Perturbation methods: Solving DSGE models with Dynare

Randall Romero

The Ohio State University

First draft: April 5, 2013
Last revised: October 7, 2014
Dynare is a user-friendly software platform useful for solving dynamic economic models.

Dynare relies on perturbation methods to find solution to models.

The two objectives of this presentation are:

- to explain the logic behind the use of perturbation methods; and
- to provide a short introduction to the use of Dynare.

For illustration, we solve the Solow model and the Ramsey model.
Outline

1 Perturbation Methods
   - Some math results
   - Perturbation in dynamic models
   - An example: The Solow model
   - Another example: The Ramsey model

2 Some Dynare basics
   - About Dynare
   - About model types

3 Examples of models in Dynare
   - A deterministic Solow model
   - A stochastic Ramsey model
Perturbation methods

- Dynare approach to solving dynamic models is based on perturbation methods.
- These methods are based on Taylor’s expansions and the implicit function theorem.
- In this section we review the mathematics necessary to understand perturbation methods.
Fixed point of a function

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a function. A point $x^* \in \mathbb{R}^n$ is called a fixed point of $f$ if it satisfies

$$f(x^*) = x^*$$

Example: $f(x) = 2\sqrt{x}$ has two fixed points: $x^* = 0$ and $x^* = 4$. 

$$y = f(x) = 2x^{1/2}$$
Some definitions

Let $f$ be a function, $f : \mathbb{R}^n \rightarrow \mathbb{R}$, where $\mathbf{x} = (x_1 \cdots x_n)'$. We denote the first partial derivatives of $f(\mathbf{x})$ by

$$f_i(\mathbf{x}) = \frac{\partial f(\mathbf{x})}{\partial x_i} \quad \text{and} \quad \nabla f(\mathbf{x}) = \begin{pmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{pmatrix}$$

and the Hessian matrix of $f(\mathbf{x})$ by

$$H(\mathbf{x}) = \begin{bmatrix} f_{11}(\mathbf{x}) & f_{12}(\mathbf{x}) & \cdots & f_{1n}(\mathbf{x}) \\ f_{21}(\mathbf{x}) & f_{22}(\mathbf{x}) & \cdots & f_{2n}(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1}(\mathbf{x}) & f_{n2}(\mathbf{x}) & \cdots & f_{nn}(\mathbf{x}) \end{bmatrix}$$
Some notation

Let $f$ be a function, $f : \mathbb{R}^n \to \mathbb{R}^m$:

$$f(x) = \begin{pmatrix} f^1(x) \\ \vdots \\ f^m(x) \end{pmatrix}$$

We denote the Jacobian of $f(x)$ by

$$J(x) = \begin{bmatrix} f^1_1(x) & f^1_2(x) & \cdots & f^1_n(x) \\ f^2_1(x) & f^2_2(x) & \cdots & f^2_n(x) \\ \vdots & \vdots & \ddots & \vdots \\ f^m_1(x) & f^m_2(x) & \cdots & f^m_n(x) \end{bmatrix} = \begin{bmatrix} \nabla f^1(x) \\ \nabla f^2(x) \\ \vdots \\ \nabla f^m(x) \end{bmatrix}$$
Let $g(x, y)$ be a function of vectors $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$, such that $g : \mathbb{R}^{n+m} \to \mathbb{R}^m$.

Think of $g$ as a system of $m$ nonlinear equations on $m$ endogenous variables $y$ and $n$ exogenous variables $x$.

The partial Jacobians $Dg_x$ and $Dg_y$ form a partition of the Jacobian:

$$J(x, y) = \begin{bmatrix} Dg_x & Dg_y \end{bmatrix} = \begin{bmatrix} g^1_x & g^1_y & \cdots & g^m_x & g^m_y \\ \frac{\partial g^1}{\partial x_1} & \frac{\partial g^1}{\partial y_1} & \cdots & \frac{\partial g^m}{\partial x_1} & \frac{\partial g^m}{\partial y_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial g^1}{\partial x_n} & \frac{\partial g^1}{\partial y_n} & \cdots & \frac{\partial g^m}{\partial x_n} & \frac{\partial g^m}{\partial y_n} \\ \frac{\partial g^1}{\partial y_1} & \frac{\partial g^1}{\partial y_2} & \cdots & \frac{\partial g^m}{\partial y_1} & \frac{\partial g^m}{\partial y_2} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{\partial g^1}{\partial y_m} & \frac{\partial g^1}{\partial y_m} & \cdots & \frac{\partial g^m}{\partial y_m} & \frac{\partial g^m}{\partial y_m} \end{bmatrix}$$
Taylor’s Theorem

Taylor’s theorem, \( \mathbb{R} \) case

Let \( f : [a, b] \rightarrow \mathbb{R} \) be a \( n + 1 \) times continuously differentiable function on (a,b), let \( \bar{x} \) be a point in (a,b). Then

\[
f(\bar{x} + b) = f(\bar{x}) + f^{(1)}(\bar{x})b + f^{(2)}(\bar{x})\frac{b^2}{2} + \ldots + f^{(n)}(\bar{x})\frac{b^n}{n!} + f^{(n+1)}(\xi)\frac{b^{n+1}}{(n + 1)!}, \quad \xi \in (\bar{x}, \bar{x} + b)
\]
Example of Taylor’s Theorem: $f(x) = 2\sqrt{x}$

Approximation of

$$f(x) = 2\sqrt{x}$$

around the fixed point

$$x^* = 4$$
Taylor approximations

Taylor’s approximation, $\mathbb{R}^n$ case

Let $f : \mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function on a neighborhood of $x$. Then

$$f(x + h) \approx f(x) + [\nabla f(x)]' h + \frac{1}{2} h' H(x) h$$

is a second-order Taylor approximation of $f$ at the point $x$. 

