# Perturbation Methods

David N. DeJong
University of Pittsburgh

Spring 2010

# Notation

Recall our generic representation of a DSGE model:

$$\Gamma\left(E_t z_{t+1}, z_t, \upsilon_{t+1}\right) = 0,$$

where $z_t$ is an $n \times 1$ vector of stationary variables, typically in the form of detrended levels, and $\upsilon_t$ is an $m \times 1$ vector of structural shocks.

# Notation, cont.

In what follows, it shall be convenient to re-express this as

$$E_t f\left(c_{t+1}, c_t, s_{t+1}, s_t\right) = 0,$$

where

- $c_t$ : $n_c \times 1$ vector of **control variables**
- $s_t$ : $n_s \times 1$ vector of **state variables**
- $n_c + n_s = n$

# Notation, cont.

Further, $s_t$ is decomposed as

$$s_t = \begin{bmatrix} s_t^1 \\ s_t^2 \end{bmatrix},$$

where

- $s_t^1$ : $\quad n_{s1} \times 1$ vector of **endogenous state variables** (e.g., physical captial)
- $s_t^2$ : $\quad n_{s2} \times 1$ vector of **exogenous state variables** (e.g., TFP)
- $n_{s2} = m, \qquad n_{s1} = n_s - n_{s2}.$

# Notation, cont.





Finally, $s_t^2$ evolves according to

$$s_{t+1}^2 = \Lambda s_t^2 + \sigma \widetilde{\eta} \varepsilon_{t+1},$$

where

- $\sigma$ is a scalar (specifically, a **perturbation parameter**)
- $\widetilde{\eta}$ : $m \times m$ VCV matrix
- $\varepsilon_{t+1}$ : i.i.d. with zero mean, VCV matrix $I$
- $\upsilon_{t+1} = \widetilde{\eta} \varepsilon_{t+1}$

# Alternative Model Representation



Thus our generic model is fully summarized as

$$\begin{aligned} E_t f\left(c_{t+1}, c_t, s_{t+1}, s_t\right) &= 0, \\ s_{t+1}^2 &= \Lambda s_t^2 + \sigma \widetilde{\eta} \varepsilon_{t+1}. \end{aligned}$$

# Representation of the Solution



Here, we seek a solution of the model of the form

$$\begin{aligned} c_t &= c\left(s_t, \sigma\right), \\ s_{t+1} &= s\left(s_t, \sigma\right) + \sigma \eta \varepsilon_{t+1}, \end{aligned}$$

where

$$\underset{n_s \times n_{s2}}{\eta} = \begin{bmatrix} \underset{n_{s1} \times n_{s2}}{0} \\ \underset{n_{s2} \times n_{s2}}{\widetilde{\eta}} \end{bmatrix}.$$

# Non-Stochastic Steady State

Define the **non-stochastic steady state** of the model as $(\bar{c}, \bar{s})$, such that

$$f(\bar{c}, \bar{c}, \bar{s}, \bar{s}) = 0.$$

It is also true that

$$\begin{aligned} \bar{c} &= c(\bar{s}, 0), \\ \bar{s} &= s(\bar{s}, 0), \end{aligned}$$

since for $\sigma = 0$,

$$E_t f() = f().$$

# Introduction to Perturbation Methods

The goal of **perturbation methods** is to construct Taylor Series approximations to

$$\begin{aligned} c_t &= c\left(s_t, \sigma\right), \\ s_{t+1} &= s\left(s_t, \sigma\right) + \sigma\eta\varepsilon_{t+1} \end{aligned}$$

around

$$\left(s, \sigma\right) = \left(\bar{s}, 0\right).$$

# Taylor's Theorem, 1-Dimensional x





Recall that for a generic $(k+1)$ −times differentiable function

$$y = f(x),$$

with $x$ a scalar, **Taylor's Theorem** states that

$$\begin{aligned} f(x) &= f(x_0) + (x - x_0) f'(x_0) + \frac{(x - x_0)^2}{2} f''(x_0) \\ &+ \cdots + \frac{(x - x_0)^k}{k!} f^{(k)}(x_0) + R_{k+1}(x), \end{aligned}$$

where

$$R_{k+1}(x) = \frac{(x - x_0)^{(k+1)}}{(k+1)!} f^{(k+1)}(\xi)$$

for some $\xi$ between $x$ and $x_0$.

# Taylor's Theorem, n-Dimensional x

For $n-$dimensional $x$, we have

$$\begin{aligned}
f(x) &= f\left(x^0\right) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}\left(x^0\right)\left(x_i - x_i^o\right) \\
&+ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \frac{\partial^2 f}{\partial x_i \partial x_j}\left(x^0\right)\left(x_i - x_i^o\right)\left(x_j - x_j^o\right) \\
&\vdots \\
&+ \frac{1}{k!}\sum_{i_1=1}^{n} \cdots \sum_{i_k=1}^{n} \frac{\partial^k f}{\partial x_{i_1 i} \cdots \partial x_{i_k}}\left(x^0\right)\left(x_{i_1} - x_{i_1}^o\right) \cdots \left(x_{x i_k} - x_{i_k}^o\right) \\
&+ O(\|x - x_0\|^{k+1}).
\end{aligned}$$

# Intro. to Perturbation, cont.

However, in this case the functions

$$\begin{aligned} c_t &= c(s_t, \sigma), \\ s_{t+1} &= s(s_t, \sigma) + \sigma \eta \varepsilon_{t+1} \end{aligned}$$

we seek to approximate are unknown. Thus we need further help from the **Implicit Function Theorem.**

# The Implicit Function Theorem

For $k-$times differentiable $H(x, y) : R^n \times R^m \longrightarrow R^m$, with

$$H(x_0, y_0) = 0,$$

and $H_y(x_0, y_0)$ non-singular, there is a unique function $h : R^n \longrightarrow R^m$ such that

$$y_0 = h(x_0),$$

and for $x$ near $x_0$,

$$H(x, h(x)) = 0.$$

Furthermore, $h(x)$ is $k-$times differentiable, and its derivatives can be computed by implicit differentiation of the identity $H(x, h, (x)) = 0$.

