stochastic disturbances have been dropped since the model displays certainty equivalence and the model equations $\mathbf{f}_t()$ have been replaced by a linear state-space system. The solution to this problem is a linear decision rule

$$\mathbf{y}_t = \mathbf{N}\mathbf{y}_{t-1} + \mathbf{M}\mathbf{u}_t$$

and a control rule

$$\mathbf{u}_t = \mathbf{F}\mathbf{y}_t.$$

In the standard application of this approximation, due to Kydland and Prescott (1982), the matrices $\mathbf{Q}$, $\mathbf{S}$, and $\mathbf{R}$ are derived from the second order derivatives of $w_t()$ and the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ from the first order derivatives of $\mathbf{f}_t()$. The derivatives are calculated at the deterministic steady-state solution of the model equations $\mathbf{f}_t()$. While this was correct in the original application by Kydland and Prescott, since in their model $\mathbf{f}_t()$ happened to be linear, in general this is not correct, as has been pointed out repeatedly by Judd (see for example Judd (1998)).

The correct approach, first stated (in the continuous time case) by Magill (1977), is to derive the approximation as a second order approximation to the Lagrangian (2.1). In this case, the matrices $\mathbf{Q}$, $\mathbf{S}$, and $\mathbf{R}$ also depend on the second-order derivatives of $\mathbf{f}_t()$ as well as the deterministic steady-state Lagrangian multpliers. The approximation should be calculated at the deterministic steady-state solution to the Lagrangian problem, which is *not the same point* as the approximation to the deterministic steady-state solution of the model equations $\mathbf{f}_t()$.

Benigno and Woodford (2003) derive a linear-quadratic approximation to a *DSGE* model in which, because of the inefficiency of the steady-state, linear terms appear in the objective function. Recognising that this implies that their quadratic objective function will not be second-order accurate, they substitute second-order approximations to the model equations into the objective function, transforming it into a function of quadratic terms only. Levine, Pearlman and Pierse (2006) show that the method of Benigno and Woodford is equivalent to the more general method of Magill.

2.2 Linear and Log-Linear Approximation

An alternative approach is to linearise (or log-linearise) the first-order conditions (2.2) of the original problem (2.1) around the deterministic steady state of the model. The resulting equations can be expressed in the linear state space form

$$\mathbf{A}\mathbf{y}^e_{t+1} = \mathbf{B}\mathbf{y}_t + \mathbf{C}\mathbf{u}_t. \tag{2.5}$$

When the matrix $\mathbf{A}$ is non-singular, then the model can also be re-expressed as

$$\mathbf{y}^e_{t+1} = \mathbf{A}^{-1}\mathbf{B}\mathbf{y}_t + \mathbf{A}^{-1}\mathbf{C}\mathbf{u}_t. \tag{2.6}$$

This linear rational expectations model can be solved using linear methods. In the non-singular case, (2.6), the method of Blanchard and Kahn (1980), can be applied. The condition for a unique solution is that the number of unstable roots of $\mathbf{A}^{-1}\mathbf{B}$ is equal to the number of non-predetermined (jump) variables in the system. In the singular case, a generalised Schur *qz* decomposition can be applied, as described in Klein (2000) and Sims (2002). An alternative approach for singular models by system reduction is proposed by King and Watson (1998) and (2002) and matlab code is available in *reds/solds*.

In the linearised (or log-linearised) model, certainty equivalence applied. Certainty equivalence is a property of a stochastic model which means that its deterministic solution is an unbiased estimator of the expected values of its model variables. When the conditions for certainty equivalence hold, then a deterministic solution is sufficient, provided that only the expected value and not the higher order moments are of interest. The approximation will be adequate for the expected values of the variables in a neighbourhood of the approximation point but, if the model is very non-linear, then this neighbourhood may be very small.

2.3 Non-linear solution methods

Non-linear solution methods solve the first order conditions (2.2) explicitly for a solution over a finite time horizon $t = 1, \cdots, T$. The assumption of perfect foresight is made to reduce the problem to a deterministic one. However, the mean and higher moments of the distribution of the variables of the full stochastic model can be calculated through stochastic simulation. Two general approaches are possible: a standard algorithm for solving non-linear rational expectations problems such as stacked-Newton or Fair-Taylor, or the parameterised expectations algorithm.

2.4 Parameterising expectations

The principle of the parameterised expectations method of den Haan and Marcet (1990) is to approximate the non-linear decision rule

$$\mathbf{y}^e_{t+1} = \mathbf{h}(\mathbf{y}_t, \mathbf{u}_t, \varepsilon_t)$$

by a known function of a set of parameters that can be estimated.

$$\mathbf{y}^e_{t+1} = \psi_t(\mathbf{x}_t; \delta)$$

where $\mathbf{x}_t$ is the $k \times 1$ set of predetermined state variables. Typically the functional form chosen is a power function which is the antilogarithm of a polynomial and can approximate a positive variable to arbitrary accuracy.

