Rheinische Friedrich Wilhelms Universität
Bonn Graduate School of Economics
June 26 –July 4, 2008

# Lectures on Monetary Theory and Policy

Day 3

Instructor: Stephanie Schmitt-Grohé
Duke University, grohe@duke.edu

# Lecture 3: Perturbation Methods for the Numerical Analysis of DSGE Models

**Implementation of Optimal Policy**

- Ramsey solution is mute on how to implement it

- The state contingent rule may not result in a unique equilibrium

## Policy Evaluation

- Evaluate different stabilization policies

- Welfare based policy evaluation

$$V_t = E_t \sum_{t=0}^{\infty} \beta^t U(c_t, h_t)$$

- How to compute $V_t$?

- Why is a first-order approximation not enough?

- Second-order accurate welfare measures

## Perturbation Methods

- Eqm conditions of a wide variety of dynamic stochastic general equilibrium models can be written in the form of a nonlinear stochastic vector difference equation

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

$x_t$ — predetermined (or state) variables, size $n_x \times 1$
$y_t$ — nonpredetermined (or control) variables, size $n_y \times 1$
Let $n = n_x + n_y$.
$f : R^{n_y} \times R^{n_y} \times R^{n_x} \times R^{n_x} \mapsto R^n$.

$x_0$ — initial condition for the economy
No-Ponzi game constraint — terminal condition

- We restrict attention to stationary solutions to (1), that is, ignore the no-Ponzi game constraint

- $x_t = [x_t^1;\ x_t^2]'$
  $x_t^1$— endogenous predetermined state variables
  $x_t^2$— exogenous state variables
  assume $x_t^2$ follows exogenous stochastic process

$$x_{t+1}^2 = \tilde{h}(x_t^2, \sigma) + \tilde{\eta}\sigma\epsilon_{t+1},$$

  $x_t^2$, and $\epsilon_t$ are of order $n_\epsilon \times 1$.

- $\epsilon_t$ has bounded support
  $\epsilon_t \sim$ i.i.d, mean zero and variance/covariance matrix $I$.

- Eigenvalues of the Jacobian of the function $\tilde{h}$ with respect to its first argument are assumed to lie within the unit circle.

**Solution of the Model**

- Solution to equation (1)

$$y_t = \widehat{g}(x_t) \tag{2}$$

$$x_{t+1} = \widehat{h}(x_t) + \eta\sigma\epsilon_{t+1}. \tag{3}$$

The matrix $\eta$ is of order $n_x \times n_\epsilon$

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

- The shape of the functions $\widehat{h}$ and $\widehat{g}$ will in general depend on the amount of uncertainty in the economy. The key idea of perturbation methods is to interpret the solution to the model as a function of the state vector $x_t$ and of the parameter $\sigma$ scaling the amount of uncertainty in the economy, that is,

$$y_t = g(x_t, \sigma); \quad g : R^{nx} \times R^{+} \mapsto R^{ny} \tag{4}$$

and

$$x_{t+1} = h(x_t, \sigma) + \eta\sigma\epsilon_{t+1}; \quad h : R^{nx} \times R^{+} \mapsto R^{nx} \tag{5}$$

- A perturbation methods finds a *local* approximation of the functions $g$ and $h$, i.e., an approximation that is valid in the neighborhood of a particular point $(\bar{x}, \bar{\sigma})$.

- Taking a Taylor series approximation of the functions $g$ and $h$ around $(x,\sigma)=(\bar{x},\bar{\sigma})$ (assume $n_x=n_y=1$)

$$\begin{aligned}
g(x,\sigma) &= g(\bar{x},\bar{\sigma})+g_x(\bar{x},\bar{\sigma})(x-\bar{x})+g_\sigma(\bar{x},\bar{\sigma})(\sigma-\bar{\sigma}) \\
&\quad +\frac{1}{2}g_{xx}(\bar{x},\bar{\sigma})(x-\bar{x})^2+g_{x\sigma}(\bar{x},\bar{\sigma})(x-\bar{x})(\sigma-\bar{\sigma}) \\
&\quad +\frac{1}{2}g_{\sigma\sigma}(\bar{x},\bar{\sigma})(\sigma-\bar{\sigma})^2+\ldots \\
h(x,\sigma) &= h(\bar{x},\bar{\sigma})+h_x(\bar{x},\bar{\sigma})(x-\bar{x})+h_\sigma(\bar{x},\bar{\sigma})(\sigma-\bar{\sigma}) \\
&\quad +\frac{1}{2}h_{xx}(\bar{x},\bar{\sigma})(x-\bar{x})^2 \\
&\quad +h_{x\sigma}(\bar{x},\bar{\sigma})(x-\bar{x})(\sigma-\bar{\sigma}) \\
&\quad +\frac{1}{2}h_{\sigma\sigma}(\bar{x},\bar{\sigma})(\sigma-\bar{\sigma})^2+\ldots,
\end{aligned}$$

  The unknowns of an $n^{th}$ order expansion are the n-th order derivatives of the functions $g$ and $h$ evaluated at the point $(\bar{x},\bar{\sigma})$.

