

# Solving Dynamic General Equilibrium Models Using a Second-Order Approximation to the Policy Function\*

Stephanie Schmitt-Grohé<sup>†</sup>                      Martín Uribe<sup>‡</sup>  
Rutgers University and CEPR              University of Pennsylvania

First draft: June 2001  
This draft: August 28, 2001

## Abstract

This paper derives a second-order approximation to the solution of rational expectations, dynamic, general equilibrium models. To illustrate its applicability, the method is used to solve the dynamics of a simple neoclassical model. The paper closes with a brief description of a set of MATLAB programs designed to implement the method.  
*JEL Classification:* E0, C63.

*Key words:* Solving Dynamic General Equilibrium Models, Second Order Approximation, Matlab code.

---

\*We benefited from discussions on second-order approximations with Fabrice Collard, Ken Judd, Jinill Kim, Robert Kollmann, and Chris Sims.

<sup>†</sup>Phone: 732 932 2960. E-mail: grohe@econ.rutgers.edu.

<sup>‡</sup>Phone: 215 898 6260. E-mail: uribe@econ.upenn.edu.

# 1 Introduction

Since the seminal paper of Kydland and Prescott (1982) and King, Plosser, and Rebelo (1988), it has become commonplace in macroeconomics to approximate the solution to non-linear, dynamic, general equilibrium models using linear methods. Linear approximation methods are useful to characterize certain aspects of the dynamic properties of complicated models. In particular, if the support of the shocks driving aggregate fluctuations is small and an interior stationary solution exists, first-order approximations provide adequate answers to questions such as local existence and determinacy of equilibrium and the size of the second moments of the variables describing the economy.

However, first-order approximation techniques are not well suited to handle questions such as welfare comparisons across alternative stochastic or policy environments. For example, Kim and Kim (1998) show that in a simple two-agent economy, a welfare comparison based on a linear approximation to the policy function and a quadratic approximation to the utility function may yield the spurious result that welfare is higher under autarky than under full risk sharing. The problem with using linearized decision rules to evaluate second-order approximations to the objective function is that some second-order terms of the objective function are ignored when using a linearized decision rule.<sup>1</sup> Such problems do not arise when the policy function is approximated to second order or higher.

In this paper we derive a second-order approximation to the policy function of a dynamic, rational expectations model. Our approach follows the perturbation method described in Judd (1998) and Collard and Juillard (2001). We follow Collard and Juillard closely in notation and methodology. An important difference separates this paper from the work of Collard and Juillard. Namely, Collard and Juillard apply what they call a ‘bias reduction procedure’ to capture the fact that the policy function depends on the variance of the underlying shocks. Instead, we explicitly incorporate a scale parameter for the variance of the exogenous shocks as an argument of the policy function. In approximating the policy function, we take a second-order Taylor expansion with respect to the state variables as well as this scale parameter.<sup>2</sup> Because our approach is not iterative, it allows for a more precise and efficient computation of the coefficients associated with the variance of the shocks.

What distinguishes our paper from the related literature is the conjunction of the following elements: (a) We consider a very general class of dynamic stochastic discrete-time rational expectations models. A strength of our approach is not to follow a value function formulation. This allows us to tackle easily a wide variety of model economies that do not lend themselves naturally to the value function specification. (b) We provide MATLAB code to compute second-order approximations for any rational expectations model whose equilibrium conditions can be written in the general form considered in this paper. (c) We show analytically that in general the first derivative of the policy function with respect to the parameter scaling the variance/covariance matrix of the shocks is zero regardless of whether the model has the certainty-equivalence property or not. (d) In addition, we show that the cross derivative of the policy function with respect to the state vector and with respect to

---

<sup>1</sup>See Woodford (1999) for a discussion of conditions under which it is correct up to second order to approximate the level of welfare using first-order approximations to the policy function.

<sup>2</sup>This technique was formally introduced by Fleming (1971) and has been applied extensively to economic models by Judd and Judd and coauthors (see Judd (1998) and the references cited therein).

the parameter scaling the variance/covariance matrix of the shocks is zero in general.<sup>3</sup>

A recent paper by Sims (2000) also derives second-order approximations to the solution of dynamic general equilibrium models. Sims's approach to finding the second-order approximation to the policy functions is akin to the method of undetermined coefficients frequently applied to solve non-stochastic difference equations. To facilitate comparison with Sims's work, later in the paper, we apply our second-order approximation method to a simple model economy that Sims has worked out using his proposed solution.

In the next section, we present the model. In section 3 we derive first- and second-order approximations to the policy functions. In section 4 we illustrate the applicability of the method by computing the solution to the basic neoclassical growth model. We close the paper with a description of the MATLAB computer code we designed to implement the second-order solution.

## 2 The Model

The set of equilibrium conditions of a wide variety of dynamic general equilibrium models in Macroeconomics is of the form

$$E_t f(y_{t+1}, y_t, x_{t+1}, x_t) = 0, \quad (1)$$

where  $E_t$  denotes the mathematical expectations operator conditional on information available at time  $t$ . The state vector  $x_t$  is of size  $n_x \times 1$  and the co-state vector  $y_t$  is of size  $n_y \times 1$ . We define  $n = n_x + n_y$ . The function  $f$  maps  $R^{n_y} \times R^{n_y} \times R^{n_x} \times R^{n_x}$  into  $R^n$ . The state vector  $x_t$  can be partitioned as  $x_t = [x_t^1; x_t^2]'$ . The vector  $x_t^1$  consists of endogenous predetermined state variables and the vector  $x_t^2$  of exogenous state variables. Specifically, we assume  $x_t^2$  follows the exogenous stochastic process given by

$$x_{t+1}^2 = \Lambda x_t^2 + \tilde{\eta} \sigma \epsilon_{t+1}; \quad \epsilon_t \sim N(\emptyset, I).$$

where both the vector  $x_t^2$  and the innovation  $\epsilon_t$  are of order  $n_\epsilon \times 1$ . The vector  $\epsilon_t$  is independently, identically, and normally distributed with mean zero and variance/covariance matrix  $I$ . The scalar  $\sigma \geq 0$  and the  $n_\epsilon \times n_\epsilon$  matrix  $\tilde{\eta}$  are known parameters. All eigenvalues of the matrix  $\Lambda$  are assumed to have modulus less than one.

### 2.1 Example

Consider the simple neoclassical growth model given by

$$\begin{aligned} c_t^{-\gamma} &= \beta E_t c_{t+1}^{-\gamma} [\alpha A_{t+1} k_{t+1}^{\alpha-1} + 1 - \delta] \\ c_t + k_{t+1} &= A_t k_t^\alpha + (1 - \delta) k_t \\ \ln A_{t+1} &= \rho \ln A_t + \sigma \epsilon_{t+1} \end{aligned}$$

---

<sup>3</sup>We learned through private communication with Ken Judd that he obtained results (c) and (d) simultaneously and independently.

for all  $t \geq 0$ , given  $k_0$  and  $A_0$ . Let  $y_t = c_t$  and  $x_t = [k_t; \ln A_t]'$ . Then

$$E_t f(y_{t+1}, y_t, x_{t+1}, x_t) = E_t \begin{bmatrix} y(1)_t^{-\gamma} - \beta y(1)_{t+1}^{-\gamma} [\alpha e^{x(2)_{t+1}} x(1)_{t+1}^{\alpha-1} + 1 - \delta] \\ y(1)_t + x(1)_{t+1} - e^{x(2)_t} x(1)_t^\alpha - (1 - \delta)x(1)_t \\ x(2)_{t+1} - \rho x(2)_t \end{bmatrix}$$

## 2.2 Solution

The solution to the model given in equation (1) is of the form:

$$y_t = g(x_t, \sigma) \quad (2)$$

$$x_{t+1} = h(x_t, \sigma) + \eta \sigma \epsilon_{t+1}, \quad (3)$$

where  $g$  maps  $R^{n_x} \times R^+$  into  $R^{n_y}$  and  $h$  maps  $R^{n_x} \times R^+$  into  $R^{n_x}$ . The matrix  $\eta$  is of order  $n_x \times n_\epsilon$  and is given by

$$\eta = \begin{bmatrix} \emptyset \\ \tilde{\eta} \end{bmatrix}$$

We wish to find a second-order approximation of the functions  $g$  and  $h$  around the non-stochastic steady state,  $x_t = \bar{x}$  and  $\sigma = 0$ , where  $\bar{x}$  is defined next.

## 2.3 The non-stochastic steady state

We define the non-stochastic steady state as vectors  $(\bar{x}, \bar{y})$  such that

$$f(\bar{y}, \bar{y}, \bar{x}, \bar{x}) = 0$$

It is clear that  $\bar{y} = g(\bar{x}, 0)$  and  $\bar{x} = h(\bar{x}, 0)$ . To see this, note that if  $\sigma = 0$ , then  $E_t f = f$ .

## 3 Approximating the Solution

Substituting the proposed solution given by equations (2) and (3) into the model, equation (1), we can define

$$F(x, \sigma) = E_t f(g(h(x, \sigma) + \eta \sigma \epsilon', \sigma), g(x, \sigma), h(x, \sigma) + \eta \sigma \epsilon', x) \quad (4)$$

$$= 0 \quad (5)$$

Here we are dropping time subscripts if the variable is dated in period  $t$  and use a prime to indicate variables dated in period  $t + 1$ .

Because  $F(x, \sigma)$  must be equal to zero for any possible values of  $x$  and  $\sigma$ , it must be the case that the derivatives of any order of  $F$  must also be equal to zero. Formally,

$$F_{x^k \sigma^j}(x, \sigma) = 0 \quad \forall x, \sigma, j, k, \quad (6)$$

where  $F_{x^k \sigma^j}(x, \sigma)$  denotes the derivative of  $F$  with respect to  $x$  taken  $k$  times and with respect to  $\sigma$  taken  $j$  times.

### 3.1 First-order approximation

We are looking for approximations to  $g$  and  $h$  around the point  $(x, \sigma) = (\bar{x}, 0)$  of the form

$$g(x, \sigma) = g(\bar{x}, 0) + g_x(\bar{x}, 0)(x - \bar{x}) + g_\sigma(\bar{x}, 0)\sigma$$

$$h(x, \sigma) = h(\bar{x}, 0) + h_x(\bar{x}, 0)(x - \bar{x}) + h_\sigma(\bar{x}, 0)\sigma$$

As explained earlier,

$$g(\bar{x}, 0) = \bar{y}$$

and

$$h(\bar{x}, 0) = \bar{x}.$$

The remaining unknown coefficients of the first-order approximation to  $g$  and  $h$  are identified by using the fact that, by equation (6), it must be the case that:

$$F_x(\bar{x}, 0) = 0$$

and

$$F_\sigma(\bar{x}, 0) = 0.$$

Thus, using the first of these two expressions,  $g_x$  and  $h_x$  can be found as the solution to the system

$$\begin{aligned} [F_x(\bar{x}, 0)]_j^i &= [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_x]_j^\beta + [f_y]_\alpha^i [g_x]_j^\alpha + [f_{x'}]_\beta^i [h_x]_j^\beta + [f_x]_j^i \\ &= 0; \quad i = 1, \dots, n; \quad j, \beta = 1, \dots, n_x; \quad \alpha = 1, \dots, n_y \end{aligned}$$

Here we are using tensor notation. So, for example,  $[f_{y'}]_\alpha^i$  is the  $(i, \alpha)$  element of the derivative of  $f$  with respect to  $y'$ . The derivative of  $f$  with respect to  $y'$  is an  $n \times n_y$  matrix. Therefore,  $[f_{y'}]_\alpha^i$  is the element of this matrix located at the intersection of the  $i$ -th row and  $\alpha$ -th column.<sup>4</sup> Note that the derivatives of  $f$  evaluated at  $(y', y, x', x) = (\bar{y}, \bar{y}, \bar{x}, \bar{x})$  are known. The above expression represents a system of  $n \times n_x$  equations in the  $n \times n_x$  unknowns given by the elements of  $g_x$  and  $h_x$ .

Similarly,  $g_\sigma$  and  $h_\sigma$  are identified as the solution to the following  $n$  equations:

$$\begin{aligned} [F_\sigma(\bar{x}, 0)]^i &= E_t\{[f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_\sigma]^\beta + [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [\eta]_\phi^\beta [\epsilon']^\phi + [f_{y'}]_\alpha^i [g_\sigma]^\alpha + [f_y]_\alpha^i [g_\sigma]^\alpha \\ &\quad + [f_{x'}]_\beta^i [h_\sigma]^\beta + [f_{x'}]_\beta^i [\eta]_\phi^\beta [\epsilon']^\phi\} \\ &= [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_\sigma]^\beta + [f_{y'}]_\alpha^i [g_\sigma]^\alpha + [f_y]_\alpha^i [g_\sigma]^\alpha + [f_{x'}]_\beta^i [h_\sigma]^\beta \\ &= 0; \quad i = 1, \dots, n; \quad \alpha = 1, \dots, n_y; \quad \beta = 1, \dots, n_x; \quad \phi = 1, \dots, n_\epsilon. \end{aligned} \quad (7)$$

Note that this equation is linear and homogeneous in  $g_\sigma$  and  $h_\sigma$ . Thus, if a unique solution exists, we have that

$$h_\sigma = 0.$$

and

$$g_\sigma = 0.$$

---

<sup>4</sup>This is a variation of the tensor notation found in the mathematical literature. Our notation follows Juillard and Collard (2001).

### 3.2 Second-order approximation

The second-order approximations to  $g$  and  $h$  around the point  $(x, \sigma) = (\bar{x}, 0)$  are of the form

$$\begin{aligned}
[g(x, \sigma)]^i &= [g(\bar{x}, 0)]^i + [g_x(\bar{x}, 0)]_a^i [(x - \bar{x})]_a + [g_\sigma(\bar{x}, 0)]^i [\sigma] \\
&\quad + \frac{1}{2} [g_{xx}(\bar{x}, 0)]_{ab}^i [(x - \bar{x})]_a [(x - \bar{x})]_b \\
&\quad + \frac{1}{2} [g_{x\sigma}(\bar{x}, 0)]_a^i [(x - \bar{x})]_a [\sigma] \\
&\quad + \frac{1}{2} [g_{\sigma x}(\bar{x}, 0)]_a^i [(x - \bar{x})]_a [\sigma] \\
&\quad + \frac{1}{2} [g_{\sigma\sigma}(\bar{x}, 0)]^i [\sigma] [\sigma] \\
[h(x, \sigma)]^j &= [h(\bar{x}, 0)]^j + [h_x(\bar{x}, 0)]_a^j [(x - \bar{x})]_a + [h_\sigma(\bar{x}, 0)]^j [\sigma] \\
&\quad + \frac{1}{2} [h_{xx}(\bar{x}, 0)]_{ab}^j [(x - \bar{x})]_a [(x - \bar{x})]_b \\
&\quad + \frac{1}{2} [h_{x\sigma}(\bar{x}, 0)]_a^j [(x - \bar{x})]_a [\sigma] \\
&\quad + \frac{1}{2} [h_{\sigma x}(\bar{x}, 0)]_a^j [(x - \bar{x})]_a [\sigma] \\
&\quad + \frac{1}{2} [h_{\sigma\sigma}(\bar{x}, 0)]^j [\sigma] [\sigma],
\end{aligned}$$

where  $i = 1, \dots, n_y$ ,  $a, b = 1, \dots, n_x$ , and  $j = 1, \dots, n_x$ . The unknowns of this expansion are  $[g_{xx}]_{ab}^i$ ,  $[g_{x\sigma}]_a^i$ ,  $[g_{\sigma x}]_a^i$ ,  $[g_{\sigma\sigma}]^i$ ,  $[h_{xx}]_{ab}^j$ ,  $[h_{x\sigma}]_a^j$ ,  $[h_{\sigma x}]_a^j$ ,  $[h_{\sigma\sigma}]^j$ , where we have omitted the argument  $(\bar{x}, 0)$ . These coefficients can be identified by taking the derivative of  $F(x, \sigma)$  with respect to  $x$  and  $\sigma$  twice and evaluating them at  $(x, \sigma) = (\bar{x}, 0)$ . By the arguments provided earlier, these derivatives must be zero. Specifically, we use  $F_{xx}(\bar{x}, 0)$  to identify  $g_{xx}(\bar{x}, 0)$  and  $h_{xx}(\bar{x}, 0)$ . That is,<sup>5</sup>

$$\begin{aligned}
[F_{xx}(\bar{x}, 0)]_{jk}^i &= \left( [f_{y'y'}]_{\alpha\gamma}^i [g_x]_\delta^\gamma [h_x]_k^\delta + [f_{y'y'}]_{\alpha\gamma}^i [g_x]_k^\gamma + [f_{y'x'}]_{\alpha\delta}^i [h_x]_k^\delta + [f_{y'x}]_{\alpha k}^i \right) [g_x]_\beta^\alpha [h_x]_j^\beta \\
&\quad + [f_{y'}]_\alpha^i [g_{xx}]_{\beta\delta}^\alpha [h_x]_k^\delta [h_x]_j^\beta \\
&\quad + [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_{xx}]_{jk}^\beta \\
&\quad + \left( [f_{yy'}]_{\alpha\gamma}^i [g_x]_\delta^\gamma [h_x]_k^\delta + [f_{yy}]_{\alpha\gamma}^i [g_x]_k^\gamma + [f_{yx'}]_{\alpha\delta}^i [h_x]_k^\delta + [f_{yx}]_{\alpha k}^i \right) [g_x]_j^\alpha \\
&\quad + [f_y]_\alpha^i [g_{xx}]_{jk}^\alpha \\
&\quad + \left( [f_{x'y'}]_{\beta\gamma}^i [g_x]_\delta^\gamma [h_x]_k^\delta + [f_{x'y}]_{\beta\gamma}^i [g_x]_k^\gamma + [f_{x'x'}]_{\beta\delta}^i [h_x]_k^\delta + [f_{x'x}]_{\beta k}^i \right) [h_x]_j^\beta \\
&\quad + [f_{x'}]_\beta^i [h_{xx}]_{jk}^\beta \\
&\quad + [f_{xy'}]_{j\gamma}^i [g_x]_\delta^\gamma [h_x]_k^\delta + [f_{xy}]_{j\gamma}^i [g_x]_k^\gamma + [f_{xx'}]_{j\delta}^i [h_x]_k^\delta + [f_{xx}]_{jk}^i \\
&= 0; \quad i = 1, \dots, n_y, \quad j, k, \beta, \delta = 1, \dots, n_x; \quad \alpha, \gamma = 1, \dots, n_y.
\end{aligned}$$

---

<sup>5</sup>At this point, an additional word about tensor notation is in order. Take for example the expression  $[f_{y'y'}]_{\alpha\gamma}^i$ . Note that  $f_{y'y'}$  is a three dimensional array with  $n$  rows,  $n_y$  columns, and  $n_y$  pages. Then  $[f_{y'y'}]_{\alpha\gamma}^i$  denotes the element of  $f_{y'y'}$  located at the intersection of row  $i$ , column  $\alpha$  and page  $\gamma$ .

Since we know the derivatives of  $f$  as well as the first derivatives of  $g$  and  $h$  evaluated at  $(y', y, x', x) = (\bar{y}, \bar{y}, \bar{x}, \bar{x})$ , it follows that the above expression represents a system of  $n \times n_x \times n_x$  linear equations in the  $n \times n_x \times n_x$  unknowns given by the elements of  $g_{xx}$  and  $h_{xx}$ .

Similarly,  $g_{\sigma\sigma}$  and  $h_{\sigma\sigma}$  can be obtained by solving the linear system  $F_{\sigma\sigma}(\bar{x}, 0) = 0$ . More explicitly,

$$\begin{aligned}
[F_{\sigma\sigma}(\bar{x}, 0)]^i &= [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_{\sigma\sigma}]^\beta \\
&\quad + [f_{y'y'}]_{\alpha\gamma}^i [g_x]_\delta^\gamma [\eta]_\xi^\delta [g_x]_\beta^\alpha [\eta]_\phi^\beta [I]_\xi^\phi \\
&\quad + [f_{y'x'}]_{\alpha\delta}^i [\eta]_\xi^\delta [g_x]_\beta^\alpha [\eta]_\phi^\beta [I]_\xi^\phi \\
&\quad + [f_{y'}]_\alpha^i [g_{xx}]_{\beta\delta}^\alpha [\eta]_\xi^\delta [\eta]_\phi^\beta [I]_\xi^\phi \\
&\quad + [f_{y'}]_\alpha^i [g_{\sigma\sigma}]^\alpha \\
&\quad + [f_y]_\alpha^i [g_{\sigma\sigma}]^\alpha \\
&\quad + [f_{x'}]_\beta^i [h_{\sigma\sigma}]^\beta \\
&\quad + [f_{x'y'}]_{\beta\gamma}^i [g_x]_\delta^\gamma [\eta]_\xi^\delta [\eta]_\phi^\beta [I]_\xi^\phi \\
&\quad + [f_{x'x'}]_{\beta\delta}^i [\eta]_\xi^\delta [\eta]_\phi^\beta [I]_\xi^\phi \\
&= 0; \quad i = 1, \dots, n; \quad \alpha, \gamma = 1, \dots, n_y; \quad \beta, \delta = 1, \dots, n_x; \quad \phi, \xi = 1, \dots, n_\epsilon.
\end{aligned} \tag{8}$$

This is a system of  $n$  linear equations in the  $n$  linear unknowns given by the elements of  $g_{\sigma\sigma}$  and  $h_{\sigma\sigma}$ .

Finally, we show that the cross derivatives  $g_{x\sigma}$  and  $h_{x\sigma}$  are equal to zero when evaluated at  $(\bar{x}, 0)$ . We write the system  $F_{\sigma x}(\bar{x}, 0) = 0$  taking into account that all terms containing either  $g_\sigma$  or  $h_\sigma$  are zero at  $(\bar{x}, 0)$ . Then we have,

$$\begin{aligned}
[F_{\sigma x}(\bar{x}, 0)]_j^i &= [f_{y'}]_\alpha^i [g_x]_\beta^\alpha [h_{\sigma x}]_j^\beta + [f_{y'}]_\alpha^i [g_{\sigma x}]_\gamma^\alpha [h_x]_j^\gamma + [f_y]_\alpha^i [g_{\sigma x}]_j^\alpha + [f_{x'}]_\beta^i [h_{\sigma x}]_j^\beta \\
&= 0; \quad i = 1, \dots, n; \quad \alpha = 1, \dots, n_y; \quad \beta, \gamma, j = 1, \dots, n_x.
\end{aligned} \tag{9}$$

This is a system of  $n \times n_x$  equations in the  $n \times n_x$  unknowns given by the elements of  $g_{\sigma x}$  and  $h_{\sigma x}$ . But clearly, the system is homogeneous in the unknowns. Thus, if a unique solution exists, it is given by

$$g_{\sigma x} = 0$$

and

$$h_{\sigma x} = 0.$$

### 3.3 Higher-order approximations

It is straightforward to apply the method described thus far to finding higher-order approximations to the policy function. For example, given the first- and second-order terms of the Taylor expansion of  $h$  and  $g$ , the third-order terms can be identified by solving a linear system of equations. More generally, one can construct sequentially the  $n$ th-order approximation of the policy function by solving a linear system of equations whose (known) coefficients are the lower-order terms and the derivatives up to order  $n$  of  $f$  evaluated at  $(y', y, x', x) = (\bar{y}, \bar{y}, \bar{x}, \bar{x})$ .

## 4 An Application: The Neoclassical Growth Model

In this section we apply the second-order approximation method to solve the the simple neoclassical model, described in section 2.1.

We calibrate the model by setting  $\beta = 0.95$ ,  $\delta = 1$ ,  $\alpha = 0.3$ ,  $\rho = 0$ , and  $\gamma = 2$ . We choose these parameter values to facilitate comparison with the results obtained by applying Sims's (2000) method.<sup>6</sup> Here we are interested in a quadratic approximation to the policy function around the natural logarithm of the steady state. Thus, unlike in section 2.1, we now define:

$$x_t = \begin{bmatrix} \ln k_t \\ \ln A_t \end{bmatrix}$$

and

$$y_t = \ln c_t.$$

Then the steady-state values of  $y_t$  and  $x_t$ , which coincide with the constant terms of the quadratic expansion of  $g$  and  $h$ , respectively, are:

$$\bar{y} = -0.8734.$$

and

$$\bar{x} = \begin{bmatrix} -1.7932 \\ 0 \end{bmatrix}$$

The coefficients of the linear terms are:

$$g_x = [0.2525 \quad 0.8417]$$

and

$$h_x = \begin{bmatrix} 0.4191 & 1.3970 \\ 0.0000 & 0.0000 \end{bmatrix}$$

The coefficients of the quadratic terms are given by:

$$g_{xx}(:, :, 1) = [-0.0051 \quad -0.0171]$$

$$g_{xx}(:, :, 2) = [-0.0171 \quad -0.0569]$$

and

$$h_{xx}(:, :, 1) = \begin{bmatrix} -0.0070 & -0.0233 \\ 0 & 0 \end{bmatrix}$$

$$h_{xx}(:, :, 2) = \begin{bmatrix} -0.0233 & -0.0778 \\ 0 & 0 \end{bmatrix}$$

Finally, the coefficients of the quadratic terms in  $\sigma$  are:

$$g_{\sigma\sigma} = -0.1921$$

---

<sup>6</sup>See the MATLAB script sessionEG.m in Sims's website (<http://eco-072399b.princeton.edu/yftp/gensys2/GrowthEG>)



and

$$h_{\sigma\sigma} = \begin{bmatrix} 0.4820 \\ 0 \end{bmatrix}.$$

A more familiar representation is given by the evolution of the original variables. Let

$$\hat{c}_t \equiv \ln(c_t/\bar{c})$$

and

$$\hat{k}_t \equiv \ln(k_t/\bar{k}).$$

Then, the laws of motion of these two variables is given by

$$\hat{c}_t = 0.2525\hat{k}_t + 0.8417\hat{A}_t + \frac{1}{2} \left[ -0.0051\hat{k}_t^2 - 0.0341\hat{k}_t\hat{A}_t - 0.0569\hat{A}_t^2 - 0.1921\sigma^2 \right]$$

and

$$\hat{k}_{t+1} = 0.4191\hat{k}_t + 1.3970\hat{A}_t + \frac{1}{2} \left[ -0.0070\hat{k}_t^2 - 0.0467\hat{k}_t\hat{A}_t - 0.0778\hat{A}_t^2 + 0.4820\sigma^2 \right].$$

## 5 Matlab Codes

We prepared a set of Matlab codes that implements the second-order approximation developed above. The programs can be found on line at [http://www.econ.upenn.edu/~uribe/2nd\\_order.htm](http://www.econ.upenn.edu/~uribe/2nd_order.htm).

The program `gx_hx.m` computes the matrices  $g_x$  and  $h_x$ . The inputs to the program are the first derivatives of  $f$  evaluated at the steady state. That is,  $f_y$ ,  $f_x$ ,  $f_{y'}$ , and  $f_{x'}$ . This step amounts to obtaining a first-order approximation to the policy functions. A number of packages are available for this purpose. We use the one prepared by Paul Klein of the University of Western Ontario, which consists of the three programs `solab.m`, `qzswitch.m`, and `reorder.m`.

The program `gxx_hxx.m` computes the arrays  $g_{xx}$  and  $h_{xx}$ . The inputs to the program are the first and second derivatives of  $f$  evaluated at the steady state and the matrices  $g_x$  and  $h_x$  produced by `gx_hx.m`.

The program `gss_hss.m` computes the arrays  $g_{\sigma\sigma}$  and  $h_{\sigma\sigma}$ . The inputs to the program are the first and second derivatives of  $f$  evaluated at the steady state, the matrices  $g_x$  and  $h_x$  produced by the program `gx_hx.m`, the array  $g_{xx}$  produced by the program `gxx_hxx.m`, and the matrix  $\eta$ .

### 5.1 Computing the derivatives of $f$

Computing the derivatives of  $f$ , particularly the second derivatives, can be a daunting task if the model is large. We approach this problem as follows. The MATLAB Toolbox Symbolic Math can handle analytical derivatives. We wrote programs that compute the analytical derivatives of  $f$  and evaluate them at the steady state:

The program `anal_deriv.m` computes the analytical derivatives of  $f$  and the program `num_eval.m` evaluates the analytical derivatives of  $f$  at the steady state.

## 5.2 Example: The Neoclassical Growth Model

To illustrate the use of the programs described thus far, we include the programs needed to obtain the second-order approximation to the decision rules of the simple neoclassical model discussed in section 4. To obtain the second-order approximation to the policy functions of the neoclassical growth model run the program `neoclassical_model_run.m`. The output of this program are the matrices  $g_x$  and  $h_x$  and the arrays  $g_{xx}$ ,  $h_{xx}$ ,  $g_{\sigma\sigma}$  and  $h_{\sigma\sigma}$ .

This program calls the program `neoclassical_model.m`, which produces the first- and second derivatives of  $f$ . More generally, `neoclassical_model.m` illustrates how to write down analytically the equations of a DSGE model using the MATLAB Toolbox Symbolic Math.

## References

- Collard, Fabrice, and Michel Juillard, “Perturbation Methods for Rational Expectations Models,” Manuscript. Paris: CEPREMAP, February 2001.
- Fleming, W. H., “Stochastic Control for Small Noise Intensities,” *SIAM Journal of Control* 9, 1971, 473-517.
- Judd, Kenneth, *Numerical Methods in Economics*, Cambridge, MA, MIT Press, 1998.
- Kim, Jinill and Sunghyun Henry Kim, “Inaccuracy of Log-Linear Approximation in Welfare Calculations: The Case of International Risk Sharing,” Manuscript. Charlottesville: The University of Virginia, October 1998.
- King, R.G., C.L. Plosser and S.T. Rebelo, “Production, Growth and Business Cycles I: The Basic Neoclassical Model,” *Journal of Monetary Economics* 21, 1988, 191-232.
- Kydland, Finn and Edward Prescott, “Time to Build and Aggregate Fluctuations,” *Econometrica* 50, November 1982, 1345-1370.
- Sims, Christopher, “Second Order Accurate Solution of Discrete Time Dynamic Equilibrium Models,” Manuscript. Princeton: Princeton University, December 2000.
- Woodford, Michael, “Inflation Stabilization and Welfare,” Manuscript. Princeton: Princeton University, June 1999.