The model solution is an estimation procedure to find δ to minimise

$$(\mathbf{y}_{t+1} - \psi_t(\mathbf{x}_t; \delta))^2.$$

Given a choice of δ , the model is no longer forward-looking and can be solved using any non-linear solution technique.

A test of the accuracy of the approximation is proposed in den Haan and Marcet (1994). This is implemented by increasing the degree of the power function and testing the significance of the additional coefficients.

2.5 Perturbation Methods

Perturbation is a general method for providing accurate approximations to nonlinear problems, based on the application of Taylor's theorem and the implicit function theorem. When analyic derivatives can be computed automatically, these approximations can be generated recursively, involving only the solution of linear sets of equations. While these methods have been widely used in the physical sciences for many years, it is only recently that their importance in economics has come to be recognised. Judd (1996) provides a survey of these methods applied to economic problems. He shows how traditional ad hoc methods of linear-quadratic approximation do not always produce locally valid Taylor series for equilibrium behaviour rules. In contrast, perturbation methods always produce valid approximations of the chosen order. Furthermore, such approximations are easy to compute recursively to any order of accuracy.

One problem with standard linear-quadratic approximations is that they impose the assumption of certainty equivalence, even when the true problem is generally not certainty equivalent. Judd and Guu (1993), Collard and Julliard (2001) and Schmitt-Grohé and Uribe (2004) extend perturbation methods to a stochastic environment, avoiding the assumption of certainty equivalence. Kim and Kim (2003) show that conventional linearisation can lead to spurious welfare reversals where an incomplete-markets economy produces a higher level of welfare than the complete-markets economy. These problems are avoided by using second-order perturbation methods to approximate welfare.Various applications of perturbation methods are suggested by Collard and Julliard (2001), Jin and Judd (2002), Kim and Kim (2003) and Schmitt-Grohé and Uribe (2004). Matlab code (requiring the symbolic toolbox for analytic differentiation) is provided by Judd and also by Schmitt-Grohé and Uribe.

3 A simple example

In this section, a simple stochastic growth model will be used to illustrate some of the different methods of solution of *DGSE* models. This model is of interest because: (a) although it is extremely simple, it does not admit to analytic solution and (b) it has been extensively discussed in the literature, in particular in a special issue of the *Journal of Business and Economic Statistics* (see Taylor and Uhlig (1990) and subsequent articles).

The problem to be solved is:

$$\max E_0 \sum_{t=0}^{\infty} \beta^t (1-\tau)^{-1} c_t^{1-\tau} \tag{3.1}$$

subject to

$$\begin{aligned} c_t + k_t &= \theta_t k_{t-1}^{\alpha} + \mu k_{t-1} & (3.2)\\ \ln \theta_t &= \rho \ln \theta_{t-1} + \varepsilon_t, & (3.3) \end{aligned}$$

where c_t is consumption, k_t is the end of period capital stock, and θ_t is technology. Technology θ_t is assumed to be stochastic, following an autoregressive process with coefficient ρ, $|\rho| < 1$, where the shock ε_t is a serially uncorrelated normally distributed random variable with zero mean and constant variance σ^2. The other parameters of the model are: τ, the coefficient of relative risk aversion $0 < \tau < 1$, μ, (one minus) the rate of capital depreciation, $0 \leq \mu \leq 1$ and β, the rate of time discount, $0 < \beta < 1$.

The Lagrangian of the problem is

$$L = \max E_0 \sum_{t=0}^{\infty} \beta^t \left[\begin{array}{c} (1-\tau)^{-1} c_t^{1-\tau} + \lambda_{1t}(c_t + k_t - \theta_t k_{t-1}^{\alpha} - \mu k_{t-1}) \\ +\lambda_{2t}(\ln \theta_t - \rho \ln \theta_{t-1} - \varepsilon_t) \end{array} \right]. \tag{3.4}$$