Taylor approximation around a fixed point

Suppose that the sequence $\mathbf{x}_{t+1} = f(\mathbf{x}_t)$ of vectors in $\mathbb{R}^n$ has a fixed point $\mathbf{x}^*$. Then the second-order Taylor approximation around $\mathbf{x}^*$ is

$$
\mathbf{x}_{t+1} - \mathbf{x}^* \approx [\nabla f(\mathbf{x}^*)]'(\mathbf{x}_t - \mathbf{x}^*) + \frac{1}{2}(\mathbf{x}_t - \mathbf{x}^*)'H(\mathbf{x})(\mathbf{x}_t - \mathbf{x}^*)
$$

The first-order approximation is simply

$$
\mathbf{x}_{t+1} - \mathbf{x}^* \approx [\nabla f(\mathbf{x}^*)]'(\mathbf{x}_t - \mathbf{x}^*)
$$

These expressions can be interpreted as describing “deviations from equilibrium”. Notice the similarity of the latter expression to a VAR(1) model written as deviation from the mean (without the stochastic term).
Implicit Function Theorem

Suppose $y = f(x)$ is a function of $x, f: \mathbb{R} \rightarrow \mathbb{R}$, but $y$ is implicitly defined by:

$$0 = g(x, y) = g(x, f(x))$$

How to compute the derivative of $y$ with respect to $x$ at a point $a$? Simply take the derivative of $g$ with respect to $x$:

$$0 = \frac{\partial g(x, y)}{\partial x} + \frac{\partial g(x, y)}{\partial y} \frac{dy}{dx}$$

As long as $g_y \neq 0$, the derivative $\frac{dy}{dx}$ at point $a$ is

$$\frac{dy}{dx} = - \left[ \frac{\partial g(a, f(a))}{\partial y} \right]^{-1} \frac{\partial g(a, f(a))}{\partial x}$$
Implicit Function Theorem
Second derivative

We can use $0 = g_x(x, y) + g_y(x, y)y'$ to obtain the second derivative $y''$:

\[
0 = \frac{\partial}{\partial x} g_x(x, y) + y' \frac{\partial}{\partial x} g_y(x, y) + g_y(x, y) \frac{\partial}{\partial x} y' \\
= g_{xx} + g_{xy} y' + y' (g_{yx} + g_{yy} y') + g_y y''
\]

Then, the second derivative is

\[
y'' = -\frac{1}{g_y} \left[ g_{xx} + 2g_{xy} y' + g_{yy} (y')^2 \right]
\]
Implicit Function Theorem, several variables

Now suppose that $y = f(x)$ is a function, $f : \mathbb{R}^n \to \mathbb{R}^m$, but $f$ is implicitly defined by $m$ possibly nonlinear functions $g : \mathbb{R}^{n+m} \to \mathbb{R}^m$:

$$0 = g(x, y) = \begin{bmatrix} g^1(x, y) \\ \vdots \\ g^m(x, y) \end{bmatrix}$$

The Jacobian of $f(x)$ at a point $\bar{x}$ is given by

$$J(\bar{x}) = - \left[ Dg_y(\bar{x}, f(\bar{x})) \right]^{-1} Dg_x(\bar{x}, f(\bar{x}))$$

where the $m \times m$ matrix $Dg_y$ is invertible.
Regular perturbation: the basic idea

- Suppose that a problem reduces to solving

\[ f(x, \epsilon) = 0 \]

for \( x \) where \( \epsilon \) is a parameter.

- We assume that for each value of \( \epsilon \) the equation has a solution for \( x \).

- Let \( x = x(\epsilon) \) denote a smooth function such that \( f(x(\epsilon), \epsilon) = 0 \).

- In general, we cannot solve the equation for arbitrary \( \epsilon \), but there may be special values of \( \epsilon \) for which the equation can be solved.
Example

Consider the function

\[ f(x(\epsilon), \epsilon) = x^3 - \epsilon x - 1 = 0 \]

For \( \epsilon = 0 \), it’s easy to find a solution:

\[ f(x(0), 0) = x^3 - 1 = 0 \Rightarrow x(0) = 1 \]

We will approximate solutions to

\[ x^3(\epsilon) - \epsilon x(\epsilon) - 1 = 0 \]

for arbitrary \( \epsilon \), based on knowing \( x(0) = 1 \).
Regular perturbation: the basic idea (cont.)