- How to find the n-th order derivatives of the functions $g$ and $h$ evaluated at the point $(\bar{x}, \bar{\sigma})$?

  Substitute the proposed solution given by equations (4) and (5) into equation (1)

$$
\begin{aligned}
F(x,\sigma) &\equiv E_t f(g(h(x,\sigma)+\eta\sigma\epsilon',\sigma), g(x,\sigma), h(x,\sigma)+\eta\sigma\epsilon', x) \\
&= 0. \qquad (6)
\end{aligned}
$$

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,

$$F_{x^k \sigma^j}(x, \sigma) = 0 \quad \forall x, \sigma, j, k, \tag{7}$$

- A particularly convenient point to approximate the functions $g$ and $h$ around is the non-stochastic steady state, $x_t = \bar{x}$ and $\sigma = 0$. We define the non-stochastic steady state as vectors $(\bar{x}, \bar{y})$ such that

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

Then

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

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

- In principle can approximate $g$ and $h$ around any arbitrary point $(x_t, \sigma)$

provided derivatives of $F(x, \sigma)$ known at that point. For example, approximate $g$ and $h$ around a point $x_t \neq \bar{x}$ and $\sigma = 0$.

**Example: The neoclassical growth model**

$$\max E_0 \sum_{t=0}^{\infty} \beta^t \frac{c_t^{1-\gamma}}{1-\gamma} \qquad 0 < \beta < 1, \gamma > 0, \text{ and } \gamma \neq 1.$$

Period-by-period budget constraint of the household is given by

$$A_t k_t^{\alpha} = c_t + k_{t+1} - (1-\delta) k_t, \tag{8}$$

$$(A_{t+1} - 1) = \rho_A (A_t - 1) + \sigma \eta_A \epsilon_{t+1}$$

$\epsilon_{t+1} \sim N(0,1)$
$\eta_A$ = standard deviation
$0 \leq \rho_A < 1$

The Lagrangian of the household's optimization problem takes the form

$$\mathcal{L} = E_0 \sum_{t=0}^{\infty} \beta^t \left\{ \frac{c_t^{1-\gamma}}{1-\gamma} + \lambda_t \left[ A_t k_t^{\alpha} - c_t - k_{t+1} + (1-\delta) k_t \right] \right\}$$

Eqm conditions are:

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

Let

$$y_t = c_t$$

$$x_t = \begin{bmatrix} x_t^1 \\ x_t^2 \end{bmatrix} = \begin{bmatrix} k_t \\ A_t \end{bmatrix}$$

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

NEOCLASSICAL_MODEL.M

```
%Define parameters
syms SIG DELTA ALFA BETTA RHO

%Define variables
syms c cp k kp a ap

%Write equations fi, i=1:3
f1 = c + kp - (1-DELTA) * k - a * k^ALFA;
f2 = c^(-SIG)-BETTA * cp^(-SIG)*(ap * ALFA * kp^(ALFA-1)+1-DELTA);
f3 = log(ap) - RHO * log(a);

%Create function f
f = [f1;f2;f3];

% Define the vector of controls, y, and states, x
x = [k a];
y = c;
xp = [kp ap];
yp = cp;
```

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 (7), it must be the case that:

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

and

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

To find those derivatives let's repeat equation (6)

$$\begin{aligned} F(x,\sigma) &\equiv E_t f(g(h(x,\sigma) + \eta\sigma\epsilon', \sigma), g(x,\sigma), h(x,\sigma) + \eta\sigma\epsilon', x) \\ &= 0. \end{aligned}$$

Taking derivative with respect to the scalar $\sigma$ we find:

$$\begin{aligned} F_\sigma(\bar{x},0) &= E_t f_{y'}[g_x(h_\sigma + \eta\epsilon') + g_\sigma] + f_y g_\sigma + f_{x'}(h_\sigma + \eta\epsilon') \\ &= f_{y'}[g_x h_\sigma + g_\sigma] + f_y g_\sigma + f_{x'} h_\sigma \end{aligned}$$

This is a system of $n$ equations. Then imposing

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

one can identify $g_\sigma$ and $h_\sigma$:

$$\begin{bmatrix} f_{y'} g_x + f_{x'} & f_{y'} + f_y \end{bmatrix} \begin{bmatrix} h_\sigma \\ g_\sigma \end{bmatrix} = 0$$

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

$$g_\sigma = 0.$$

To find $g_x$ and $h_x$ differentiate (6) with respect to $x$ to obtain the following system

$$F_x(\bar{x}, 0) \;=\; f_{y'} g_x h_x + f_y g_x + f_{x'} h_x + f_x$$

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 $n \times n_x$ quadratic equations in the $n \times n_x$ unknowns given by the elements of $g_x$ and $h_x$. Imposing

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

the above expression can be written as:

$$[f_{x'} \quad f_{y'}] \begin{bmatrix} I \\ g_x \end{bmatrix} h_x = -[f_x \quad f_y] \begin{bmatrix} I \\ g_x \end{bmatrix}$$

Let $A = [f_{x'} \quad f_{y'}]$ and $B = -[f_x \quad f_y]$. Let $P$ the matrix of eigenvectors of the matrix $h_x$ such that

$$h_x P = P\Lambda,$$

where $\Lambda$ is diagonal. (The matlab command eig.m produces such a decomposition). Finally, let

$$Z \equiv \begin{bmatrix} I \\ g_x \end{bmatrix} P.$$

Then the above expression can be written as:

$$AZ\Lambda = BZ,$$

We can then map the above problem into a generalized eigenvalue problem. For given $n \times n$ matrices $A$ and $B$, there exists a matrix $V$ and a diagonal matrix $D$ such that (using the matlab command [V,D]=eig(B,A)):

$$A[V_1 \quad V_2]\begin{bmatrix} D_{11} & \emptyset \\ \emptyset & D_{22} \end{bmatrix} = B[V_1 \quad V_2]$$

Assume, without of loss of generality, that all the eigenvalues of $D_{11}$ have modulus less than unity. Then we have:

$$\Lambda = D_{11}$$

$$\begin{bmatrix} I \\ g_x \end{bmatrix} P = V_1 = \begin{bmatrix} V_{11} \\ V_{12} \end{bmatrix}$$

So that

$$\boxed{h_x = V_{11} D_{11} V_{11}^{-1}}$$

and

$$\boxed{g_x = V_{12} V_{11}^{-1}}$$

What regularity conditions are needed to use this solution algorithm? We need that the matrix $P$ is invertible.

```
function [gx,hx] = gxhx(fy,fx,fyp,fxp,stake)

A = [-fxp -fyp];
B = [fx fy];

[V,D]=eig(B,A);

cs=find(abs(diag(D))<stake); %Find the columns corresponding
                             % to the roots of D with modulus < stake

ncs=numel(cs);

NX = size(fx,2);

if ncs<NX
    warning('no local equilirium')
elseif ncs>NX
    warning('eqm is indeterminate')
end

la = D(cs,cs);
```

```
Vs = V(:, cs);
P = Vs(1:ncs,:);

if rank(P)<ncs;
error('Invertibility condition violated')
end

hx =  P*la/P;
gx = Vs(ncs+1:end,:)/P;
```

## Second Moments

- $\widehat{x}_{t+1} = h_x \widehat{x}_t + \sigma \eta \epsilon_{t+1}$

- Covariance Matrix of $x_t$

  $\Sigma_x \equiv E \widehat{x}_t \widehat{x}_t'$

- Error term: $\Sigma_\epsilon \equiv \sigma^2 \eta \eta'$.

- We have

$$\Sigma_x = h_x \Sigma_x h_x' + \Sigma_\epsilon.$$

**Two numerical methods to compute $\Sigma_x$.**

- Method 1 Let $A$, $B$, and $C$ be matrices, such that $ABC$ exists. Then

$$\text{vec}(ABC) = (C' \otimes A) \cdot \text{vec}(B),$$

apply the vec operator to both sides of

$$\Sigma_x = h_x \Sigma_x h_x' + \Sigma_\epsilon,$$

$$\begin{aligned} \text{vec}(\Sigma_x) &= \text{vec}(h_x \Sigma_x h_x') + \text{vec}(\Sigma_\epsilon) \\ &= \mathcal{F}\ \text{vec}(\Sigma_x) + \text{vec}(\Sigma_\epsilon), \end{aligned}$$

where

$$\mathcal{F} = h_x \otimes h_x.$$

$$\text{vec}(\Sigma_x) = (I - \mathcal{F})^{-1}\text{vec}(\Sigma_\epsilon)$$

Method 2 The following iterative procedure, called doubling algorithm, may be faster than Method 1 in cases in which the number of state variables ($n_x$) is large.

$$\Sigma_{x,t+1} = h_{x,t}\Sigma_{x,t}h'_{x,t} + \Sigma_{\epsilon,t}$$

$$h_{x,t+1} = h_{x,t}h_{x,t}$$

$$\Sigma_{\epsilon,t+1} = h_{x,t}\Sigma_{\epsilon,t}h'_{x,t} + \Sigma_{\epsilon,t}$$

$$\Sigma_{x,0} = I$$

$$h_{x,0} = h_x$$

$$\Sigma_{\epsilon,0} = \Sigma_{\epsilon}$$

# Other second moments

Find $E\hat{x}_t\hat{x}'_{t-j}$ for $j > 0$.

$$\mu_t = \sigma\eta\epsilon_t.$$

$$\begin{aligned} E\hat{x}_t\hat{x}'_{t-j} &= E[h_x^j\hat{x}_{t-j} + \sum_{k=0}^{j-1} h_x^k\mu_{t-k}]\hat{x}'_{t-j} \\ &= h_x^j E\hat{x}_{t-j}\hat{x}'_{t-j} \\ &= h_x^j\Sigma_x \end{aligned}$$

$y_t$ is given by $y_t = \bar{y} + g_x(x_t - \bar{x})$, write as: $\hat{y}_t = g_x\hat{x}_t$. Then

$$\begin{aligned} E\hat{y}_t\hat{y}'_t &= Eg_x\hat{x}_t\hat{x}'_t g'_x \\ &= g_x[E\hat{x}_t\hat{x}'_t]g'_x \\ &= g_x\Sigma_x g'_x \end{aligned}$$

and, more generally,

$$\begin{aligned} E\widehat{y}_t\widehat{y}'_{t-j} &= g_x[E\widehat{x}_t\widehat{x}'_{t-j}]g'_x \\ &= g_x h_x^j \Sigma_x g'_x, \end{aligned}$$

for $j \geq 0$.

The Matlab routine mom.m posted on our website computes second moments.

## Impulse Response Functions

The impulse response of $z_t$ in period $t+j$ to an impulse in period $t$ is defined as:

$$IR(z_{t+j}) \equiv E_t z_{t+j} - E_{t-1} z_{t+j}$$

$$IR(\widehat{x}_t) \equiv E_0 \widehat{x}_t - E_{-1} \widehat{x}_t = h_x^t [x_0 - E_{-1} x_0] = h_x^t [\eta \sigma \epsilon_0] = h_x^t x; \quad t \geq 0.$$

$$IR(\widehat{y}_t) = g_x h_x^t x.$$

The Matlab routine ir.m posted on our website calculates impulse responses.

## Second-Order Approximation

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

$$\begin{aligned}
[g(x,\sigma)]^i &= [g(\bar{x},0)]^i + [g_x(\bar{x},0)]^i_a[(x-\bar{x})]_a + [g_\sigma(\bar{x},0)]^i[\sigma] \\
&+\frac{1}{2}[g_{xx}(\bar{x},0)]^i_{ab}[(x-\bar{x})]_a[(x-\bar{x})]_b \\
&+\frac{1}{2}[g_{x\sigma}(\bar{x},0)]^i_a[(x-\bar{x})]_a[\sigma] \\
&+\frac{1}{2}[g_{\sigma x}(\bar{x},0)]^i_a[(x-\bar{x})]_a[\sigma] \\
&+\frac{1}{2}[g_{\sigma\sigma}(\bar{x},0)]^i[\sigma][\sigma] \\
[h(x,\sigma)]^j &= [h(\bar{x},0)]^j + [h_x(\bar{x},0)]^j_a[(x-\bar{x})]_a + [h_\sigma(\bar{x},0)]^j[\sigma] \\
&+\frac{1}{2}[h_{xx}(\bar{x},0)]^j_{ab}[(x-\bar{x})]_a[(x-\bar{x})]_b \\
&+\frac{1}{2}[h_{x\sigma}(\bar{x},0)]^j_a[(x-\bar{x})]_a[\sigma] \\
&+\frac{1}{2}[h_{\sigma x}(\bar{x},0)]^j_a[(x-\bar{x})]_a[\sigma] \\
&+\frac{1}{2}[h_{\sigma\sigma}(\bar{x},0)]^j[\sigma][\sigma] \\
& i=1,\ldots,n_y, a,b=1,\ldots,n_x \quad j=1,\cdots,n_x.
\end{aligned}$$

The unknowns of this expansion are $[g_{xx}]^i_{ab}$, $[g_{x\sigma}]^i_a$, $[g_{\sigma x}]^i_a$, $[g_{\sigma\sigma}]^i$, $[h_{xx}]^j_{ab}$, $[h_{x\sigma}]^j_a$, $[h_{\sigma x}]^j_a$, $[h_{\sigma\sigma}]^j$

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 $h_{xx}(\bar{x}, 0)$.

$$
\begin{aligned}
&[F_{xx}(\bar{x},0)]^i_{jk} = \qquad (9)\\
&\left([f_{y'y'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{y'y}]^i_{\alpha\gamma}[g_x]^\gamma_k + [f_{y'x'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{y'x}]^i_{\alpha k}\right)[g_x]^\alpha_\beta[h_x]^\beta_j\\
&+[f_{y'}]^i_\alpha[g_{xx}]^\alpha_{\beta\delta}[h_x]^\delta_k[h_x]^\beta_j\\
&+[f_{y'}]^i_\alpha[g_x]^\alpha_\beta[h_{xx}]^\beta_{jk}\\
&+\left([f_{yy'}]^i_{\alpha\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{yy}]^i_{\alpha\gamma}[g_x]^\gamma_k + [f_{yx'}]^i_{\alpha\delta}[h_x]^\delta_k + [f_{yx}]^i_{\alpha k}\right)[g_x]^\alpha_j\\
&+[f_y]^i_\alpha[g_{xx}]^\alpha_{jk}\\
&+\left([f_{x'y'}]^i_{\beta\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{x'y}]^i_{\beta\gamma}[g_x]^\gamma_k + [f_{x'x'}]^i_{\beta\delta}[h_x]^\delta_k + [f_{x'x}]^i_{\beta k}\right)[h_x]^\beta_j\\
&+[f_{x'}]^i_\beta[h_{xx}]^\beta_{jk}\\
&+[f_{xy'}]^i_{j\gamma}[g_x]^\gamma_\delta[h_x]^\delta_k + [f_{xy}]^i_{j\gamma}[g_x]^\gamma_k + [f_{xx'}]^i_{j\delta}[h_x]^\delta_k + [f_{xx}]^i_{jk}\\
&= 0; \quad i = 1,\ldots n, \quad j,k,\beta,\delta = 1,\ldots n_x; \quad \alpha,\gamma = 1,\ldots n_y.
\end{aligned}
$$

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

$$
\begin{aligned}
[F_{\sigma\sigma}(\bar{x},0)]^i = \; & [f_{y'}]^i_\alpha [g_x]^\alpha_\beta [h_{\sigma\sigma}]^\beta \\
& + [f_{y'y'}]^i_{\alpha\gamma} [g_x]^\gamma_\delta [\eta]^\delta_\xi [g_x]^\alpha_\beta [\eta]^\beta_\phi [I]^\phi_\xi \\
& + [f_{y'x'}]^i_{\alpha\delta} [\eta]^\delta_\xi [g_x]^\alpha_\beta [\eta]^\beta_\phi [I]^\phi_\xi \\
& + [f_{y'}]^i_\alpha [g_{xx}]^\alpha_{\beta\delta} [\eta]^\delta_\xi [\eta]^\beta_\phi [I]^\phi_\xi \\
& + [f_{y'}]^i_\alpha [g_{\sigma\sigma}]^\alpha \\
& + [f_y]^i_\alpha [g_{\sigma\sigma}]^\alpha \\
& + [f_{x'}]^i_\beta [h_{\sigma\sigma}]^\beta \\
& + [f_{x'y'}]^i_{\beta\gamma} [g_x]^\gamma_\delta [\eta]^\delta_\xi [\eta]^\beta_\phi [I]^\phi_\xi \\
& + [f_{x'x'}]^i_{\beta\delta} [\eta]^\delta_\xi [\eta]^\beta_\phi [I]^\phi_\xi \\
= \; & 0; i = 1, \ldots, n; \quad \alpha, \gamma = 1, \ldots, n_y; \quad \beta, \delta = 1, \ldots, n_x; \quad \phi, \xi
\end{aligned}
$$

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

The cross derivatives $g_{x\sigma}$ and $h_{x\sigma}$ are equal to zero when evaluated at $(\bar{x}, 0)$. Use $F_{\sigma x}(\bar{x}, 0) = 0$ taking into account that all terms containing either $g_\sigma$ or $h_\sigma$ are zero at $(\bar{x}, 0)$.

$$\begin{aligned} [F_{\sigma x}(\bar{x}, 0)]^i_j &= [f_{y'}]^i_\alpha [g_x]^\alpha_\beta [h_{\sigma x}]^\beta_j + [f_{y'}]^i_\alpha [g_{\sigma x}]^\alpha_\gamma [h_x]^\gamma_j + [f_y]^i_\alpha [g_{\sigma x}]^\alpha_j + [f_{x'}]^i_\beta [h_{\sigma x}]^\beta_j \\ &= 0; \quad i = 1, \ldots n; \quad \alpha = 1, \ldots, n_y; \quad \beta, \gamma, j = 1, \ldots, n_x. \end{aligned} \tag{11}$$

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

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

These equations represent an important theoretical result. They show that in general, up to second-order, the coefficients of the policy function on the terms that are linear in the state vector do not depend on the size of the variance of the underlying shocks.

This result implies that the second-order approximation to the policy function of a stochastic model belonging to the general class given in equation (1) differs from that of its non-stochastic counterpart only in a constant term given by $\frac{1}{2}g_{\sigma\sigma}\sigma^2$ for the control vector $y_t$ and by $\frac{1}{2}h_{\sigma\sigma}\sigma^2$ for the state vector $x_t$.

As we stressed above what is simple about finding the second order terms of the functions $g$ and $h$ that it only involves linear operations. This suggests that one can postulate the problem in matrix form. This is what is done, numerically, in the programs gxx_hxx.m and gss_hss.m.

## Implementing Second-Order Methods

## Computing the derivatives of $f$

The Schmitt-Grohe and Uribe toolkit uses symbolic math software. The particular one we use is the MATLAB Toolbox Symbolic Math. Symbolic Math can handle analytical derivatives. We wrote programs, that compute the analytical derivatives of $f$ and evaluate them at the steady state.

- anal_deriv.m computes analytical derivatives of $f$

- num_eval.m evaluates the analytical derivatives of $f$.

You can check these programs out on the courses website.

NEOCLASSICAL_RUN.M

```
%Analytical Derivatives
[fx,fxp,fy,fyp,fypyp,fypy,fypxp,fypx,fyyp,fyy,fyxp,fyx,fxpyp,
fxpy,fxpxp,fxpx,fxyp,fxy,fxxp,fxx,f] = neoclassical_model;

%Steady State and Parameter Values
[SIG,DELTA,ALFA,BETTA,RHO,eta,c,cp,k,kp,a,ap,A,K,C]=neoclassical_model_ss;

%Obtain numerical derivatives of f
num_eval

%First-order approximation
[gx,hx] = gxhx(nfy,nfx,nfyp,nfxp)

%Second-order approximation
[gxx,hxx] = gxx_hxx(nfx,nfxp,nfy,nfyp,nfypyp,nfypy,nfypxp,nfypx,nfyyp,nfyy,nfyxp,.
                    nfyx,nfxpyp,nfxpy,nfxpxp,nfxpx,nfxyp,nfxy,nfxxp,nfxx,hx,gx)

[gss,hss] = gss_ss(nfx,nfxp,nfy,nfyp,nfypyp,nfypy,nfypxp,nfypx,nfyyp,nfyy,nfyxp,
                   nfyx,nfxpyp,nfxpy,nfxpxp,nfxpx,nfxyp,nfxy,nfxxp,nfxx,hx,gx,gxx,
```

## Constructing second-order accurate time series

- Iterate on: (Assume w.l.g. that the steady state value of $x$ is zero.)

$$x_1^i = h_x^i x_0 + x_0' h_{xx}^i x_0 + \sigma \eta^i \epsilon_1.$$

This may lead to explosive parts.

- Example from Kim, Kim, Schaumburg, and Sims (2005)

$$x_{t+1} = \rho x_t + \alpha x_t^2 + \epsilon_{t+1}; \quad 0 < \rho < 1$$

Note that this system has two steady states. One at $\bar{x} = 0$ and the other at $\bar{x} = (1-\rho)/\alpha$. Linearizing around $x = \bar{x}$ yields:

$$\widehat{x}_{t+1} = \rho \widehat{x}_t + \epsilon_{t+1}$$

Given the assumpmtion that $\rho < 1$, this system is stable in the neighborhood of $x = 0$. But the system is unstable in the neighborhood of $x = (1 - \rho)/\alpha$. Thus, once $x_t > (1 - \rho)/\alpha$, the system will diverge.

- Therefore, one should choose among the second-order accurate expansions the one that implies stability. A stable solution can be obtained by 'PRUNING' (a term coined by Chris Sims) out the the extraneous higher-order terms in each iteration by computing the projections of the second-order terms based on a first-order expansions.

  Let $x_t^f$ be the first-order iterate of $x_t$ given some $x_0$, that is, $x_0^f = x_0$, and let $x_t^s$ be the second-order iterate of $x_t$ given $x_0$ and set $x_0^s = 0$. Then define

$$x_{t+1}^f = h_x x_t^f + \sigma \eta \epsilon_{t+1}$$

  and

$$x_{t+1}^{is} = h_x^i x_t^s + \frac{1}{2} x_t^{f'} h_{xx}^i x_t^f + \frac{1}{2} h_{\sigma\sigma}^i \sigma^2$$

The program simu_2nd.m posted on our Perturbation website constructs such time series.

## How to compute unconditional first moments

- Note that the variance of either $x_t$ or $y_t$ are correct to second-order accuracy when computed from the first-order approximation alone. (That is, you can use the program mom.m to find the second-oder accurate second-moments.)

- Then, suppose that the unconditional expectation of $x_t$ exists, we can approximate this from its law of motion (assume without loss of generality that $x^{SS} = 0$:

$$x^i_{t+1} \approx h^i_x x_t + \frac{1}{2} x'_t h^i_{xx} x_t + \frac{1}{2} h^i_{\sigma\sigma}\sigma^2 + \sigma \eta^i \epsilon_{t+1}$$

  Note that the scalar

$$x'_t h^i_{xx} x_t = x'_t \otimes x'_t vec(h^i_{xx})$$

- and that $Ex'_t \otimes x'_t = vec(\Sigma_x)'$.

- We then have that:

$$\begin{aligned} E x_t' h_{xx}^i x_t &= E x_t' \otimes x_t' vec(h_{xx}^i) \\ &= (vec(\Sigma_x))' vec(h_{xx}^i) \end{aligned}$$

It follows that

$$E x_{t+1} = h_x E x_t + \begin{bmatrix} vec(h_{xx}^1)' \\ vec(h_{xx}^2)' \\ \dots \\ vec(h_{xx}^{n_x})' \end{bmatrix} vec(\Sigma_x)/2 + \frac{1}{2} h_{\sigma\sigma}\sigma^2$$

$$E x_t = (I - h_x)^{-1} \left( \begin{bmatrix} vec(h_{xx}^1)' \\ vec(h_{xx}^2)' \\ \dots \\ vec(h_{xx}^{n_x})' \end{bmatrix} vec(\Sigma_x)/2 + \frac{1}{2} h_{\sigma\sigma}\sigma^2 \right)$$

The program unconditional_mean.m computes unconditional means.

## Second-order accurate welfare approximations

Suppose we wish to evaluate the welfare consequences of alternative monetary policy arrangements. How should welfare be measured? Welfare is defined as the present dicounted value of lifetime utility. One could measure this either using the conditional expectation of lifetime utility or the unconditional expectation:

## Conditional Welfare Measures

- Let $V_t$ denote the time $t$ value of the present disounted value of lifetime utility. For example,

$$V_t \equiv E_t \sum_{j=0}^{\infty} \beta^j U(c_{t+j})$$

- Write $V_t$ recursively as:

$$V_t = U(c_t) + \beta E_t V_{t+1}$$

- Then add this expression the vector valued function $f$ and make $V_t$ an element of the vector of endogenous non-predetermined variables $y_t$.

- In this way, once one has computed second-order approximations to the function $g$, one has the second-order approximation of $V_t$ in hand.

- The idea is that we can express $V_t$ as some non-linear function of the state vector $x_t$ and of the scalar parameter $\sigma$, that is,

$$V_t = g^V(x_t, \sigma)$$

where, here the function $g^V$ is just one element of the function $g$ that we introduced earlier. It maps $R^{n_x} \times R^+$ into $R$. And by making $V_t$ one row of $f$, we can obtain the second-order accurate approximation to the function $g^V$.

- Clearly, the level of welfare will depend on the initial state being considered, that is, on the value of the state vector $x_t$. Depending on the particular question one would like to address, one may consider a number of relevant initial values for $x_t$ and average over those to obtain an average value of conditional welfare levels.

```
%Define parameters
syms SIG DELTA ALFA BETTA RHO

%Define variables
syms c cp k kp a ap vt vtp

%Write equations fi, i=1:4
f1 = c + kp - (1-DELTA) * k - a * k^ALFA;
f2 = c^(-SIG) - BETTA * cp^(-SIG) * (ap * ALFA * kp^(ALFA-1) + 1 - DELTA);
f3 = log(ap) - RHO * log(a);
f4 = -vt + (c^(1-SIG)-1)/(1-SIG) + BETTA * vtp;

%Create function f
f = [f1;f2;f3;f4];

% Define the vector of controls, y, and states, x
x = [k a];
y = [c vt];
xp = [kp ap];
yp = [cp vtp];
```

## Unconditional Welfare Measures

Just proceed as above to find $V_t$ and then use program unconditional_mean.m to find $E(V_t)$

## Welfare Cost Measures

One way of quantifying welfare differences across monetary policy regimes is to ask what percentage of the consumption stream associated with policy A households are willing to give to be as well off under policy A as under policy B.

Let $c_t^A$ the contingent plan for consumption associated with policy A and $c_t^B$ the contingent plan for consumption associated with policy B.

We can then implicitly define the welfare cost of adopting policy B rather than policy A as the value of $\lambda$ such that

$$V_t^B = E_t \sum_{j=0}^{\infty} \beta^j U(c_{t+j}^A(1-\lambda), h_{t+j})$$

Suppose the utility function is of the form:

$$U(c,h) = \frac{c^{1-\gamma}v(h) - 1}{1-\gamma}$$

So that

$$\begin{aligned} U(c(1-\lambda),h) &= \left(\frac{(1-\lambda)^{1-\gamma}c^{1-\gamma}v(h) - 1}{1-\gamma}\right) \\ &= (1-\lambda)^{1-\gamma}\left(\frac{c^{1-\gamma}v(h) - 1}{1-\gamma}\right) + \frac{(1-\lambda)^{1-\gamma} - 1}{1-\gamma} \end{aligned}$$

In this case we have:

$$V_t^B = (1-\lambda)^{1-\gamma}V_t^A + \frac{(1-\lambda)^{1-\gamma} - 1}{(1-\gamma)(1-\beta)} \tag{12}$$

Use

$$V_t^B = g^B(x_t, \sigma) \quad \text{and } V_t^A = g^A(x_t, \sigma)$$

to elimiate $V_t^A$ and $V_t^B$ from (12)

$$g^B(x_t, \sigma) = (1-\lambda)^{1-\gamma} g^A(x_t, \sigma) + \frac{(1-\lambda)^{1-\gamma} - 1}{(1-\gamma)(1-\beta)}$$

It follows that

$$\lambda = \Lambda(x_t, \sigma)$$

We want to find a second-order accurate approximation to $\Lambda$ around $(x_t, \sigma) = (x, 0)$

If $x_t = x$, only the derivatives of $\Lambda$ with respect to $\sigma$ have to be considered.

$$\lambda \approx \Lambda(x,0) + \Lambda_\sigma(x,0)\sigma + \frac{\Lambda_{\sigma\sigma}(x,0)}{2}\sigma^2.$$

Because the deterministic steady-state level of welfare is the same across all monetary policies we wish to compare, it follows that $\lambda$ vanishes at the point $(x_t, \sigma) = (x, 0)$.

$$\Lambda(x,0) = 0.$$

Totally differentiating with respect to $\sigma$, evaluating the result at $(x_0, \sigma) = (x, 0)$, and using the result derived in Schmitt-Grohé and Uribe (2004c) that the first derivatives of the policy functions with respect to $\sigma$ evaluated at $(x_0, \sigma) = (x, 0)$ are nil ($V_\sigma^A = V_\sigma^B = 0$), it follows immediately that

$$\Lambda_\sigma(x,0) = 0.$$

Totally differentiating twice with respect to $\sigma$, and evaluating the result at $(x_0, \sigma) = (x, 0)$ yields

$$\Lambda_{\sigma\sigma}(x, 0) = \frac{1-\beta}{c^{1-\sigma}v(h)}[V^A_{\sigma\sigma}(x, 0) - V^B_{\sigma\sigma}(x, 0)]$$

Thus, the conditional welfare cost measure is given by

$$\lambda \approx \left(\frac{1-\beta}{c^{1-\sigma}v(h)}\right)[V^A_{\sigma\sigma}(x, 0) - V^B_{\sigma\sigma}(x, 0)]\,\frac{\sigma^2}{2}. \tag{13}$$