# Perturbation: The Basics (Judd, 1998)

Consider a problem of the form

$$f(x,\sigma) = 0,$$

with $(x,\sigma)$ scalars. We seek an approximation to the solution

$$x = x(\sigma),$$

given that $x(0)$ is known. The approximation is in the form of a Taylor Series expansion:

$$x \approx \overline{x} + x'(\sigma = 0)\,\sigma + \frac{1}{2}x''(\sigma = 0)\,\sigma^2 + \ldots$$

# The Basics, cont.

Substituting for $x$ using the solution we seek, we have

$$f\left(x\left(\sigma\right),\sigma\right)=0.$$

Then differentiating with respect to $\sigma$, we have by the Implicit Function Theorem

$$f_x\left(x\left(\sigma\right),\sigma\right)x'\left(\sigma\right)+f_\sigma\left(x\left(\sigma\right),\sigma\right)=0.$$

Then since $x\left(0\right)$ is known, and the functional form of $f\left(\right)$ is given, this yields

$$x'\left(0\right)=-\frac{f_\sigma\left(x\left(0\right),0\right)}{f_x\left(x\left(0\right),0\right)}.$$

# The Basics, cont.





Having calculated

$$x'(0) = -\frac{f_\sigma(x(0), 0)}{f_x(x(0), 0)},$$

the first-order approximation to $x = x(\sigma)$ we seek is given by

$$x \approx x(0) - \frac{f_\sigma(x(0), 0)}{f_x(x(0), 0)}\sigma.$$

**Key Observation:** Given $x(0)$, $x'(0)$ obtains linearly from the first difference of $f(x(\sigma), \sigma)$ with respect to $\sigma$.

# The Basics, cont.



To expand the approximation to second-order, we require an expression for $x''(0)$. Relying again on the Implicit Function Theorem, we can differentiate

$$f_x(x(\sigma),\sigma)\,x'(\sigma)+f_\sigma(x(\sigma),\sigma)=0$$

with respect to $\sigma$, yielding

$$\begin{aligned} & f_x(x(\sigma),\sigma)\,x''(\sigma)+f_{xx}(x(\sigma),\sigma)\left(x'(\sigma)\right)^2 \\ & +2f_{x\sigma}(x(\sigma),\sigma)\,x'(\sigma)+f_{\sigma\sigma}(x(\sigma),\sigma) \\ = \; & 0. \end{aligned}$$

# The Basics, cont.





Given the expression for $x'(0)$ calculated for the first-order approximation, and once again given that $x(0)$ is known, solving for $x''(0)$ yields

$$x''(0) = -\frac{f_{xx}(x(0),0)(x'(0))^2 + 2f_{x\sigma}(x(0),0)\,x'(0) + f_{\sigma\sigma}(x(0),0)}{f_x(x(0),0)}.$$

The second-order approximation we seek is then given by

$$x \approx x(0) - \frac{f_\sigma(x(0),0)}{f_x(x(0),0)}\sigma + \frac{1}{2}x''(0)\,\sigma^2.$$

**Key Observation:** Given $x(0)$, and $x'(0)$, $x''(0)$ obtains linearly from the second difference of $f(x(\sigma),\sigma)$ with respect to $\sigma$. Higher-order approximations obtain via straightforward recursion.

# Example

Consider the impact of a per-unit tax $\tau$ on the equilibrium quantity and price $(Q, P)$ of a generic good. Demand and supply for the good are given by

$$Q_D = \left(\frac{1}{P}\right)^{\alpha}, \qquad Q_S = (P - \tau)^{\beta}, \qquad \alpha, \beta > 0.$$

The impact we seek is in the form of the relationship between $\tau$ and equilibrium price $P$ :

$$P = P(\tau).$$

# Example, cont.

Establishing the equilibrium price as the solution to

$$\begin{aligned} f(P,\tau) &= Q_D - Q_S \\ &= \left(\frac{1}{P}\right)^{\alpha} - (P-\tau)^{\beta} \\ &= 0, \end{aligned}$$

note that for $\tau = 0$, $P = 1$. Substituting for $P$ using the form of the solution we seek, the problem is expressed as

$$f(P(\tau),\tau) = \left(\frac{1}{P(\tau)}\right)^{\alpha} - (P(\tau)-\tau)^{\beta} = 0.$$

# Example, cont.

**Exercise:** Applying the Implicit Function Theorem to

$$f\left(P\left(\tau\right),\tau\right)=\left(\frac{1}{P\left(\tau\right)}\right)^{\alpha}-\left(P\left(\tau\right)-\tau\right)^{\beta}=0,$$

derive the second-order approximation

$$P\left(\tau\right)\approx P\left(0\right)+P'\left(0\right)\tau+\frac{1}{2}P''\left(0\right)\tau^{2}.$$

# Approximating Solutions to DSGE Models

Returning to our generic DSGE model

$$\begin{aligned} E_t f\left(c_{t+1}, c_t, s_{t+1}, s_t\right) &= 0, \\ s^2_{t+1} &= \Lambda s^2_t + \sigma \widetilde{\eta} \varepsilon_{t+1}, \end{aligned}$$

our goal is to construct a *kth*-order Taylor Series approximation to the unknown solution

$$\begin{aligned} c_t &= c\left(s_t, \sigma\right), \\ s_{t+1} &= s\left(s_t, \sigma\right) + \sigma \eta \varepsilon_{t+1}. \end{aligned}$$

# Notation

Here, I will make explicit the construction of a second-order approximation, following Schmitt-Grohe and Uribe (2004). Let

$$[c_s]^i_a, \qquad [c_{ss}]^i_{ab}$$

denote the $(i, a)$ and $(i, ab)$ elements of the $n_c \times n_s$ and $n_c \times n_s^2$ matricies

$$\frac{\partial c(s_t, \sigma)}{\partial s_t}, \qquad \frac{\partial^2 c(s_t, \sigma)}{\partial s \partial s'},$$

evaluated at $(\bar{s}, 0)$.