- If $f(x, \epsilon)$ and $x(\epsilon)$ are differentiable, and $x(0)$ is known, we can differentiate $f(x(\epsilon), \epsilon) = 0$ to get an implicit expression for $x'(\epsilon)$:

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

- Evaluated at known $x(0)$, it becomes

$$f_x(x(0), 0)x'(0) + f_\epsilon(x(0), 0) = 0 \Rightarrow x'(0) = -\frac{f_\epsilon(x(0), 0)}{f_x(x(0), 0)}$$

- Then, the linear approximation of $x(\epsilon)$ for $\epsilon$ near zero is

$$x(\epsilon) \approx x^{L}(\epsilon) = x(0) - \frac{f_\epsilon(x(0), 0)}{f_x(x(0), 0)} \epsilon$$
Example (cont.)

- For $f(x(\epsilon), \epsilon) = x^3 - \epsilon x - 1 = 0$ we get
  \[ f_x(x(\epsilon), \epsilon) = 3x^2 - \epsilon; \quad f_\epsilon(x(\epsilon), \epsilon) = -x \]

- Since $x(0) = 1$, the derivative is
  \[ x'(0) = -\frac{x(0)}{3x^2(0) - 0} = \frac{1}{3} \]

- The linear approximation is
  \[ x(\epsilon) \approx 1 + \frac{1}{3} \epsilon \]
The economist’s problem

- You have an intertemporal model with $n$ endogenous variables $x_t$, whose dynamics are described by $x_{t+1} = \Psi(x_t)$.
- You have found $n$ equilibrium conditions of the form $g(x_t, x_{t+1}) = 0$.
- The model is stationary, as such $\Psi$ has a fixed point $x^*$ (the steady state): $x^* = \Psi(x^*)$.
- **Problem:** How to analyze the dynamics of the model *without* an explicit solution for $\Psi(x)$?
The perturbation method solution

- **Solution:** Use Taylor’s theorem and the implicit function theorem to *approximate* the function $\Psi$ around $x^*$:

For example, the first order approximation is

$$x_{t+1} - x^* \approx \left[ \nabla f(x^*) \right]' (x_t - x^*)$$

(Taylor)

$$= - \left[ Dg_{x_t+1}(x^*, f(x^*)) \right]^{-1} Dg_{x_t}(x^*, f(x^*)) (x_t - x^*)$$

(IFT)

$$= - \left[ Dg_{x_t+1}(x^*, x^*) \right]^{-1} Dg_{x_t}(x^*, x^*) (x_t - x^*)$$

(FP)
Example: Solow model

- To illustrate the procedure of perturbation, we use Solow model
- It has only one dynamic equation, which will make easier to see the logic
Example: Solow model

In the Solow model, capital accumulates according to:

\[ 0 = g(k_t, k_{t+1}) = sA k_t^\alpha + (1 - \delta) k_t - (1 + n) k_{t+1} \]

The steady state is

\[ k^* = \left( \frac{sA}{n+\delta} \right)^{\frac{1}{1-\alpha}} \]

Notice that in this example, the solution is trivial: \( k_{t+1} = \Psi(k_t) = \frac{sA}{1-n} k_t^\alpha + \frac{1-\delta}{1-n} k_t \). We will use this model to illustrate the technique and to evaluate the quality of the approximation.
Finding the partial derivatives of Solow equation

\[ g(k_t, k_{t+1}) = sA k_t^\alpha + (1 - \delta) k_t - (1 + n) k_{t+1} \]

derivative . . .

\[ g_k = \alpha s A k_t^{\alpha-1} + (1 - \delta) \]
\[ g_{k'} = -(1 + n) \]
\[ g_{kk} = \alpha(\alpha - 1) sA k_t^{\alpha-2} \]
\[ g_{kk'} = 0 \]
\[ g_{k'k'} = 0 \]

\[ \rightarrow \text{ evaluated at } k^* \]
\[ \alpha(\delta + n) + 1 - \delta \]
\[ -1 - n \]
\[ \alpha(\alpha - 1)(\delta + n)(k^*)^{-1} \]
\[ 0 \]
\[ 0 \]
Finding the implicit derivatives on Solow equation

\[ k_{t+1} = f(k_t) \]

Using the implicit function theorem, the first derivative of \( k_{t+1} \) with respect to \( k_t \), evaluated at \( k^* \), is

\[ f'(k^*) = \frac{d k'_t}{d k_t} = -\frac{g_{k}}{g_{k'}} = \frac{\alpha(\delta + n) + 1 - \delta}{1 + n} \]

and the second derivative is

\[
\frac{d^2 k'_t}{d k'^2} = -\frac{1}{g_{k'}} \left\{ g_{kk} + 2g_{kk'} f'(k^*) + g_{kk'} [f'(k^*)]^2 \right\} \\
= \frac{\alpha(\alpha - 1)(\delta + n)}{(1 + n)k^*}
\]
First-order approximation to Solow model

In this case

\[ k_{t+1} - k^* \approx - \left[ Dg_{k_t+1}(k^*, k^*) \right]^{-1} Dg_{k_t}(k^*, k^*)(k_t - k^*) \]

\[ = \frac{\alpha(\delta + n) + 1 - \delta}{1 + n} (k_t - k^*) \]

\[ = \frac{1 + n - (1 - \alpha)(n + \delta)}{1 + n} (k_t - k^*) \]