First order conditions for a maximum are

$$\frac{\partial L_t}{\partial c_t} = \beta^t (c_t^{-\tau} + \lambda_{1t}) = 0 \tag{3.5}$$

$$\frac{\partial (L_t + L_{t+1})}{\partial k_t} = \beta^t \lambda_{1t} - \beta^{t+1} E_t[\lambda_{1t+1}(\theta_{t+1} \alpha k_t^{\alpha-1} + \mu)] = 0 \tag{3.6}$$

$$\frac{\partial (L_t + L_{t+1})}{\partial \theta_t} = \beta^t (\lambda_{2t} \theta_t^{-1} - \lambda_{1t} k_{t-1}^{\alpha}) - \beta^{t+1} \lambda_{2t+1} \rho \theta_t^{-1} = 0 \tag{3.7}$$

$$\frac{\partial L_t}{\partial \lambda_{1t}} = c_t + k_t - \theta_t k_{t-1}^{\alpha} - \mu k_{t-1} = 0 \tag{3.8}$$

$$\frac{\partial L_t}{\partial \lambda_{2t}} = \ln \theta_t - \rho \ln \theta_{t-1} - \varepsilon_t = 0 \tag{3.9}$$

Substituting (3.5) into (3.6) and rearranging gives the Euler equation

$$c_t^{-\tau} = \beta E_t[c_{t+1}^{-\tau}(\theta_{t+1} \alpha k_t^{\alpha-1} + \mu)]. \tag{3.10}$$

Solving the model is achieved either by explicitly solving the optimisation problem (3.4) or by solving the first order conditions defined by the set of equations (3.10) together with (3.8) and (3.9). (Note that the equation (3.7) that serves to determine λ_{2t} can be neglected since λ_{2t} doesn't appear in any other equation.)

The solution of the problem will be a pair of nonlinear decision rules

$$\begin{aligned} c_t &= h_1(k_{t-1}, \theta_t) \\ k_t &= h_2(k_{t-1}, \theta_t). \end{aligned}$$

A deterministic steady state of the model is given by

$$\begin{aligned} c^* &= (\frac{\alpha\beta}{1-\beta\mu})^{\alpha/(1-\alpha)} + (\mu-1)(\frac{\alpha\beta}{1-\beta\mu})^{1/(1-\alpha)} \\ k^* &= (\frac{\alpha\beta}{1-\beta\mu})^{1/(1-\alpha)} \\ \theta^* &= 1. \end{aligned} \tag{3.11}$$

3.1 Linear Quadratic Approximation

McGratten (1990) derives a linear quadratic approximation to the nonlinear optimisation problem (3.4) around the deterministic steady state (3.11). However, her solution, which follows the approach of Kydland and Prescott (1982), has been criticised by Judd (1998), because it neglects the second order terms arising from the curvature of the model equations (3.2) and (3.3) at the steady state. The correct linear quadratic approximation follows Magill (1977). Levine, Pearlman and Pierse (2006) show that this is identical to the approach used by Benigno and Woodford (2003) in the context of a different model.

The Magill approach is equivalent to taking a second order Taylor expansion of the Lagrangian around the steady state solution of the deterministic optimisation problem. Computing second order derivatives at the steady state we have

$$\frac{\partial^2 L_t}{\partial c_t^2} = -\tau\beta^t c_*^{-\tau-1} = Q_{11}, \quad \frac{\partial^2 L_t}{\partial c_t \partial k_t} = 0 = Q_{12}$$

$$\frac{\partial^2 (L_t + L_{t+1})}{\partial k_t^2} = -\beta^{t+1}[\lambda_{1*}\alpha(\alpha-1)k_*^{\alpha-2}] = Q_{22}$$

$$\frac{\partial^2 (L_t + L_{t+1})}{\partial k_t \partial \theta_t} = -\beta^t \lambda_{1*}\alpha k_*^{\alpha-1} = Q_{23}$$

$$\frac{\partial^2 (L_t + L_{t+1})}{\partial \theta_t^2} = \beta^t \lambda_{2*} - \beta^{t+1}\lambda_{2*}\rho = Q_{33}$$

$$\frac{\partial^2 L}{\partial\lambda_{1t}\partial c_t} = 1 = J_{11}$$
$$\frac{\partial^2 (L_t + L_{t+1})}{\partial\lambda_{1t}\partial k_t} = 1 - \alpha k_*^{\alpha-1} - \mu = J_{12}$$
$$\frac{\partial^2 L_t}{\partial\lambda_{1t}\partial\theta_t} = -k_*^{\alpha} = J_{13}$$
$$\frac{\partial^2 L_t}{\partial\lambda_{2t}\partial c_t} = 0 = J_{21}, \quad \frac{\partial^2 L_t}{\partial\lambda_{2t}\partial k_t} = 0 = J_{22}$$
$$\frac{\partial^2 (L_t + L_{t+1})}{\partial\lambda_{2t}\partial\theta_t} = (1-\rho) = J_{23}.$$