# Notation, cont.

Also, let

$$
\begin{aligned}
\left[c_s\right]^i_a \left[s - \bar{s}\right]_a &= \sum_{a=1}^{n_s} \left[c_s\right]^i_a \left(s_a - \bar{s}_a\right), \\
\left[c_{ss}\right]^i_{ab} \left[s - \bar{s}\right]_a \left[s - \bar{s}\right]_b &= \sum_{a=1}^{n_s} \sum_{b=1}^{n_s} \left[c_{ss}\right]^i_{ab} \left(s_a - \bar{s}_a\right)\left(s_b - \bar{s}_b\right).
\end{aligned}
$$

etc.

# Approximating DSGEs, cont.





Then the approximation to $c(s_t, \sigma)$ we seek is of the form

$$
\begin{aligned}
[c(s_t,\sigma)]^i \quad = \quad & [\bar{c}]^i + [c_s]^i_a [s-\bar{s}]_a + [c_\sigma]^i \sigma \\
& + \frac{1}{2}[c_{ss}]^i_{ab}[s-\bar{s}]_a [s-\bar{s}]_b \\
& + \frac{1}{2}[c_{s\sigma}]^i_a [s-\bar{s}]_a \sigma \\
& + \frac{1}{2}[c_{\sigma s}]^i_a [s-\bar{s}]_a \sigma \\
& + \frac{1}{2}[c_{\sigma\sigma}]^i \sigma^2,
\end{aligned}
$$

$i = 1, ..., n_c;\ a, b = 1, ..., n_s.$

# Approximating DSGEs, cont.

Likewise, the approximation to $s(s_t, \sigma)$ we seek is of the form

$$
\begin{aligned}
[s(s_t,\sigma)]^j &= [\bar{s}]^j + [s_s]^j_a [s-\bar{s}]_a + [s_\sigma]^j \sigma \\
&+ \frac{1}{2}[s_{ss}]^j_{ab}[s-\bar{s}]_a[s-\bar{s}]_b \\
&+ \frac{1}{2}[s_{s\sigma}]^j_a [s-\bar{s}]_a \sigma \\
&+ \frac{1}{2}[s_{\sigma s}]^j_a [s-\bar{s}]_a \sigma \\
&+ \frac{1}{2}[s_{\sigma\sigma}]^j \sigma^2,
\end{aligned}
$$

$j = 1, ..., n_s;\ a, b = 1, ..., n_s.$

# Approximating DSGEs, cont.



Note that if

$$\left[c_s\right]_a^i, \qquad \left[c_{ss}\right]_{ab}^i,$$

etc. are in the form of elasticities, then $\left[s-\bar{s}\right]_a$, etc. represent logged deviations from steady states. That is, our approximations can accommodate both linear and log-linear model representations.

# Approximating DSGEs, cont.





To construct these approximations, we proceed by substituting for $(c_t, c_{t+1}, s_{t+1})$ in

$$E_t f\left(c_{t+1}, c_t, s_{t+1}, s_t\right) = 0,$$

using

$$\begin{aligned} c_t &= c\left(s_t, \sigma\right), \\ s_{t+1} &= s\left(s_t, \sigma\right) + \sigma \eta \varepsilon_{t+1}. \end{aligned}$$

Eliminating time subscripts, and denoting time-$(t+1)$ variables with primes, substitution yields

$$F\left(s, \sigma\right) \equiv E_t f\left(c\left(s\left(s, \sigma\right) + \sigma \eta \varepsilon\prime, \sigma\right), c\left(s, \sigma\right), s\left(s, \sigma\right) + \sigma \eta \varepsilon\prime, s\right) = 0.$$

# Linear Approximation

To construct our linear approximation, we use the Implicit Function Theorem to obtain

$$F_s(s,\sigma) = 0,$$
$$F_\sigma(s,\sigma) = 0,$$

where the first expression represents a set of $n \cdot n_s$ equalities, and the second a set of $n$ equalities.

# Linear Approximation, cont.

Denoting

$$\left[f_{c'}\right]^i_a \left[c_s\right]^a_b \left[s_s\right]^b_j = \sum_{a=1}^{n_c}\sum_{b=1}^{n_s} \frac{\partial f^i}{\partial c'^a}\frac{\partial c^a}{\partial s^b}\frac{\partial s^b}{\partial s^j},$$

etc., $F_s\left(\bar{s},0\right)=0$ is given by

$$\begin{aligned}\left[F_s\left(\bar{s},0\right)\right]^i_j &= \left[f_{c'}\right]^i_a \left[c_s\right]^a_b \left[s_s\right]^b_j + \left[f_c\right]^i_a \left[c_s\right]^a_j + \left[f_{s'}\right]^i_b \left[s_s\right]^b_j + \left[f_s\right]^i_j \\ &= 0,\end{aligned}$$

$i = 1, ..., n;\ j, b = 1, ..., n_s;\ a = 1, ..., n_c.$

# Linear Approximation, cont.

Since the derivatives of $f\left(\right)$ evaluated at $\left(\overline{c}, \overline{s}\right)$ are known,

$$\left[F_s\left(\overline{s}, 0\right)\right]^i_j = 0$$

is a system of $n \cdot n_s$ quadratic equations in the $n \cdot n_s$ unknown elements of $c_s\left(\right)$ and $s_s\left(\right)$.

This takes us halfway towards our linear approximations of $c\left(\right)$ and $s\left(\right)$. Below we shall discuss an alternative approach to obtaining $c_s\left(\right)$ and $s_s\left(\right)$.