Notice that \( \left| \frac{1 + n - (1 - \alpha)(n + \delta)}{1 + n} \right| < 1 \), so the system is stable.
Second-order approximation to Solow model

We just need to add the quadratic term to the previous approximation

\[ k_{t+1} - k^* \approx \frac{1+n-(1-\alpha)(n+\delta)}{1+n}(k_t - k^*) - \frac{\alpha(1-\alpha)(\delta+n)}{2(1+n)k^*} (k_t - k^*)^2 \]

Let \( \hat{k} = (k - k^*)/k^* \). After dividing both sides by \( k^* \), the approximation becomes

\[ \hat{k}_{t+1} \approx \frac{1+n-(1-\alpha)(n+\delta)}{1+n} \hat{k}_t - \frac{\alpha(1-\alpha)(\delta+n)}{2(1+n)} \hat{k}_t^2 \]

This will converge as long as

\[ -\frac{1}{\alpha} < \hat{k}_t < \frac{2(1+n)-(1-\alpha)(n+\delta)}{\alpha(1-\alpha)(n+\delta)} \]
Approximating the adjustment to steady state

Linear and quadratic approximations around $k^* = 0.51$

Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>0.10</td>
</tr>
<tr>
<td>$A$</td>
<td>1.00</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.40</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.10</td>
</tr>
<tr>
<td>$n$</td>
<td>0.05</td>
</tr>
</tbody>
</table>
Approximating the impulse response function

**Linear and quadratic** response to a 20% increase in $A$

Parameters:

<table>
<thead>
<tr>
<th></th>
<th>pre-</th>
<th>post-</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s$</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>$A$</td>
<td>1.00</td>
<td>1.20</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>$\delta$</td>
<td>0.10</td>
<td>0.10</td>
</tr>
<tr>
<td>$n$</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>$k^*$</td>
<td>0.51</td>
<td>0.69</td>
</tr>
</tbody>
</table>
Perturbation vs Chebyshev collocation

Quadratic polynomial approximations:

- **Chebyshev over** $k_i \in [0.8k^*, 1.2k^*]$, requires knowing $f(k_i)$ at three specific nodes $k_i$.

- **Taylor around** $k^* = 0.51$, requires knowing $k^* = f(k^*)$, $f'(k^*)$, and $f''(k^*)$.
Example: Ramsey model

The solution of the Ramsey problem is characterized by the equations

\[ \begin{align*}
0 &= g^1(K_t, C_t, K_{t+1}, C_{t+1}) = K_{t+1} - f(K_t) + C_t \\
0 &= g^2(K_t, C_t, K_{t+1}, C_{t+1}) = u'(C_t) - \beta u'(C_{t+1})f'(K_{t+1})
\end{align*} \]

The perturbation method will allow us to approximate the function \( \Psi \)

\[
\begin{bmatrix} K_{t+1} \\ C_{t+1} \end{bmatrix} = \Psi \left( \begin{bmatrix} K_{t+1} \\ C_{t+1} \end{bmatrix} \right)
\]

\[1\]This example follows the presentation by Heer and Maußner 2009, pp. 77–
Finding the steady state

The steady state is given by

\[
0 = g^1(K^*, C^*, K^*, C^*) = K^* - f(K^*) + C^*
\]
\[
0 = g^2(K^*, C^*, K^*, C^*) = \mu'(C^*) - \beta \mu'(C^*) f'(K^*)
\]

that is, the steady state $k^*$, $c^*$ satisfies

\[
f'(k^*) = \beta^{-1}
\]
\[
c^* = f(k^*) - k^*
\]
Finding the Jacobian

Let $\mathbf{x}_t = (k_t, c_t)'$. From the equations

\begin{align*}
0 &= g^1(K_t, C_t, K_{t+1}, C_{t+1}) = K_{t+1} - f(K_t) + C_t \\
0 &= g^2(K_t, C_t, K_{t+1}, C_{t+1}) = u'(C_t) - \beta u'(C_{t+1})f'(K_{t+1})
\end{align*}

we find the Jacobian of $g$ evaluated at the steady state

$J(\mathbf{x}_t, \mathbf{x}_{t+1}) = \begin{bmatrix} Dg_{\mathbf{x}_t} | Dg_{\mathbf{x}_{t+1}} \end{bmatrix}$

\[
= \begin{bmatrix}
-f'(k^*) & 1 & 1 & 0 \\
0 & u''(c^*) - \beta u'(c^*)f''(k^*) & -\beta u''(c^*)f'(k^*) & 0 \\
-\beta^{-1} & 1 & 1 & 0 \\
0 & u'' & -\beta uf'' & -u''
\end{bmatrix}
\]
The first-order approximation

