

Introdução ao Dynare

Modelos dinâmicos e estocástico de equilíbrio geral

Cássio Roberto de Andrade Alves

PPGEco/UFSC

Orientador: Prof. Dr. Guilherme Valle Moura

25 de setembro de 2018

Introdução

Tópicos abordados:

- 1 Explicação sobre o Dynare.
- 2 Resolver modelo de expectativas racionais.
- 3 Gerar Função impulso-resposta do Modelo Novo Keynesiano básico.
- 4 Simular séries.

Introdução

O que é o Dynare

- É um software para trabalhar com uma ampla gama de modelos econômicos;
- É 'executado' no Matlab ou octave;
- Permite trabalhar com modelos determinísticos e estocásticos;
- Vamos focar nos modelos estocásticos (DSGE).

Introdução

O que o Dynare pode fazer?

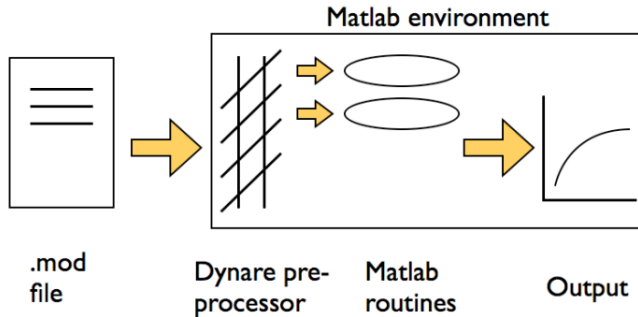
- O Dynare permite trabalhar com modelos calibrados.
 - Encontrar as funções políticas;
 - Simular as séries de tempo;
 - Análise de impulso-resposta.

Introdução

O que o Dynare pode fazer?

- O Dynare permite trabalhar com modelos calibrados.
 - Encontrar as funções políticas;
 - Simular as séries de tempo;
 - Análise de impulso-resposta.
- E permite também trabalhar com estimação de modelos de expectativas racionais (ER).
 - Estimação Clássica;
 - **Estimação Bayesiana.**

Como o Dynare funciona?



Modelo Novo-Keynesiano

- Como exemplo de modelo de ER, considere o modelo Novo-Keynesiano básico.
- O Dynare *aceita* tanto modelo não-linear quanto modelo log-linear(izado).

Exemplo utilizado

Considere o modelo Novo-Keynesiano log-linearizado.

Modelo

- O modelo pode ser resumido com as seguintes equações:

$$\pi_t = \beta \mathbb{E}_t[\pi_{t+1}] + \kappa \tilde{y}_t \quad (1)$$

$$\tilde{y}_t = \mathbb{E}_t[\tilde{y}_{t+1}] - \frac{1}{\sigma} (R_t - \mathbb{E}_t[\pi_{t+1}] - r_t^n) \quad (2)$$

$$i_t = \phi_\pi \pi_t + \phi_y \tilde{y}_t + \nu_t \quad (3)$$

$$r_t^n = \sigma \psi_n^{ya} (\mathbb{E}_t[a_{t+1}] - a_t) \quad (4)$$

$$r_t = i_t - \mathbb{E}_t[\pi_{t+1}] \quad (5)$$

$$y_t^n = \psi_{ya}^n a_t \quad (6)$$

$$\tilde{y}_t = y_t - y_t^n \quad (7)$$

$$m_t = y_t - \eta i_t \quad (8)$$

$$y_t = a_t + (1 - \alpha) n_t \quad (9)$$

Modelo (continuação...)

- As figuras reportadas por Galí (2008) mostram algumas variáveis anualizadas. Além dos processos AR, acrescente as seguintes eq.:

$$\nu_t = \rho_\nu \nu_{t-1} + \epsilon_t^\nu \quad (10)$$

$$a_t = \rho_a a_{t-1} + \epsilon_t^a \quad (11)$$

$$i_t^{(anual)} = 4i_t \quad (12)$$

$$r_t^{(anual)} = 4r_t \quad (13)$$

$$r_t^{n(anual)} = 4r_t^n \quad (14)$$

$$\pi_t^{(anual)} = 4\pi_t \quad (15)$$

$$\Delta m_t^{(anual)} = 4[(y_t - y_{t-1}) - \eta(i_t - i_{t-1}) + \pi_t] \quad (16)$$

Modelo (continuação...)

- No Dynare escrevemos as equações da forma como está em (1)-(16)
- Mas é importante saber que esse modelo pode ser escrito na forma matricial, coletando todas as variáveis em um vetor X_t :

$$\Gamma_0 X_t = \Gamma_1 X_t + \Psi \epsilon_t + \Pi \eta_t \quad (17)$$

- E então um método pode ser utilizado para resolver o modelo. Os métodos mais utilizados:
 - [Blanchard e Kahn \(1980\)](#)
 - [Sims \(2002\)](#)
 - [Klein \(2000\)](#)

Para uma exposição desses métodos ver [DeJong e Dave \(2011\)](#)

Modelo

- A solução do modelo na forma canônica de (17) pode ser escrita por:

$$X_t = \Phi X_{t-1} + R\epsilon_t \quad (18)$$

- Assim, tem-se as funções políticas e se pode fazer a análise de impulso-resposta
Sobre o método utilizado pelo Dynare ver [Villemot et al. \(2011\)](#)

Primeiros passos no código

- Podemos dividir nosso código em quatro seções.

Estrutura

```
// Preâmbulo
```

```
// Modelo
```

```
// Choques
```

```
// Simulação/Estimação
```

Boas práticas no Dynare

- ❶ No Dynare, sempre finalizamos uma linha de código com ponto-vírgula ';'.
- ❷ Devemos tomar cuidado com o nome de algumas variáveis ou parâmetros:
 - Por exemplo, 'alpha', 'beta', etc. são nomes reservados do Dynare/Matlab.
 - Por conta disso, devemos evitá-los.
 - Sugestão:
alpha → alppha
beta → betta

Boas práticas no Dynare

- ❶ Lembrar de fechar os blocos com **end;**.
- ❷ Convenção temporal.
 - No Dynare o período de uma variável se refere ao momento em que a variável é decidida.
 - Para denotar o tempo usamos:

$$y_t \longrightarrow y$$

$$\mathbb{E}_t[y_{t+1}] \longrightarrow y(+1)$$

$$y_{t-1} \longrightarrow y(-1)$$

Modelos com Variáveis predeterminadas

- É comum alguns artigos apresentarem a lei de movimento de algumas variáveis predeterminadas com a notação em $t + 1$.
- Por exemplo, modelos que incluem uma lei de movimento para o capital k_t .
- No entanto, no Dynare essas variáveis entram como período t .

1. Preâmbulo

// Variáveis do Modelo

var ppi y_gap i r_nat r_real y_nat nu ... m_real;

// Variáveis exógenas

varexo eps_nu eps_a;

// Parâmetros

parameters betta siggma phi epsilon phi_pi phi_y ... eta;

// Valores dos parâmetros

betta = 0.99;

siggma = 1;

...

eta = 6;

2. Modelo

```
model(linear);  
    // Parâmetros compostos  
    #Omega = (1-alpha)/(1-alpha+alpha*epsilon);  
    ...  
    #kappa = lambda*(sigma+(phi+alpha)/(1-alpha));  
  
    // Equações  
    // 1. Curva de Phillips Novo–Keynesiano  
    ppi = betta*ppi(+1) + kappa*y_gap;  
    ...  
    // 16. Demanda real por moeda  
    m_real = y-eta*i;  
end;
```


3. Choques

```
shocks;  
    var eps_nu = 0.25^2;  
    var eps_a = 1^2;  
end;
```

4. Simulação

```
// Estado estacionário: todos iguais a zero nesse caso
```

```
resid(1);  
steady;  
check;
```

```
// Simulação e FIR
```

```
stoch_simul(order = 1, irf=12, periods=200)  
y_gap pi_an y_n i_an r_real_an m_cresc_an a;
```

```
// Existem outras opções de parâmetros para stoch_simul  
// Ver na documentação
```

Executando o código no Matlab

- Quando terminamos o código no arquivo .mod, rodamos ele no Matlab
- Antes, porém, precisamos adicionar o caminho do Dynare.
- Isso pode ser feito por meio do botão “set path” -> “add folder”...
- Ou pela linha de comando (Windows):

```
>> addpath c:\dynare\4.x.y\matlab
```
- Ajuste x e y para versão do Dynare que você instalou
- Caso tenha instalado em uma pasta diferente da padrão, ajuste o caminho.

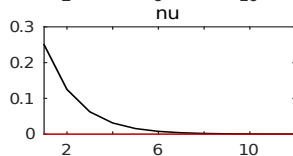
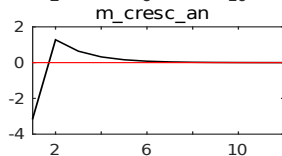
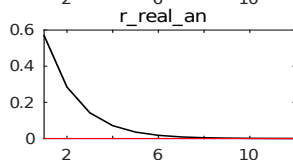
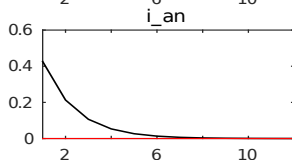
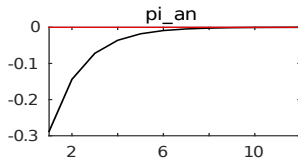
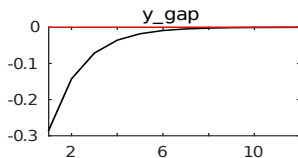
Executando o código no Matlab

- Ou pela linha de comando (Linux):

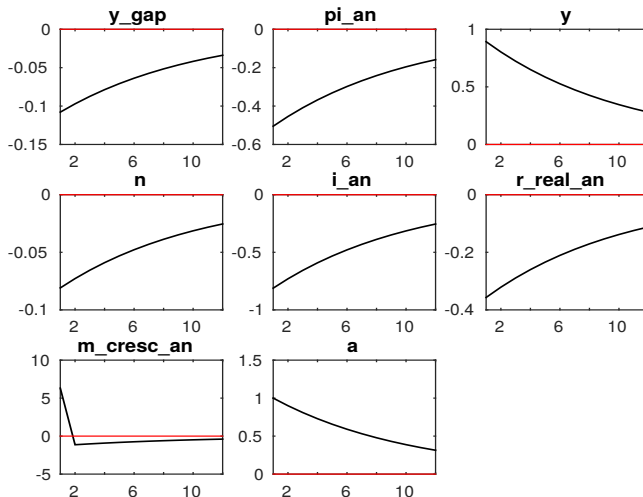
```
>> addpath /usr/lib/dynare/matlab
```
- Além disso, o diretório corrente do Matlab deve estar na pasta que contem o .mod
- Com os requisitos acima satisfeitos, basta chamar o arquivo .mod por:

```
>> dynare NOME_DO_ARQUIVO.mod
```

Função impulso-resposta: choque de política monetária



Função impulso-resposta: choque em tecnológico



Interpretação dos resultados

- Além das FIR geradas pelo comando `stoch_sim`, o Dynare imprime outros resultados na tela.
 - (Ver no Matlab os principais resultados)
- 1 Autovalores e condição de [Blanchard e Kahn \(1980\)](#).
 - 2 Resumo do modelo.
 - 3 Matriz de covariância dos choques exógenos.
 - 4 Funções políticas.

Onde os resultados foram armazenados?

- Em geral, as rotinas de Matlab criada pelo Dynare se encarrega de criar uma pasta no diretório e salvar alguns resultados
- As figuras geradas, por exemplo, são automaticamente salvas.
- A pasta `oo_` armazena os resultados
- E podemos acessá-los via linha de comando
- Por exemplo:

```
>> oo_.irfs
```

para acessar as funções impulso resposta.

- Com isso, uma das possibilidades é usar as ferramentas do Matlab para trabalhar com nossos resultados.

Resultados

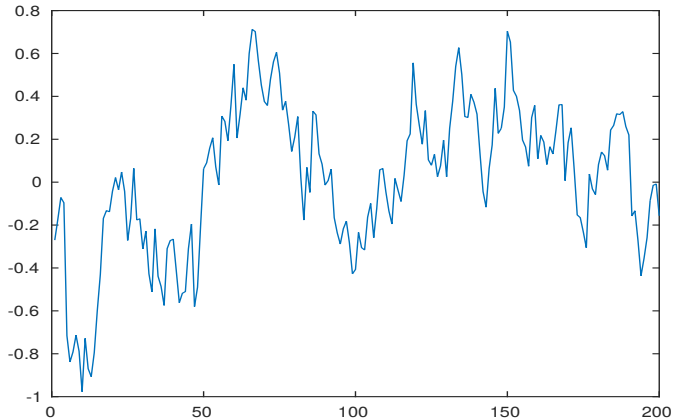
- Podemos acessar, por exemplo, a série gerada para o hiato do produto:

```
>> oo_.endo_simul(1,:)
```

- Bem como plotar um gráfico como é usual no Matlab

```
>> plot(oo_.endo_simul(1,:))
```

- Essa é uma forma de customizar nossos gráficos e não usar o padrão o Dynare.



Estimação de parâmetros

- Até aqui tratamos os parâmetros como conhecidos, i.e, calibramos o modelo.
- Podemos também estimar esses parâmetros.
- A abordagem bayesiana possui algumas vantagens por permitir incorporar informações de estudos anteriores.
- Além disso, o uso de distribuições a priori estabiliza a estimação dos parâmetros, o que é importante especialmente em contexto de séries de tempo curtas.

Estimação de parâmetros

- O arquivo .mod para estimação de parâmetros é parecida com o modelo calibrado

Dica

Quando vamos estimar um modelo é interessante fazer uma versão calibrada com parâmetros “razoáveis”. Isso pode nos ajudar a identificar algum possível erro no .mod.

- Até o bloco que computa o estado estacionário, o código é exatamente o mesmo do modelo calibrado.
- O próximo passo é declarar as variáveis observáveis.

Estimação de parâmetros

- Observações:
 - 1 As variáveis observáveis devem manter uma simetria com o modelo teórico.
 - 2 Caso a variável observável não seja exatamente simétrica com o modelo teórico, por exemplo, $\tilde{y}_t^{obs} = \tilde{y}_t$, será necessário incluir as equações de medida.
A esse respeito ver [Pfeifer \(2014\)](#)
 - 3 É necessário que o número de observáveis seja menor ou igual ao número de choques.

Declarando variáveis observáveis

```
// Variáveis observáveis  
varobs ppi i;
```

Parâmetros estimados e distribuição a priori

```
estimated_params;  
/* PARAM NAME, INITVAL, LB, UB,  
   PRIOR_SHAPE, PRIOR_P1, PRIOR_P2 */  
phi_pi, , , , normal_pdf, 1.5, 0.05;  
phi_y, , , , gamma_pdf, 0.25, 0.1;  
betta, , , , beta_pdf, 0.5, 0.25;  
rho_nu, , , , beta_pdf, 0.5, 0.25;  
rho_a, , , , beta_pdf, 0.5, 0.25;  
  
stderr eps_nu, inv_gamma_pdf, 0.5, inf; // dp do choque nu  
stderr eps_a, inv_gamma_pdf, 0.5, inf; // dp do choque a  
end;
```

Estimação de parâmetros

- Podemos passar os valores iniciais usando o seguinte comando:

Valores iniciais para max. a posteriori

```
/* Usa os valores calibrados para iniciar o algoritmo  
que maximiza a posteriori*/  
estimated_params_init(use_calibration);  
end;
```

- Após declararmos as variáveis observáveis, quais parâmetros serão estimados, as distribuições a priori e os valores iniciais, podemos passar para a estimação.

Estimação

Estimação

```
estimation(datafile=dados, mh_replic=100000,  
           mode_compute=4, mh_nblocks=2,  
           mh_drop=0.5, mh_jscale=0.2, mode_check);
```

- `datafile=NOME`: nome do arquivo contendo a base de dados a ser utilizada. Colunas do arquivo precisam ser nomeadas de acordo com os nomes em `varobs`. Arquivos `.mat`, `.xls` e `.m` são aceitos.

Estimação

- `nobs = INTEIRO`: número de obs. a ser utilizada na estimação (default=todas).
- `mh_replic = INTEIRO`: de replicações do algoritmo Metropolis-Hastings (default=20000).
- `mh_nblocks = INTEIRO`: de cadeias paralelas (default=2).
- `mh_drop = DOUBLE`: fração inicial da amostra que deve ser descartada antes do cálculo das estatísticas a posteriori (default=0.5).

Estimação

- `mh_jscale = DOUBLE`: escala usada na distribuição de saltos do MH. Deve ser ajustado de forma a gerar uma taxa de aceitação de 25% (default=0.2).
- `mode_compute = INTEIRO`: especifica qual otimizador usar para encontrar a moda da dist. a posteriori (default=4).
- `mode_check`: mostra o shape da posteriori para cada parâmetro.