# Linear Approximation, cont.





To complete the construction of our linear approximations, we use

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

which is given by

$$\begin{aligned}[F_\sigma(\bar{s}, 0)]^i &= [f_{c'}]^i_a [c_s]^a_b [s_\sigma]^b + [f_{c'}]^i_a [c_\sigma]^a + [f_c]^i_a [c_\sigma]^a + [f_{s'}]^i_b [s_\sigma]^b \\ &= 0,\end{aligned}$$

$i = 1, ..., n$; $a = 1, ..., n_c$; $b = 1, ..., n_s$. (Note: expressions involving $\varepsilon'$ are eliminated by application of the expectations operator.)

As these equations are linear and homogeneous in $\left([s_\sigma]^b, [c_\sigma]^a\right)$, these must be zero.

# Linear Approximation, cont.

Thus our first-order approximations are given by

$$\begin{aligned} [c(s_t,\sigma)]^i &= [\bar{c}]^i + [c_s]^i_a [s-\bar{s}]_a \\ [s(s_t,\sigma)]^j &= [\bar{s}]^j + [s_s]^j_a [s-\bar{s}]_a , \end{aligned}$$

$i = 1, ..., n; \; j = 1, ..., n_s.$

# Linear Approximation, cont.

As an aside, note that an alternative approach to obtaining

$$(c_s(), s_s())$$

involves the transformation of the linear model representation

$$x_{t+1} = Fx_t + Gv_{t+1}$$

(obtained, e.g., using Sims' method) into

$$\begin{aligned} c_t &= Cs_t, \\ s_{t+1} &= \Gamma s_t. \end{aligned}$$

# Linear Approximation, cont.





A simple method for doing so is as follows:

- Simulate $\{v_t\}_{t=1}^T$ from its known distribution ($T$ need not be large!).
- Using $x_0 = \overline{x}$ and $\{v_t\}_{t=1}^T$, simulate $\{x_t\}_{t=1}^T$ using

$$x_{t+1} = Fx_t + Gv_{t+1}.$$

- Divide $\{x_t\}_{t=1}^T$ into $\{c_t\}_{t=1}^T$, $\{s_t\}_{t=1}^T$, construct $y$ as the $T \times n_c$ matrix with $t^{th}$ row $c_t'$, and $X$ as the $T \times n_s$ matrix with $t^{th}$ row $s_t'$, and obtain

$$C' = (X'X)^{-1} X'y.$$

# Linear Approximation, cont.

Then to construct $s_{t+1} = \Gamma s_t$ :

- Divide $\{x_t\}_{t=1}^{T}$ into $\left\{s_t^1\right\}_{t=1}^{T}$, $\left\{s_t^2\right\}_{t=1}^{T}$, construct $y$ as the $T \times n_{s1}$ matrix with $t^{th}$ row $s_t^{1\prime}$, and $X$ as the $T \times n_s$ matrix with $t^{th}$ row $s_t'$, and obtain

$$\widetilde{\Gamma}' = \left(X'X\right)^{-1} X'y.$$

- Then recalling that

$$s_{t+1}^2 = \Lambda s_t^2 + \sigma \widetilde{\eta} \varepsilon_{t+1},$$

construct

$$\Gamma = \left[ \begin{array}{c} \widetilde{\Gamma} \\ 0\tilde{\ }\Lambda \end{array} \right], \qquad \underbrace{\widetilde{\Gamma}}_{n_{s1} \times n_s} \quad \underbrace{0\tilde{\ }\Lambda}_{n_{s2} \times n_s}$$

# Second-Order Approximation

To obtain second-order approximations to

$$\begin{aligned} c_t &= c\left(s_t, \sigma\right), \\ s_{t+1} &= s\left(s_t, \sigma\right) + \sigma\eta\varepsilon_{t+1}, \end{aligned}$$

we first differentiate

$$\left[F_s\left(\bar{s}, 0\right)\right]_j^i = 0$$

with respect to $s$ to identify

$$c_{ss}\left(\bar{s}, 0\right), \qquad s_{ss}\left(\bar{s}, 0\right).$$

# Second-Order Approximation, cont.



Next, we differentiate

$$F_{\sigma}\left(\bar{s},0\right)=0$$

with respect to $\sigma$ to identify

$$c_{\sigma\sigma}\left(\bar{s},0\right), \qquad s_{\sigma\sigma}\left(\bar{s},0\right).$$

Finally, we differentiate

$$F_{\sigma}\left(\bar{s},0\right)=0$$

with respect to $s$ to identify

$$c_{s\sigma}\left(\bar{s},0\right), \qquad s_{s\sigma}\left(\bar{s},0\right).$$

# Second-Order Approximation, cont.

Differentiating $[F_s(\bar{s},0)]^i_j = 0$ with respect to $s$:

$$[F_{ss}(\bar{s},0)]^i_{jk} =$$

$$
\begin{aligned}
& \left( [f_{c'c'}]^i_{a\gamma} [c_s]^\gamma_\delta [s_s]^\delta_k + [f_{c'c}]^i_{a\gamma} [c_s]^\gamma_k \right. \\
& \left. + [f_{c's'}]^i_{a\delta} [s_s]^\delta_k + [f_{c's}]^i_{ak} \right) [c_s]^a_b [s_s]^b_j \\
& + [f_{c'}]^i_a [c_{ss}]^a_{b\delta} [s_s]^\delta_k [s_s]^k_j \\
& + [f_{c'}]^i_a [c_s]^a_b [s_{ss}]^b_{jk} \\
& + \left( [f_{cc'}]^i_{a\gamma} [c_s]^\gamma_\delta [s_s]^\delta_k + [f_{cc}]^i_{a\gamma} [c_s]^\gamma_k + [f_{cs'}]^i_{a\delta} [s_s]^\delta_k + [f_{cs}]^i_{ak} \right) [c_s]^a_j \\
& + [f_c]^i_a [c_{ss}]^a_{jk} \\
& + \left( [f_{s'c'}]^i_{b\gamma} [c_s]^\gamma_\delta [s_s]^\delta_k + [f_{s'c}]^i_{b\gamma} [c_s]^\gamma_k + [f_{s's'}]^i_{b\delta} [s_s]^\delta_k + [f_{s's}]^i_{bk} \right) [s_s]^b_j \\
& + [f_{s'}]^i_b [s_{ss}]^b_{jk} \\
& + [f_{sc'}]^i_{j\gamma} [c_s]^\gamma_\delta [s_s]^\delta_k + [f_{sc}]^i_{j\gamma} [c_s]^\gamma_k + [f_{ss'}]^i_{j\delta} [s_s]^\delta_k + [f_{ss}]^i_{jk} \\
= \; & 0; \qquad i = 1, \ldots, n, \qquad j, k, b, \delta = 1, \ldots, n_s; \qquad a, \gamma = 1, \ldots, n_c.
\end{aligned}
$$