The first order approximation is

\[
\begin{bmatrix}
k_{t+1} - k^* \\
c_{t+1} - c^*
\end{bmatrix}
= -\begin{bmatrix}
1 & 0 \\
-\beta u'f'' & -u''
\end{bmatrix}^{-1}
\begin{bmatrix}
-\beta^{-1} & 1 \\
0 & u''
\end{bmatrix}
\begin{bmatrix}
k_t - k^* \\
c_t - c^*
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\beta^{-1} & -1 \\
\frac{-u'f''}{u''} & 1 + \frac{\beta u'f''}{u''}
\end{bmatrix}
\begin{bmatrix}
k_t - k^* \\
c_t - c^*
\end{bmatrix}
\]
Outline

1. Perturbation Methods
   - Some math results
   - Perturbation in dynamic models
   - An example: The Solow model
   - Another example: The Ramsey model

2. Some Dynare basics
   - About Dynare
   - About model types

3. Examples of models in Dynare
   - A deterministic Solow model
   - A stochastic Ramsey model
What is Dynare?

- “Dynare is a powerful and highly customizable engine, with an intuitive front-end interface, to solve, simulate and estimate DSGE models.”
- It is a pre-processor and a collection of Matlab routines that has the great advantages of reading DSGE model equations written almost as in an academic paper.
- This not only facilitates the inputting of a model, but also enables you to easily share your code as it is straightforward to read by anyone.
Dynare’s workflow

In slightly less flowery words, it is a pre-processor and a collection of Matlab routines that has the great advantages of reading DSGE model equations written almost as in an academic paper. This not only facilitates the inputting of a model, but also enables you to easily share your code as it is straightforward to read by anyone.

Figure 1.2 gives you an overview of the way Dynare works. Basically, the model and its related attributes, like a shock structure for instance, is written equation by equation in an editor of your choice. The resulting file will be called the .mod file. That file is then called from Matlab. This initiates the Dynare pre-processor which translates the .mod file into a suitable input for the Matlab routines (more precisely, it creates intermediary Matlab or C files which are then used by Matlab code) used to either solve or estimate the model. Finally, results are presented in Matlab. Some more details on the internal files generated by Dynare is given in section 4.2 in chapter 4.

Each of these steps will become clear as you read through the User Guide, but for now it may be helpful to summarize what Dynare is able to do:

Source: Mancini Griffoli 2013
What is Dynare able to do?

- Compute the steady state of a model
- Compute the solution of deterministic models
- Compute the first and second order approximation to solutions of stochastic models
- Estimate parameters of DSGE models using either a maximum likelihood or a Bayesian approach
- Compute optimal policies in linear-quadratic models
A fundamental distinction

- Dynare can solve both deterministic and stochastic models.
- The distinction hinges on whether future shocks are known.
  - In deterministic models, the occurrence of all future shocks is known exactly at the time of computing the model’s solution.
  - In stochastic models, only the distribution of future shocks is known.
- If you only consider a first-order linear approximation of the stochastic model, or a linear model, the two cases become practically the same, due to certainty equivalence.
Deterministic vs stochastic models

**Deterministic models:**
- assume full information, perfect foresight, and no uncertainty about shocks.
- shocks can last one or more periods.
- the solution does not require linearization: exact path of endogenous variables.
- solution is useful even when economy is far away from steady state.

**Stochastic models**
- assume that shocks hit today (with a surprise), but thereafter their expected value is zero.
- expected future shocks, or permanent changes in the exogenous variables cannot be handled due to the use of Taylor approximations around a steady state.
- solution can be poor when economy is far from steady state.
A .mod file for a stochastic model

### Structure of the .mod file

- **Preamble**
  - Define variables & parameters

- **Model**
  - Spell out equations of model

- **Steady state or initial value**
  - Indicate steady state or initial value

- **Shocks**
  - Define shocks

- **Computation**
  - Ask to undertake specific operations

Source: Mancini Griffoli 2013
Installing Dynare in Matlab: a warning

- When installing Dynare, you should add Dynare to the Matlab path.
- This can be done by typing `addpath('c:\dynare\4.x.y\matlab')` , where “x.y” refers to the version of Dynare (e.g., the examples in this presentation were done with version 4.3.2)
- Do NOT add all Dynare subfolders in “c:\dynare\4.x.y” to the Matlab path, as doing so will add functions whose names conflict with those of Matlab functions.
Outline

1. Perturbation Methods
   - Some math results
   - Perturbation in dynamic models
   - An example: The Solow model
   - Another example: The Ramsey model

2. Some Dynare basics
   - About Dynare
   - About model types

3. Examples of models in Dynare
   - A deterministic Solow model
   - A stochastic Ramsey model
Simulating deterministic models in Dynare

- When the framework is deterministic, Dynare can be used for models with the assumption of perfect foresight.
- Typically, the system is supposed to be in a state of equilibrium before a period ‘1’ when the news of a contemporaneous or of a future shock is learned by the agents in the model.
- The purpose of the simulation is to describe the reaction in anticipation of, then in reaction to the shock, until the system returns to the old or to a new state of equilibrium.
- In most models, this return to equilibrium is only an asymptotic phenomenon, which one must approximate by an horizon of simulation far enough in the future.
- Another exercise for which Dynare is well suited is to study the transition path to a new equilibrium following a permanent shock.
**The output from Dynare**

**oo_.endo_simul**  The simulated endogenous variables are available in global matrix `oo_.endo_simul`. This variable stores the result of a deterministic simulation (computed by `simul`) or of a stochastic simulation (computed by `stoch_simul` with the `periods` option or by `extended_path`). The variables are arranged row by row, in order of declaration (as in `M_.endo_names`). Note that this variable also contains initial and terminal conditions, so it has more columns than the value of the `periods` option.