This gives rise to a second order approximation of the Lagrangian

$$\begin{aligned} L_t(c_t, k_t, \theta_t) \quad &\simeq \quad L_t(c_*, k_*, 1) + \frac{1}{2}\begin{pmatrix} c_t - c_* \\ k_t - k_* \\ \theta_t - 1 \end{pmatrix}' \begin{bmatrix} Q_{11} & 0 & 0 \\ 0 & Q_{22} & Q_{23} \\ 0 & Q_{23}' & Q_{33} \end{bmatrix} \begin{pmatrix} c_t - c_* \\ k_t - k_* \\ \theta_t - 1 \end{pmatrix} \\ &+ \begin{pmatrix} \lambda_{1t} - \lambda_1^* \\ \lambda_{2t} - \lambda_2^* \end{pmatrix}' \begin{pmatrix} 1 & J_{12} & J_{13} \\ 0 & 0 & J_{23} \end{pmatrix} \begin{pmatrix} c_t - c_* \\ k_t - k_* \\ \theta_t - 1 \end{pmatrix}. \end{aligned}$$

3.2 Linear and Log-Linear Approximation

Christiano (1990) discusses solving the stochastic growth model by linear and log-linear approximation. He derives explicit formulae for the linear (3.12) and log-linear (3.13) approximations to the decision rule for k_t taken around the deterministic steady state values k_* and c_*:

$$b_0 = (1-\lambda)k_*, \quad b_1 = \lambda, \quad b_2 = \frac{q\lambda}{1-\beta\rho\lambda}$$
$$k_t = b_0 + b_1\theta_t + b_2 k_{t-1} \tag{3.12}$$
$$a_0 = (1-\lambda)\log k_*, \quad a_1 = \frac{q}{k_*}\frac{\lambda}{1-\beta\rho\lambda}, \quad a_2 = \lambda$$
$$\log k_t = a_0 + a_1 \log\theta_t + a_2 \log k_{t-1} \tag{3.13}$$

where

$$q = \beta\left[(1-\rho)\left(\frac{c_*}{k_*} + 1 - \mu\right) + \frac{\rho\beta}{\tau}(\beta^{-1} - \mu)\frac{c_*}{k_*}\right]k_*$$
$$\psi = 1 + \beta^{-1} + \frac{(1-\alpha)(1-\beta\mu)}{\tau}\frac{c_*}{k_*}$$

and

$$\lambda = \frac{\psi - \sqrt{\psi^2 - 4/\beta}}{2}.$$

3.3 Non-linear solution methods

Non-linear solution methods solve the first order conditions (3.10) , (3.8) and (3.9) explicitly for a solution over a finite time horizon $t = 1, \cdots, T$. The assumption of perfect foresight is made to reduce the problem to a deterministic one. Two general approaches are possible: a standard algorithm for solving non-linear rational expectations problems such as stacked-Newton or Fair-Taylor, or the parameterised expectations algorithm.

General first-order solution methods such as Fair-Taylor require that the equations are normalised, with a different endogenous variable on the left-hand side of each equation. One such normalisation is

$$c_t = [\beta c_{t+1}^{-\tau}(\theta_{t+1}\alpha k_t^{\alpha-1} + \mu)]^{-1/\tau} \tag{3.14}$$

$$k_t = \theta_t k_{t-1}^{\alpha} + \mu k_{t-1} - c_t.$$

An alternative normalisation is

$$c_t = \theta_t k_{t-1}^{\alpha} + \mu k_{t-1} - k_t$$

$$k_t = k_t + (\theta_t k_{t-1}^{\alpha} + \mu k_{t-1} - k_t)^{-\tau} - \beta(\mu + \theta_{t+1}^{e}\alpha k_t^{\alpha-1})(\theta_{t+1}k_t^{\alpha} + \mu k_t - k_{t+1})^{-\tau}. \tag{3.15}$$

Note that in the first normalisation, c_t is the jump variable. In the second, there are two jump variables, k_t and θ_t.

One important consideration in the solution of finite horizon non-linear *RE* models, is the setting of terminal conditions for the jump variables. In the first normalisation, a value needs to be set for c_{T+1}. In the second normalisation, terminal values are needed for k_{T+1} and θ_{T+1}. Terminal conditions can help pin down a solution, even in cases where no steady state or multiple steady states may exist.

3.4 Parameterising expectations

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

$$c_{t+1}^{e} \tag{3.16}$$

in (3.10) where

$$c_{t+1}^{e} = c_t[\beta(\alpha\theta_t k_{t-1}^{\alpha-1} + \mu)]^{\frac{1}{\tau}} .$$

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

$$c_{t+1}^{e} = c_{t+1} .$$

den Haan and Marcet (1990) propose a general method for solving models by approximating expectations such as (3.16) using a functional form

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

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

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

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

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

where

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

is the $T \times k$ matrix of derivatives of ψ with respect to the parameters δ evaluated at iteration $s-1$.