# Second-Order Approximation, cont.

Since we know the derivatives of $f$ as well as the first derivatives of $c$ and $s$ evaluated at $(c', c, s', s) = (\overline{c}, \overline{c}, \overline{s}, \overline{s})$, the above expression represents a system of $n \times n_s \times n_s$ linear equations in the $n \times n_s \times n_s$ unknowns given by the elements of $c_{ss}$ and $s_{ss}$.

# Second-Order Approximation, cont.

Differentiating $F_\sigma(\bar{s}, 0) = 0$ with respect to $\sigma$:

$$
\begin{aligned}
&[F_{\sigma\sigma}(\bar{s},0)]^i = \\
&\quad [f_{c'}]^i_a [c_s]^a_b [s_{\sigma\sigma}]^b \\
&\quad + [f_{c'c'}]^i_{a\gamma} [c_s]^\gamma_\delta [\eta]^\delta_\xi [c_s]^a_b [\eta]^b_\phi [I]^\phi_\xi \\
&\quad + [f_{c's'}]^i_{a\delta} [\eta]^\delta_\xi [c_s]^a_b [\eta]^b_\phi [I]^\phi_\xi \\
&\quad + [f_{c'}]^i_a [c_{ss}]^a_{b\delta} [\eta]^\delta_\xi [\eta]^b_\phi [I]^\phi_\xi \\
&\quad + [f_{c'}]^i_a [c_{\sigma\sigma}]^a \\
&\quad + [f_c]^i_a [c_{\sigma\sigma}]^a \\
&\quad + [f_{s'}]^i_b [s_{\sigma\sigma}]^b \\
&\quad + [f_{s'c'}]^i_{b\gamma} [c_s]^\gamma_\delta [\eta]^\delta_\xi [\eta]^b_\phi [I]^\phi_\xi \\
&\quad + [f_{s's'}]^i_{b\delta} [\eta]^\delta_\xi [\eta]^b_\phi [I]^\phi_\xi \\
&= 0; \qquad i = 1, \ldots, n \qquad a, \gamma = 1, \ldots, n_c; \qquad b, \delta = 1, \ldots, n_s; \\
\phi, \xi &= 1, \ldots, n_c
\end{aligned}
$$

# Second-Order Approximation, cont.

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

# Second-Order Approximation, cont.

Differentiating $F_\sigma(\bar{s}, 0) = 0$ with respect to $s$, taking into account that all terms containing either $c_\sigma$ or $s_\sigma$ are zero at $(\bar{s}, 0)$, we have $[F_{\sigma s}(\bar{s}, 0)]^i_j =$

$$\begin{aligned}
[F_{\sigma s}(\bar{s}, 0)]^i_j &= [f_{c'}]^i_a [c_s]^a_b [s_{\sigma s}]^b_j + [f_{c'}]^i_a [c_{\sigma s}]^a_\gamma [s_s]^\gamma_j \\
&\quad + [f_c]^i_a [c_{\sigma s}]^a_j + [f_{s'}]^i_b [s_{\sigma s}]^b_j \\
&= 0; \qquad i = 1, \ldots n; \qquad a = 1, \ldots, n_c, \qquad b, \gamma, j = 1, \ldots, n_s.
\end{aligned}$$

This is a system of $n \times n_s$ equations in the $n \times n_s$ unknowns given by the elements of $c_{\sigma s}$ and $s_{\sigma s}$.
But the system is homogeneous in the unknowns, thus

$$c_{\sigma s} = 0 \qquad s_{\sigma s} = 0$$

# Example: One Tree Model

Consider a version of the one-tree model featuring only dividends as a stochastic process. Recalling that in equilibrium

$$c_t = d_t \qquad \forall t,$$

the model is given by

$$\begin{aligned} u'\left(d_t\right) p_t &= \beta E_t \left(u'\left(d_{t+1}\right)\left(p_{t+1} + d_{t+1}\right)\right) \\ d_{t+1} &= \left(1-\rho\right)\overline{d} + \rho d_t + \sigma \varepsilon_{t+1} \qquad \varepsilon_{t+1} \sim N\left(0, \sigma_\varepsilon^2\right). \end{aligned}$$

# One Tree Model, cont.

Note in this case that the state is comprised exclusively as the exogenous dividend process

$$s_t = d_t - \overline{d},$$

implying

$$\eta = \widetilde{\eta} = \sigma_\varepsilon.$$

Moreover, this implies that the state-transition equation need not be constructed, but is given directly as

$$\begin{aligned} s_{t+1} &= s\left(s_t, \sigma\right) + \sigma \eta \varepsilon_{t+1} \\ &= \left(1-\rho\right)\overline{d} + \rho d_t + \sigma \sigma_\varepsilon \varepsilon_{t+1}. \end{aligned}$$

# One Tree Model, cont.





Thus in this case we merely require the approximation of the policy function

$$c_t = c\left(s_t, \sigma\right),$$

where the controls are comprised exclusively as

$$c_t = p_t,$$

with steady state

$$\overline{p} = \frac{1}{r}\overline{d}.$$

# One Tree Model, cont.





The approximation we seek is of the form

$$\begin{aligned}
[c(s_t,\sigma)] &= \overline{p} + [c_s]\left(d-\overline{d}\right) + [c_\sigma]\,\sigma \\
&\quad + \frac{1}{2}[c_{ss}]\left(d-\overline{d}\right)^2 \\
&\quad + \frac{1}{2}[c_{\sigma\sigma}]\,\sigma^2,
\end{aligned}$$

since it is known that

$$[c_{s\sigma}] = [c_{\sigma s}] = 0.$$

# One Tree Model, cont.

In terms of the representation

$$F(s,\sigma) \equiv E_t f\left(c(s(s,\sigma)+\sigma\eta\varepsilon\prime,\sigma), c(s,\sigma), s(s,\sigma)+\sigma\eta\varepsilon\prime, s\right) = 0,$$

under the redefinition

$$c(s,\sigma) \equiv p(d,\sigma), \qquad s(s_t,\sigma) = (1-\rho)\,\overline{d} + \rho d_t,$$

the model is given by $F(d,\sigma) =$

$$\begin{aligned} & E_t \left[ \begin{array}{c} u'(d)\,p(d,\sigma) - \beta u'\left((1-\rho)\,\overline{d} + \rho d + \sigma\sigma_\varepsilon\varepsilon'\right)\cdot \\ \left(p\left((1-\rho)\,\overline{d} + \rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma\right) + (1-\rho)\,\overline{d} + \rho d + \sigma\sigma_\varepsilon\varepsilon'\right) \end{array} \right] \\ = \; & 0. \end{aligned}$$

# One Tree Model, cont.





Hereafter, to ease notation, we shall drop the appearance of the constant term $(1-\rho)\overline{d}$ from all expressions in $F(d,\sigma)$ involving $d_{t+1}$ expressed as a function of $d_t$.

Thus the model is expressed as

$$
\begin{aligned}
& F(d,\sigma) = \\
& \quad E_t \left[ \begin{array}{c} u'(d)\, p(d,\sigma) \\ -\beta u'(\rho d + \sigma\sigma_\varepsilon \varepsilon') \cdot \left( p(\rho d + \sigma\sigma_\varepsilon \varepsilon', \sigma) + \rho d + \sigma\sigma_\varepsilon \varepsilon' \right) \end{array} \right] \\
& = \quad 0.
\end{aligned}
$$

# One Tree Model, First-Order Approximation

Differentiating $F(d,\sigma)$ with respect to $d$, we obtain
$F_d(d,\sigma) =$

$$E_t \left[ \begin{array}{c} u''(d)\, p(d,\sigma) + u'(d)\, p_d(d,\sigma) \\ -\beta u''(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \rho \cdot \\ (p(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \rho d + \sigma\sigma_\varepsilon\varepsilon') \\ -\beta u'(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot (p_d(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) \cdot \rho + \rho) \end{array} \right]$$

$$= 0.$$

# First-Order Approx., cont.

Applying the expectations operator, and evaluating at $\left(\overline{d}, \overline{p}, \sigma = 0\right)$, we obtain

$$\begin{aligned} & u''\left(\overline{d}\right) p\left(\overline{d}, 0\right) + u'\left(\overline{d}\right) p_d\left(\overline{d}, 0\right) - \beta u''\left(\overline{d}\right) \rho \cdot \left(\overline{p} + \overline{d}\right) \\ & -\beta u'\left(\overline{d}\right) \left(p_d\left(\overline{d}, 0\right) \rho + \rho\right) \\ = \;& 0. \end{aligned}$$

Note in the linear utility case,

$$u'\left(\overline{d}\right) = 1, \qquad u''\left(\overline{d}\right) = 0,$$

and thus

$$p_d\left(\overline{d}, 0\right) - \beta\left(p_d\left(\overline{d}, 0\right) \rho + \rho\right) = 0,$$

or

$$p_d\left(\overline{d}, 0\right) = \frac{\rho\beta}{1 - \rho\beta}.$$

# First-Order Approx., cont.

For

$$u'(d) = d^{-\gamma},$$

solving for $p_d\left(\overline{d}, 0\right)$ yields

$$\begin{aligned}
p_d\left(\overline{d}, 0\right) &= \frac{-u''\left(\overline{d}\right) p\left(\overline{d}, 0\right) + \beta u''\left(\overline{d}\right) \rho \cdot \left(\overline{p} + \overline{d}\right) + \rho\beta u'\left(\overline{d}\right)}{u'\left(\overline{d}\right)(1-\rho\beta)} \\
&= \left(\frac{-u''\left(\overline{d}\right)\overline{d}}{u'\left(\overline{d}\right)}\right)\left(\frac{1/r - \beta\rho\cdot(1/r+1) - \rho\beta u'\left(\overline{d}\right)/u''\left(\overline{d}\right)\overline{d}}{(1-\rho\beta)}\right) \\
&= \gamma\left(\frac{1/r - \beta\rho\cdot(1/r+1) + \rho\beta\gamma^{-1}}{(1-\rho\beta)}\right) \\
&= \gamma\left(\frac{1/r(1-\rho\beta) + \rho\beta\left(\gamma^{-1}-1\right)}{(1-\rho\beta)}\right).
\end{aligned}$$

# First-Order Approx., cont.

Next, differentiating $F(d,\sigma)$ with respect to $\sigma$, we obtain

$$
\begin{aligned}
& F_\sigma(d,\sigma) = \\
& E_t \left[ \begin{array}{c} u'(d)\, p_\sigma(d,\sigma) - \beta u''(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \sigma_\varepsilon\varepsilon' \cdot \\ (p(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \rho d + \sigma\sigma_\varepsilon\varepsilon') \\ -\beta u'(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \\ (p_d(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) \cdot \sigma_\varepsilon\varepsilon' + p_\sigma(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \sigma_\varepsilon\varepsilon') \end{array} \right] \\
= \; & 0.
\end{aligned}
$$

# First-Order Approx., cont.

Applying the expectations operator, and evaluating at $\left(\overline{d}, \overline{p}, \sigma = 0\right)$, we obtain

$$u'\left(\overline{d}\right) p_\sigma\left(\overline{d}, 0\right) - \beta u'\left(\overline{d}\right) p_\sigma\left(\overline{d}, 0\right) = 0.$$

Thus

$$p_\sigma\left(\overline{d}, 0\right) = 0.$$

# First-Order Approx., cont.

Thus our linear approximation of the policy function is given by

$$[c(s_t, \sigma)] = \overline{p} + \gamma \left( \frac{1/r(1-\rho\beta) + \rho\beta(\gamma^{-1} - 1)}{(1-\rho\beta)} \right) (d - \overline{d}) .$$

# Second-Order Approximation

Pursuing the second-order approximation, differentiating $F_d(d,\sigma)$ with respect to $d$, we obtain
$F_{dd}(d,\sigma) =$

$$
\begin{aligned}
& E_t \left[ \begin{array}{c}
u'''(d)\, p(d,\sigma) + 2u''(d)\, p_d(d,\sigma) \\
+u'(d)\, p_{dd}(d,\sigma) \\
-\beta u'''(\rho d + \sigma\sigma_\varepsilon \varepsilon') \cdot \rho^2 \cdot \\
(p(\rho d + \sigma\sigma_\varepsilon \varepsilon', \sigma) + \rho d + \sigma\sigma_\varepsilon \varepsilon') \\
-2\beta u''(\rho d + \sigma\sigma_\varepsilon \varepsilon') \cdot \rho \cdot \\
(p_d(\rho d + \sigma\sigma_\varepsilon \varepsilon', \sigma) + \rho) \\
-\beta u'(\rho d + \sigma\sigma_\varepsilon \varepsilon') \cdot \left(p_{dd}(\rho d + \sigma\sigma_\varepsilon \varepsilon', \sigma) \cdot \rho^2\right)
\end{array} \right] \\
= \quad & 0.
\end{aligned}
$$

# Second-Order Approx., cont.

Applying the expectations operator and evaluating at $\left(\overline{d}, \overline{p}, \sigma = 0\right)$, we obtain

$$\begin{aligned} & u'''\left(\overline{d}\right) p\left(\overline{d}, 0\right) + 2u''\left(\overline{d}\right) p_d\left(\overline{d}, 0\right) \\ & + u'\left(\overline{d}\right) p_{dd}\left(\overline{d}, 0\right) - \beta u'''\left(\overline{d}\right) \cdot \rho^2 \cdot \left(p\left(\overline{d}, 0\right) + \overline{d}\right) \\ & - 2\beta u''\left(\overline{d}\right) \cdot \rho \cdot \left(p_d\left(\overline{d}, 0\right) + 1\right) \\ & - \beta u'\left(\overline{d}\right) p_{dd}\left(\overline{d}, 0\right) \cdot \rho^2 \\ = \; & 0. \end{aligned}$$

# Second-Order Approx., cont.

Solving for $p_{dd}\left(\bar{d},0\right)$, we obtain

$$p_{dd}\left(\bar{d},0\right)=\frac{-a-2b+c+2d}{u'\left(\bar{d}\right)\left(1-\rho\beta\right)},$$

where

$$\begin{aligned} a &= u'''\left(\bar{d}\right)p\left(\bar{d},0\right) \\ b &= u''\left(\bar{d}\right)p_d\left(\bar{d},0\right) \\ c &= \rho^2\beta u'''\left(\bar{d}\right)\cdot\left(p\left(\bar{d},0\right)+\bar{d}\right) \\ d &= \rho^2\beta u''\left(\bar{d}\right)\left(p_d\left(\bar{d},0\right)+1\right). \end{aligned}$$

# Second-Order Approx., cont.

Next, differentiating $F_\sigma(d,\sigma)$ with respect to $\sigma$, we obtain $F_{\sigma\sigma}(d,\sigma) =$

$$E_t \begin{bmatrix} u'(d)\, p_{\sigma\sigma}(d,\sigma) \\ -\beta u'''(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \sigma_\varepsilon^2 \varepsilon'^2 \cdot \\ (p(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \rho d + \sigma\sigma_\varepsilon\varepsilon') \\ -\beta u''(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \sigma_\varepsilon\varepsilon' \cdot \\ (p_d(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) \cdot \sigma_\varepsilon\varepsilon' + p_\sigma(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \sigma_\varepsilon\varepsilon') \\ -\beta u''(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \sigma_\varepsilon\varepsilon' \cdot \\ (p_d(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) \cdot \sigma_\varepsilon\varepsilon' + p_\sigma(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + \sigma_\varepsilon\varepsilon') \\ -\beta u'(\rho d + \sigma\sigma_\varepsilon\varepsilon') \cdot \\ (p_{dd}(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) \cdot \sigma_\varepsilon^2 \varepsilon'^2 + p_\sigma(\rho d + \sigma\sigma_\varepsilon\varepsilon', \sigma) + p_{\sigma\sigma}(d,\sigma)) \end{bmatrix}$$

$$= \quad 0.$$

Note: I have exploited the fact that $p_{d\sigma}() = p_{\sigma d}() = 0$.

# Second-Order Approx., cont.

Applying the expectations operator, evaluating at $\left(\overline{d}, \overline{p}, \sigma = 0\right)$, and recalling $p_\sigma\left(\overline{d}, 0\right) = 0$, we obtain

$$\begin{aligned} & u'\left(\overline{d}\right) p_{\sigma\sigma}\left(\overline{d}, 0\right) \\ & -\beta u'''\left(\overline{d}\right) \cdot \sigma_\varepsilon^2 \cdot \left(\overline{p} + \overline{d}\right) \\ & -\beta u'\left(\overline{d}\right) \cdot \\ & \left(p_{dd}\left(\overline{d}, 0\right) \cdot \sigma_\varepsilon^2 + p_{\sigma\sigma}\left(\overline{d}, 0\right)\right) \\ = \; & 0. \end{aligned}$$

# Second-Order Approx., cont.





Given the expression for $p_{dd}\left(\overline{d},0\right)$, we solve for $p_{\sigma\sigma}\left(\overline{d},0\right)$ as

$$p_{\sigma\sigma}\left(\overline{d},0\right)=\sigma_{\varepsilon}^{2}\frac{\beta}{1-\beta}\left(\frac{u'''\left(\overline{d}\right)}{u'\left(\overline{d}\right)}+p_{dd}\left(\overline{d},0\right)\right).$$

# Second-Order Approx., cont.





Thus our quadratic approximation of the policy function is given by

$$\begin{aligned}
\left[c\left(s_t,\sigma\right)\right] &= \overline{p}+\gamma\left(\frac{1/r\left(1-\rho\beta\right)+\rho\beta\left(\gamma-1\right)}{\left(1-\rho\beta\right)}\right)\left(d-\overline{d}\right) \\
&\quad +\frac{1}{2}p_{dd}\left(\overline{d},0\right)\left(d-\overline{d}\right)^2 \\
&\quad +\frac{1}{2}p_{\sigma\sigma}\left(\overline{d},0\right)\sigma^2
\end{aligned}$$

# Code

To adapt the GAUSS version of the code developed by Schmitt-Grohe/Uribe to a particular DSGE model, the user must provide:

- Preamble identifying the dimensionality of the model, establishing the parameter vector, and mapping the parameter vector into $\eta$
- An `src` file with procedures that return steady state values as a function of the parameters; a seperate procedure for each model equation that evaluates the equation at the steady state; and the matricies $(C, D)$ needed as input for Sims' solution procedure.

# Preamble for the RBC Model

```
neulers = 1; // # of euler equations included in the model

nexstates = 1; // # of structural shocks

nendstates = 1; // # of endogenous state variables

ncontrols = 5; // # of control variables

nstates = nendstates+nexstates; // total # of state variables

nvars = ncontrols+nstates; // # of variables included in the model

xbar = 0; // will contain ss values

xstar = 0; // ss values extended

approx = 2; // Order of approximation desired

procvec=0; // Define a system of equations as a vector of procedures

// establish parameters: alpha, beta, delta, rho, sigeps, phi, psi

let p[7,1] = 0.24 0.99 0.025 0.78, 0.0067, 1.5, 0.35;
```

# Preamble, cont.

```
eta = zeros(nstates,nexstates) ;

sigma = 1;           // perturbation parameter

vcvmat = zeros(nexstates,nexstates);      // VCV matrix of exogenous innovations

vcvmat[1,1] = p[5]^2;

sqrtvcvmat = chol(vcvmat)';

eta[1:nexstates,.] = sqrtvcvmat;
```

# Example Bits of the SRC file

```
proc(1)=Sys3(x);
 local fx, a, alag, k, klag, y, ylag, c, clag, i, ilag, n, nlag, l, llag,
      alp, bet, del, rh, sige, ph, ps, cfac, lfac;
           alag = x[8]; klag = x[9]; ylag = x[10]; clag = x[11]; ilag = x[12]; nlag = x[13]; llag = x[14];
           a = x[1]; k = x[2]; y = x[3]; c = x[4]; i = x[5]; n = x[6]; l = x[7];
           alp = p[1]; bet = p[2]; del = p[3]; rh = p[4]; sige= p[5]; ph = p[6]; ps = p[7];
          cfac = ps*(1-ph)-1; lfac = (1-ps)*(1-ph);
           fx = y - a - alp*k - (1 - alp)*n;
 retp(fx);
endp;
```

# SRC Bits, cont.

```
proc(1)=Sys(x);
 procvec=&Sys1~&Sys2~&Sys3~&Sys4~&Sys5~&Sys6~&Sys7;
          // Points to procedures for calculating Hessian matrices
 retp(Sys1(x)|Sys2(x)|Sys3(x)|Sys4(x)|Sys5(x)|Sys6(x)|Sys7(x));
          // Model equations are evaluated at steady states and stacked
endp;
```

# Key Elements of the Body Code

- First and second derivatives of $F(s, \sigma)$
- First-order model approximation ala Sims

Mapping $x_{t+1} = Fx_t + Gv_{t+1}$ into

$$\begin{aligned} c_t &= Cs_t, \\ s_{t+1} &= \Gamma s_t. \end{aligned}$$

# Body Code, cont.

- Differentiating $[F_s(\bar{s},0)]^i_j = 0$ with respect to $s$, solving for
$$c_{ss}(\bar{s},0), \qquad s_{ss}(\bar{s},0).$$
- Differentiating $F_\sigma(\bar{s},0) = 0$ with respect to $\sigma$, solving for
$$c_{\sigma\sigma}(\bar{s},0), \qquad s_{\sigma\sigma}(\bar{s},0).$$

# Final Step: Construct Policy Functions

Having obtained $(c_s, s_s, c_{ss}, s_{ss}, c_{\sigma\sigma}, s_{\sigma\sigma})$ , we map these into the coefficients of the second-order Taylor Series approximations of $c(s, \sigma)$ , $s(s, \sigma)$ .

# Final Step

```
proc spc_quad_of_s(s,sig);
 // constructs policy function using quadratic approximation
 // inputs are levels of s; outputs are levels of sp, c
 local stilde,s2tilde,conttilde,sptilde,cont,sp,ii;
 stilde = ln(s./ss[1:nstates]);
 s2tilde = stilde*stilde';
 conttilde=zeros(ncontrols,1);
 sptilde=zeros(nstates,1);
 ii=1; do while ii<=nstates;
      conttilde = conttilde + 0.5*getMatrix(gxx[.,.,.],ii)*s2tilde[ii,.]';
      sptilde = sptilde + 0.5*getMatrix(hss[.,.,.],ii)*s2tilde[ii,.]';
     ii=ii+1;endo;
 conttilde = gx*stilde + conttilde + 0.5*gss*sig^2;
 sptilde = hx*stilde + sptilde + 0.5*hss*sig^2;
 cont = ss[nstates+1:nvars].*exp(conttilde);
 sp = ss[1:nstates].*exp(sptilde);
 retp(sp|cont);
endp;
```