**oo_.exo_simul**  This variable stores the path of exogenous variables during a simulation. The variables are arranged in columns, in order of declaration (as in `M_.endo_names`). Periods are in rows. Note that this convention regarding columns and rows is the opposite of the convention for `oo_.endo_simul`!
The Solow Model

In the Solow model

- Production: $Y_t = A_t K_{t-1}^\alpha N_{t-1}^{1-\alpha}$
- Consumption: $C_t = (1 - s) Y_t$
- Capital accum: $K_t = s Y_{t-1} + (1 - \delta) k_{t-1}$
- Labor growth: $N_t = (1 + n) N_{t-1}$

The variables are de-trended: $x_t \equiv \frac{X_t}{N_t}$. Model is:

$$y_t = A_t k_{t-1}^\alpha$$
$$c_t = (1 - s) y_t$$
$$s y_t + (1 - \delta) k_{t-1}$$

The parameters are

$\alpha \in (0, 1)$
$\delta \in (0, 1)$
$s \in (0, 1)$
$n > 0$
$K_0$ given
The Solow Model: steady state

In steady state:

\[ y^* = Ak^*\alpha \]

\[ c^* = (1 - s)y^* \]

\[ (1 + n)k^* = sy^* + (1 - \delta)k^* \]

The steady state values are

\[ k^* = \left( \frac{sA}{n+\delta} \right)^{\frac{1}{1-\alpha}} \]

\[ y^* = A \left( \frac{sA}{n+\delta} \right)^{\frac{\alpha}{1-\alpha}} \]

\[ c^* = (1 - s)A \left( \frac{sA}{n+\delta} \right)^{\frac{\alpha}{1-\alpha}} \]
### From model to Dynare: Preamble

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td>// Declare variables and parameters</td>
<td>// Declare variables and parameters</td>
</tr>
<tr>
<td>$c_t$  $y_t$  $k_t$</td>
<td>var c y k ;</td>
</tr>
<tr>
<td>$A_t$</td>
<td>varexo A;</td>
</tr>
<tr>
<td>$\alpha$  $\delta$  $s$  $n$</td>
<td>parameters alpha delta s n;</td>
</tr>
<tr>
<td>$\alpha = 0.40$</td>
<td>alpha = 0.40;</td>
</tr>
<tr>
<td>$\delta = 0.10$</td>
<td>delta = 0.10;</td>
</tr>
<tr>
<td>$s = 0.10$</td>
<td>s = 0.10;</td>
</tr>
<tr>
<td>$n = 0.05$</td>
<td>n = 0.05;</td>
</tr>
</tbody>
</table>
## From model to Dynare: Model

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_t = A_t k_t^\alpha )</td>
<td>( \exp(y) = A \exp(\alpha k(-1)) )</td>
</tr>
<tr>
<td>( c_t = (1 - s) y_t )</td>
<td>( \exp(c) = (1 - s) \exp(y) )</td>
</tr>
<tr>
<td>( (1 + n) k_t = s y_t + (1 - \delta) k_{t-1} )</td>
<td>( (1+n) \exp(k) = s \exp(y(-1)) + (1-\delta) \exp(k(-1)) )</td>
</tr>
</tbody>
</table>

Note: writing \( \exp(x) \) instead of \( x \) allows to compute impulse response function as percent deviation from steady-state.
From model to Dynare: Initial values

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>// Initial values</td>
</tr>
<tr>
<td></td>
<td>initval;</td>
</tr>
<tr>
<td>$\hat{c}$</td>
<td>$c = \log(0.675);$</td>
</tr>
<tr>
<td>$\hat{y}$</td>
<td>$y = \log(0.75);$</td>
</tr>
<tr>
<td>$\hat{k}$</td>
<td>$k = \log(0.5);$</td>
</tr>
<tr>
<td>$\hat{A}$</td>
<td>$A = 1; \quad \text{end; \quad steady;}$</td>
</tr>
</tbody>
</table>

- Note: since model is written in log form, the (approximate) steady state values in this block are also written in log form.
- The command *steady* forces all initial values to the (exact) steady states.
From model to Dynare: Solving the model

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>// Shocks</td>
</tr>
<tr>
<td></td>
<td>shocks;</td>
</tr>
<tr>
<td>$A = 1.2$</td>
<td>var A;</td>
</tr>
<tr>
<td>$t = 4 : 6$</td>
<td>periods 1:1;</td>
</tr>
<tr>
<td></td>
<td>values 1.2;</td>
</tr>
<tr>
<td></td>
<td>end;</td>
</tr>
<tr>
<td></td>
<td>steady;</td>
</tr>
<tr>
<td></td>
<td>// Solving</td>
</tr>
<tr>
<td></td>
<td>solve! simul(periods=100);</td>
</tr>
</tbody>
</table>