The functional form of ψ 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 $R_+^p \to R_+$ arbitrarily well. For the stochastic growth model they suggest

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

but also consider higher order power functions.

4 WinSolve tutorial on *DSGE* models

WinSolve provides several algorithms for solving *DSGE* models. The stacked-Newton and Fair-Taylor algorithms allow direct solution of the non-linear first-order conditions or the expectations can be parameterised using the den Haan and Marcet algorithm. Alternatively, an automatic linear or non-linear approximation can be done and the linearised model solved via the *qz* algorithm. This tutorial demonstrates these different solution methods on the stochastic growth model developed in Section 3.

4.1 Non-linear methods

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

$$\begin{aligned} \mathit{ltheta} &= \mathit{rho*ltheta(\text{-}1)+norm(sigma2);}\ \ \mathit{theta{=}exp(ltheta);} \\ c &= \mathit{(beta*c(1)\hat{}(\text{-}tau)*(theta(1)*alpha*k\hat{}\ (alpha\text{-}1)+mu))\hat{}(\text{-}1/tau);} \qquad (4.1) \\ k &= \mathit{theta*k(\text{-}1)\hat{}alpha + mu*k(\text{-}1)\ \text{-}\ c;} \end{aligned}$$

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 (3.10). The model parameters have been set to the values $\rho = .95$, $\alpha = .33$, $\beta = .95$, $\mu = 1.0$, $\tau = 0.5$, and $\sigma = .1$ corresponding to case 1 in Taylor and Uhlig (1990), which is a high variance case.

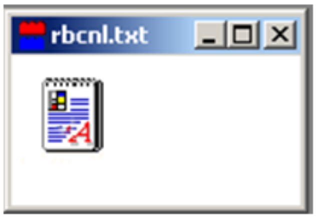


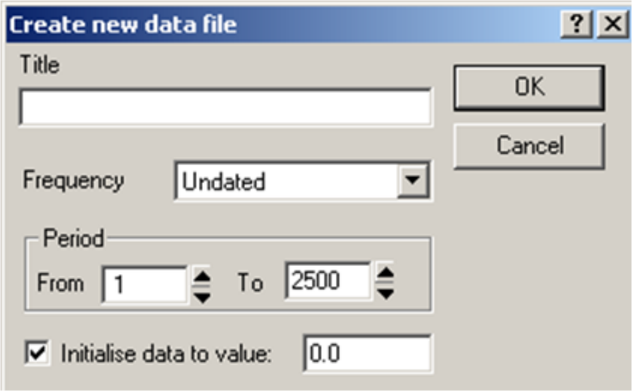
Figure 4.1: The model object

The model to be opened is file *rbcnl.txt*. The equations can seen by double clicking on the icon in the model object window (Figure 4.1) or selecting *Edit model* from the *File* menu, which opens the text editor.

The first step is to solve numerically for the *deterministic steady state* of the model. In order to do this, we first need to create a new data file of 2500 undated observations that will be replaced by the steady state solution. Click on the *Create new data file* icon 🗋, or select the *Create new data file...* option from the *Data* menu. In the dialog box, select frequency *Undated* and data period 1 to 2500. Finally, check the *Initialise data* box to initialise all observations to zero. These options are illustrated in Figure 4.2. Click on the *OK* box to exit.

Before solving for the steady state, the variables c and k must be reset to a positive value, otherwise, the solution will fail. Go into the *Edit data / adjustments* dialog box and reset the data values for c and k to 1.

Now we can solve for the deterministic steady state of the model. Click on the *Solve model* icon ⚡ or select *Solve model ...* from the Solve menu.



Title:

Frequency: Undated

Period

From: 1 To: 2500

☑ Initialise data to value: 0.0

Figure 4.2: Create new data file dialog box

Choose *Steady state solution* from the Solution mode list as in Figure **??** and click *OK*. Viewing the results, it will be seen that the deterministic steady state values of k and c are 15.4864 and 2.46993 respectively. It can be verified that these values indeed correspond to the analytic solutions given in (3.11). Having found the deterministic steady state, this will now be used as the data base for further runs of the model. From the Solve menu, select the option *Set last solution as base.* A popup box will warn you that ”*this will overwrite your current base values*”. Click on *Yes* to agree to the change.

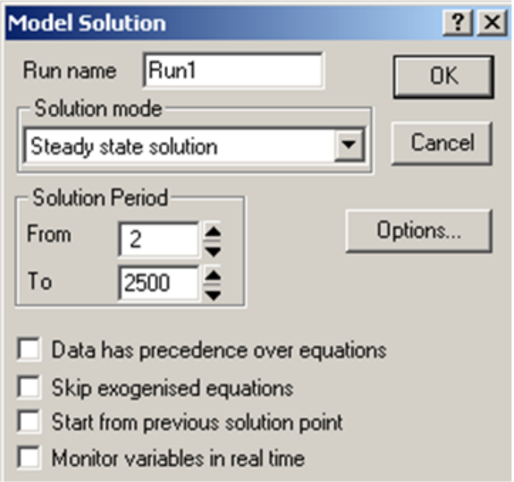
The model can now be solved dynamically, using the deterministic steady state as base values. Click on the *Solve model* icon ⚡ or select *Solve model ...* from the Solve menu. This time, choose *Dynamic model solution* from the Solution mode list. The non-linear model equations will be solved dynamically using the stacked Newton algorithm and the assumption of perfect foresight.

The model will now be solved again, this time using the parameterised expectations algorithm of den Haan and Marcet (1990). *WinSolve* implements the parameterised expectations algorithm through a function defined in the model definition language. For the case of the stochastic growth model, the expected value *c(1)* can be parameterised by replacing the original equation for c in (4.1) with

$$\begin{aligned} &cexp{=}c\hat{\ }tau{*}(alpha{*}theta{*}k(\text{-}1)\hat{\ }(alpha\text{-}1){+}mu); \\ &c = (beta{*}parexp(cexp(1),k(\text{-}1),theta,1,2))\hat{\ }(1/tau); \end{aligned} \tag{4.2}$$

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

$$parexp(y, x_1, \cdots, x_p\ [, \delta_1, \cdots, \delta_k]\ , n, p)$$



Run name Run1

Solution mode: Steady state solution

Solution Period

From 2

To 2500

□ Data has precedence over equations

□ Skip exogenised equations

□ Start from previous solution point

□ Monitor variables in real time

Figure 4.3: Model solution dialog box

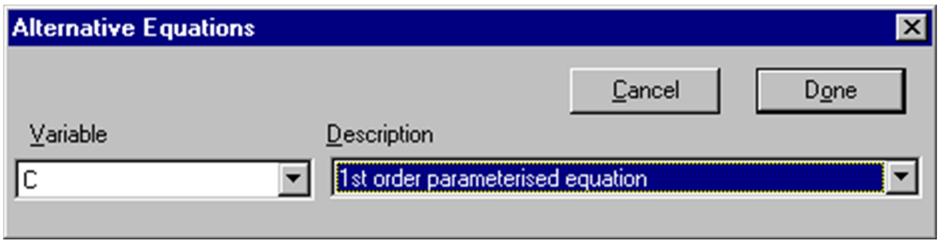
where y is the expectation to be parameterised, $x_1, \cdots, x_p$ are the state variables, n is the order of the power function and p is the number of state variables. $\delta_1, \cdots, \delta_k$ represent *optional* initial values for the parameters of the power function. Good initial values will improve the speed of convergence of the method. Once a model has been solved 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 will be automatically 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 equations (4.2) have already been included in the file *rbcnl.txt* as an alternative equation for c. To activate the parameterised expectations algorithm, all we need to do is to switch to the alternative equation. Select the *Switch alternative equations* option in the *Assumptions* menu which brings up the dialog box in Figure (4.4). Choose the equation with description '1st order parameterised equation' from the Description list box, and click *Done.*

Now the model must be solved again. *WinSolve* will automatically select an appropriate solution method for the parameterised expectation equation. Simply click on the Solve model icon ⚡ or select the *Solve model* option from the *Solve* menu, and then click on *OK* to commence model solution.

The three solutions to the model for variable c are graphed in Figure 4.5.


Alternative Equations

Variable: C

Description: 1st order parameterised equation

Figure 4.4: Switch alternative equations dialog box

It can be seen that there is very little difference between the stacked Newton and parameterised expectations solutions. (Both were solved using the same drawing of shocks to technology).

4.2 Approximation methods

We now want to solve the stochastic growth model using linear and log-linear approximation of the non-linear model equations (4.1). It will be demonstrated that the approximate linear decision rules computed analytically by Christiano (1990) can be replicated in *WinSolve* using an automatic linearisation (or log-linearisation) procedure computed at the deterministic steady state.

In order to proceed, the model needs to be written in a slightly different way, based on equation (3.15), where the variable c has been subsituted out so that the only variables are k and θ. The rewritten model is given by

```
log(theta) = rho*log(theta(-1)) + norm(sigma*sigma);
k = k+(theta(-1)*k(-1) ^ alpha + mu*k(-1)-k)^(-tau)
-beta*(mu+theta*alpha*k^(alpha-1))*(theta*k^ alpha+mu*k-k(1))^(-tau);
```

and the model equations are in file *rbclin.txt*.

As before, the first step is to solve for the deterministic steady state. Open the model and create a new data file of 2500 undated observations, initialised this time to 1. Then solve for the steady state and set this solution as the data base. It should be verified that the steady state solution for k is 15.4864 as before.

To compute a linearisation around the deterministic steady state, select *Linear solution options...* from the *Solve* menu. The *Model linearisation dialog box* shown in Figure 4.6 will open. Choose *Data base* as the run to use and choose observation 4 (3 or greater) as the point at which to linearise.

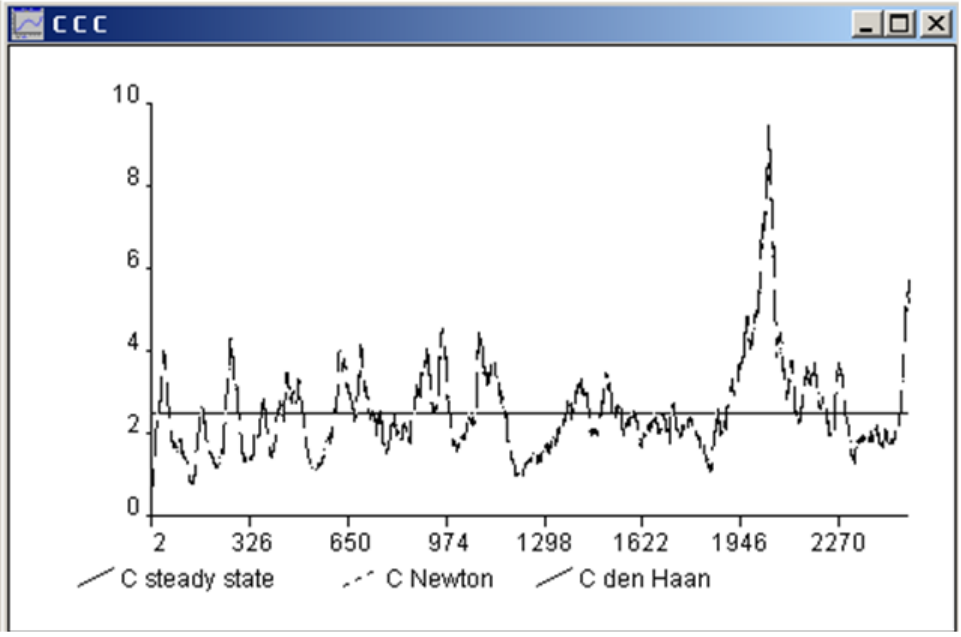


Figure 4.5: Nonlinear solutions of stochastic growth model

Select *linearise* or *log-linearise* as appropriate and uncheck all boxes except *Compute Blanchard-Kahn solution.*Then click *OK* to finish.

The output is presented in a table. Firstly, the (generalised) eigenvalues of the state space representation are shown and the Blanchard-Kahn conditions checked. Then the decision matrices are displayed. For the linear case, the eigenvalues are

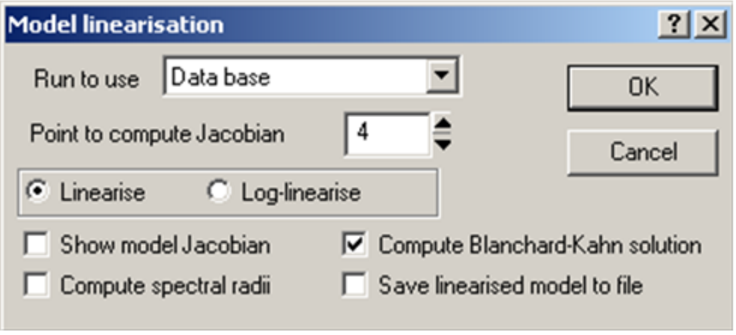
$$1.14, \quad 0.95, \quad 0.92$$

so that there is one unstable eigenvalue and two stable eigenvalues and the conditions for a unique solution are satisfied. In this case there are no infinite eigenvalues so that the state space representation is non-singular. The slope coefficients of the decision rule are given in the table:

	θ_t	k_{t-1}
k_t	1.88723	0.923547

The present version of *WinSolve* does not display the intercept in the decision rule (the next release will do so). Nevertheless, the slope coefficients are identical with those computed from the analytical formulae derived by Christiano in (3.12) which gives the rule as

$$k_t = 1.18398 + 1.887230\theta_t + 0.923547k_{t-1}.$$



Run to use: Data base

Point to compute Jacobian: 4

● Linearise ○ Log-linearise

□ Show model Jacobian ☑ Compute Blanchard-Kahn solution

□ Compute spectral radii □ Save linearised model to file

Figure 4.6: Model linearisation dialog box

Similarly, for the log-linear approximation, the eigenvalues are

$$1.14, \quad 0.95, \quad 0.92$$

as before and the coefficients of the decision rule are

$$\begin{array}{ccc} & \theta_t & k_{t-1} \\ k_t & 0.121863 & 0.923547 \end{array}$$

The Christiano decision rule in this case is

$$\log k_t = 0.209478 + 0.121863 \log \theta_t + 0.923547 \log k_{t-1}$$

so that, as before, the slope coefficients produced by *WinSolve* are identical with those derived by Christiano. Note that it is only possible to derive decision rules analytically in very simple cases, whereas the numerical procedure used by WinSolve can be applied for any model, however large.

References

[1] Benigno, P. and M. Woodford (2003), 'Optimal monetary and fiscal policy: a linear quadratic approach', in M. Gertler and K. Rogoff (eds.) *NBER Macroeconomics Annual 2003*, MIT Press, Cambridge, MA, USA.

[2] Blanchard, O. and C. Kahn (1980), 'The solution of linear difference models under rational expectations', *Econometrica*, 48, 1305–1311.

[3] Christiano, L.J. (1990), 'Solving the stochastic growth model by linear-quadratic approximation and by value-function iteration', *Journal of Business and Economic Statistics*, 8, 23–26.

[4] Collard, F. and M. Juillard (2001), 'Accuracy of stochastic perturbation methods: the case of asset pricing models', *Journal of Economic Dynamics and Control*, 25, 979–999.

[5] den Haan, W.J. and A. Marcet (1990), 'Solving the stochastic growth model by parameterizing expectations', *Journal of Business and Economic Statistics*, 8, 31–34.

[6] den Haan, W.J. and A. Marcet (1994), 'Accuracy in simulations', *Review of Economic Studies*, 61, 3–17.

[7] Jin, H and K.L. Judd (2002), 'Perturbation methods for general dynamic stochastic models', mimeo.

[8] Judd, K.L. (1996), 'Approximation, perturbation and projection methods in economic analysis', chapter 12 in H.M. Amman, D.A. Kendrick and J. Rust (eds.), *Handbook of Computational Economics, Volume I*, Elsevier Science, Amsterdam, 509–585.

[9] Judd, K.L. (1998), *Numerical Methods in Economics*, The MIT Press, Cambridge, MA, USA.

[10] Judd, K.L. and S.-M. Guu (1993), 'Perturbation solution methods for economic growth models', in H. Varian (ed.), *Economic and Financial Modelling with Mathematica*, Springer-Verlag, New York, 80–103.

[11] Kim, J. and S.H. Kim (2003), 'Spurious welfare reversals in international business cycle models', *Journal of International Economics*, 60, 471–500.

[12] King, R. G. and M. W. Watson (1998), 'The solution of singular linear difference systems under rational expectations', *International Economic Review*, 39, 1015–1026.

[13] King, R.G. and M.W. Watson (2002), 'System reduction and solution algorithms for singular linear difference systems under rational expectations', *Computational Economics*, 20, 57–86.

[14] Klein, P. (2000), 'Using the generalized Schur form to solve a multivariate linear rational expectations model', *Journal of Economic Dynamics and Control*, 24, 1405–1423.

[15] Kydland, P.E. and E.C. Prescott (1982), 'Time to build and aggregate fluctuations', *Econometrica*, 50, 1345–1370.

[16] Levine, P., J. Pearlman and R.G. Pierse (2006), 'Linear-Quadratic approximation:, efficiency and target-implementability', Department of Economics, University of Surrey.

[17] Magill, M. (1977), 'A local analysis of n-sector capital accumulation under uncertainty', *Journal of Economic Theory*, 15, 211–218.

[18] McGratten, E.R. (1990), 'Solving the stochastic growth model by linear-quadratic approximation', *Journal of Business and Economic Statistics*, 8, 41–43.

[19] Pierse, R. G. (2000), 'WinSolve Version 3: An introductory Guide', Department of Economics, University of Surrey.

[20] Schmitt-Grohé, S. and M. Uribe (2004), 'Solving dynamic general equilibrium models using a second-order approximation to the policy function', *Journal of Economic Dynamics and Control*, 28, 755–775.

[21] Sims, C.A. (2002), 'Solving linear rational expectations models', *Computational Economics*, 20, 1–20.

[22] 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.

[23] Woodford, M. (2002), 'Inflation stabilization and welfare' *Contributions to Macroeconomics*, 2, Article 1.

[24] Woodford, M. (2003), *Interest and Prices: Foundations of a Theory of Monetary Policy*, Princeton University Press, NJ.