- We are interested in analyzing the effect of increasing productivity from $A = 1$ to $A = 1.2$ from period 4 to period 6.
- We then simulate the model for 100 periods.
Examples of models in Dynare

A deterministic Solow model

Results: percent deviation from steady state

- Since marginal rate of savings is constant, the response of consumption just mirrors the response on income.
- There is a quick response from capital accumulation to increased productivity; once the shock is gone, capital adjusts slowly towards steady state.
- The initial jump in capital is due to additional savings; the slow adjustment is simply the effect of depreciation.
The Ramsey model

- The following model\(^2\) is a stripped down version of the celebrated model of Kydland and Prescott (1982), who were awarded the Nobel Price in economics 2004 for their contribution to the theory of business cycles and economic policy.
- The model provides an integrated framework for studying economic fluctuations in a growing economy.
- Since it depicts an economy without money it belongs to the class of real business cycle models.

\(^2\)This example is based on Heer and Maußner (2009, pp. 44-46)
The economy

- The economy is inhabited by a representative consumer-producer who derives utility from consumption $C_t$ and leisure $1 - N_t$ and uses labor $N_t$ and capital services $K_t$ to produce output $Y_t = Z_t(A_t N_t)^{1-\alpha} K_t^\alpha$.
- Labor augmenting technical progress at the deterministic rate $a > 1$ accounts for output growth: $A_{t+1} = a A_t$.
- Stationary shocks to total factor productivity $Z_t$ induce deviations from the balanced growth path of output: $\ln Z_{t+1} = \rho \ln Z_t + \epsilon_{t+1}$.
- Capital is accumulated according to $K_{t+1} = (1 - \delta) K_t + Y_t - C_t$. 
A Ramsey model

The representative agent solves:

\[
\max_{C_t, N_t, K_{t+1}} \mathbb{E}_0 \left[ \sum_{t=0}^{\infty} \beta^t \frac{C_t^{1-\eta}(1-N_t)^{\theta(1-\eta)}}{1-\eta} \right]
\]

subject to

\[
K_{t+1} + C_t = Y_t + (1 - \delta) K_t \\
Y_t = Z_t (A_t N_t)^{1-\alpha} K_t^\alpha \\
A_{t+1} = a A_t \\
\ln Z_{t+1} = \rho \ln Z_t + \epsilon_{t+1} \\
0 \leq C_t \\
0 \leq K_{t+1}
\]

\[
\begin{align*}
a &\geq 1 \\
\alpha &\in (0, 1) \\
\beta &\in (0, 1) \\
\eta &> \theta/(1 + \theta) \\
\theta &\geq 0 \\
\rho &\in (0, 1) \\
\sigma &> 0 \\
\epsilon &\sim N(0, \sigma^2)
\end{align*}
\]

\[K_0, Z_0 \text{ given}\]
First order conditions

From the Lagrangean

\[ \mathcal{L} := \mathbb{E}_0 \left\{ \sum_{t=0}^{\infty} \beta^t \frac{C_{t-\eta}(1 - N_t)^{\theta(1-\eta)}}{1 - \eta} + \Lambda_t [Y_t(N_t, K_t) - C_t - K_{t+1}] \right\} \]

we derive the first-order conditions

wrt \( C_t \):
\[ 0 = C_t^{-\eta}(1 - N_t)^{\theta(1-\eta)} - \Lambda_t \]

wrt \( N_t \):
\[ 0 = \theta C_t^{-\eta}(1 - N_t)^{\theta(1-\eta)-1} - \Lambda_t(1 - \alpha) \frac{Y_t}{N_t} \]

wrt \( K_{t+1} \):
\[ 0 = \Lambda_t - \beta \mathbb{E}_t \Lambda_{t+1} \left( 1 - \delta + \alpha \frac{Y_{t+1}}{K_{t+1}} \right) \]

wrt \( \Lambda_t \):
\[ 0 = K_{t+1} + (1 - \delta) K_t + C_t - Y_t \]
First order conditions, stationary version

Substitute out $\Lambda_t$ to get the system

\[
0 = \theta C_t - (1 - \alpha) Y_t \frac{1 - N_t}{N_t}
\]

\[
0 = C_t^{-\eta} (1 - N_t)^{\theta(1-\eta)} - \beta \mathbb{E}_t C_{t+1}^{-\eta} (1 - N_{t+1})^{\theta(1-\eta)} \left( 1 - \delta + \alpha \frac{Y_{t+1}}{K_{t+1}} \right)
\]

\[
0 = K_{t+1} - (1 - \delta) K_t + C_t - Y_t
\]

Define $x_t := X_t/A_t$ as a de-trended variable, then FOCs become:

\[
0 = \theta c_t - (1 - \alpha) y_t \frac{1 - N_t}{N_t}
\]

\[
0 = c_t^{-\eta} (1 - N_t)^{\theta(1-\eta)} - \beta a^{-\eta} \mathbb{E}_t c_{t+1}^{-\eta} (1 - N_{t+1})^{\theta(1-\eta)} \left( 1 - \delta + \alpha \frac{Y_{t+1}}{K_{t+1}} \right)
\]

\[
0 = a k_{t+1} - (1 - \delta) k_t + c_t - y_t
\]

\[
0 = y_t - Z_t N_t^{1-\alpha} k_t^\alpha
\]
The steady state

To find the steady-state, let $Z_t = 1$, drop the expectation operator:

$$0 = \theta c - (1 - \alpha)y \frac{1 - N}{N}$$

$$0 = 1 - \beta a^{-\eta} \left( 1 - \delta + \alpha \frac{y}{k} \right)$$

$$0 = (a + \delta - 1) k + c - y$$

$$0 = y - N^{1 - \alpha} k^\alpha$$

The (recursive) solution is:

$$N^* = \left\{ 1 + \frac{\theta}{1 - \alpha} \left[ \frac{a^n - \beta (1 - \delta) + \alpha \beta (1 - a - \delta)}{a^n - \beta (1 - \delta)} \right] \right\}^{-1}$$

$$k^* = \left[ \frac{\alpha \beta}{a^n - \beta (1 - \delta)} \right]^{1 - \alpha} N^*$$

$$y^* = N^*^{1 - \alpha} K^* \alpha$$

$$c^* = y^* - (a + \delta - 1) k^*$$
### From model to Dynare: Preamble

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c_t$</td>
<td>// Declare variables and parameters</td>
</tr>
<tr>
<td>$y_t$</td>
<td>var c y k N Z;</td>
</tr>
<tr>
<td>$k_{t+1}$</td>
<td>predetermined_variables k;</td>
</tr>
<tr>
<td>$N_t$</td>
<td>varexo e;</td>
</tr>
<tr>
<td>$Z_t$</td>
<td>parameters a alpha beta delta eta theta rho sigma;</td>
</tr>
<tr>
<td>$k_t$</td>
<td></td>
</tr>
<tr>
<td>$\epsilon_{t+1}$</td>
<td></td>
</tr>
</tbody>
</table>

| $a$ | 1.005 |
| $\alpha$ | 0.27 |
| $\beta$ | 0.994 |
| $\delta$ | 0.011 |
| $\eta$ | 2.0 |
| $\theta$ | 5.81 |
| $\rho$ | 0.90 |
| $\sigma$ | 0.0072 |
| $N^*$ | 0.13 |
| $N^*_{t+1}$ | |

Randall Romero (OSU)  Perturbation methods and Dynare  2014  59 / 63
• Dynare assumes that all variables are determined at $t$.
• But in this model $k_t$ is not decided at time $t$, but at $t - 1$.
• To alert Dynare about this, we need the line `predetermined_variables k;`.
From model to Dynare: Model

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
</table>
| \[\theta c_t = (1 - \alpha)y_t \frac{1-N_t}{N_t} \] | // Declare model’s equations  
| | model; |
| | theta*c = (1-alpha)*y*(1-N)/N; |
| \[y_t = a k_{t+1} - (1 - \delta) k_t + c_t \] | y = a*k(+1) - (1-delta)*k +c; |
| \[y_t = Z_t N_t^{1-\alpha} k_t^{\alpha} \] | y = Z*Nˆ(1-alpha)*kˆalpha; |
| \[c_t^{\eta}(1 - N_t)^{\theta(1-\eta)} = \beta a^{\eta} E_t c_{t+1}^{\eta}(1 - N_{t+1})^{\theta(1-\eta)} \left(1 - \delta + \alpha \frac{y_{t+1}}{k_{t+1}}\right) \] | cˆ(-eta)*(1-N)^(-theta*(1-eta)) = \bet*aˆ(-eta)*c(+1)ˆ(-eta) * (1-N(+1))ˆ(1-\theta*(1-eta)) *(1-delta + alpha*y(+1)/k(+1)); |
| \[\ln Z_t = \rho \ln Z_{t-1} + \epsilon_t \] | ln(Z) = rho*ln(Z(-1)) + e; |

Randall Romero  (OSU)
From model to Dynare: Initial values

<table>
<thead>
<tr>
<th>Model</th>
<th>Dynare</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>// Initial values</td>
</tr>
<tr>
<td></td>
<td>initval;</td>
</tr>
<tr>
<td>$c^*$</td>
<td>c = 0.25;</td>
</tr>
<tr>
<td>$y^*$</td>
<td>y = 0.30;</td>
</tr>
<tr>
<td>$k^*$</td>
<td>k = 3.02;</td>
</tr>
<tr>
<td>$N^*$</td>
<td>N = 0.13;</td>
</tr>
<tr>
<td></td>
<td>Z = 1.00;</td>
</tr>
<tr>
<td>$E \epsilon_t$</td>
<td>e = 0;</td>
</tr>
<tr>
<td></td>
<td>end;</td>
</tr>
<tr>
<td></td>
<td>steady;</td>
</tr>
<tr>
<td></td>
<td>check;</td>
</tr>
</tbody>
</table>

- The (approximate) numeric values of the steady state are computed separately using the analytical formulas.
- The command `steady` forces all initial values to the (exact) steady states.
- The command `check` ...
Examples of models in Dynare

A stochastic Ramsey model

References

Adjemian, Stéphane et al. (2013). *Dynare Reference Manual*. 4.3.2. Centre pour la Recherche Economique et ses Applications. URL:

