Dynare 中的 OBCs 及其应用

许文立，wlxu@cityu.edu.mo

澳门城市大学金融学院

2025 年 5 月 28 日

1. 概述
2. NK 模型
3. 确定性模拟：完美预期
4. 随机模拟：OccBin
5. OccBin 估计
6. 随机模拟：EP
7. 参考文献

1 概述
2 NK 模型
3 确定性模拟：完美预期
4 随机模拟：OccBin
5 OccBin 估计
6 随机模拟：EP
7 参考文献

OBCs

- 很多人做 DSGE，“习惯性”地对数线性化，得到线性模型：
  - a 变量之间的线性关系/经济含义一目了然
  - b 易于应用，尤其是估计
- 然而，许多有趣的经济问题和政策问题都具有非线性特征，例如，OBCs(Occasionally Binding Constraints)
  - a 利率的 ELB（Christiano et al. 2011; Walsh,2017）
  - b 信贷约束（Guerrieri and Iacoviello, 2017）
  - c 向下的工资粘性（Schmitt-Grohé and Uribe, 2016）
  - d 投资不可逆性（Guerrieri, 2015）
- 注意，假设 OBC 总是成立也不一定有价值，例如 Iacoviello (2005)、Jermann and Quadrini (2012)

OBCs

- OBCs 对于许多宏观经济现象和政策都非常关键
- OBCs 的存在，使得经济对正负冲击的响应有了不对称性
- OBCs 是许多不利放大机制的核心，例如，金融危机或者能源投入约束
- ZLB 时，财政政策效果会发生变化
- 向下名义工资粘性对于 boom-bust 周期很重要（Schmitt-Grohe,Uribe, JPE 2016）
- 线性模型无法刻画的其他特征

Dynare 中的 OBCs

- Dynare 6+ 提供了三种方式来解 OBCs 的模型：
  1、 非线性完美预期模型：max/min 算子、混合互补问题（MCP, mixed complementarity problems）
  2、 带无预期冲击的线性随机模型：OccBin
  3、 随机非线性模型：扩展路径法（Extended Path）

差异

- 确定性/完美预期：所有未来冲击都是精确已知的
- 随机模拟：只有冲击分布是已知的
- 确定性：
  - ＋ 任意精度
  - ＋ 很容易刻画非线性特征（例如 OBCs）
  - \- 完美预期假设
  - \- 如何模拟时间序列？
- 随机性：
  - ＋ 刻画不确定性
  - ＋ 需要估计状态空间形式
  - \- 会遇到维度灾难
  - ＋ 用额外的工具（OccBin）来配合扰动方法来刻画 OBCs
  - \- 通常用一阶近似：确定性等价

1 概述

2 NK 模型

三方程 NK

3 确定性模拟：完美预期

4 随机模拟：OccBin

5 OccBin 估计

6 随机模拟：EP

7 参考文献

1 概述

2 NK 模型
三方程 NK

3 确定性模拟：完美预期

4 随机模拟：OccBin

5 OccBin 估计

6 随机模拟：EP

7 参考文献

三方程 NK

- 一个三方程线性 NK 模型（Gali,2015; Walsh,2017）
  1. NKIS
  $$y_t = E_t[y_{t+1}] - \frac{1}{\sigma}(i_t - E_t[\pi_{t+1}]) + \epsilon_t^c \tag{1}$$
  2. NKPC
  $$\pi_t = \beta E_t[\pi_{t+1}] + \kappa y_t + \epsilon_t^a \tag{2}$$
  3. Taylor rule
  $$i_t = \rho_i i_{t-1} + (1-\rho_i)(\phi_\pi \pi_t + \phi_y y_t) + \epsilon_t^i \tag{3}$$
- 冲击：
  $$\epsilon_t^j = \rho^j \epsilon_{t-1}^j + u_t^j \tag{4}$$

OBC:ELB

- 名义利率的有效下界约束了央行降低利率的能力（不能降息到 i^{lb} 以下）
- 修正上述模型

$$i_t = max\{i_t^{no}, i^{lb}\} \tag{5}$$

其中，i_t^{no} 表示没有 ELB 约束的名义利率：

$$i_t^{no} = \rho_i i_{t-1}^{no} + (1-\rho_i)(\phi_\pi \pi_t + \phi_y y_t) + \epsilon_t^i \tag{6}$$

- 下面，我们用需求冲击 u_t^c 来诱发 ELB 约束

Dynare 代码：模型部分

```
model(linear);
[name = 'NKIS']
y =y(1)-1/sigma*(ino-pi(1))+epsc;

[name = 'NKPC']
pi =bet*pi(1)+kappa*y+epsa;

[name = '货币政策']
ino=rhoi*ino(-1)+(1-rhoi)*(phipi*pi+phiy*y)+epsi;
```

1 概述

2 NK 模型

3 确定性模拟：完美预期
   max/min 算子
   MCP

4 随机模拟：OccBin

5 OccBin 估计

6 随机模拟：EP

7 参考文献

1 概述

2 NK 模型

3 确定性模拟：完美预期
- max/min 算子
- MCP

4 随机模拟：OccBin

5 OccBin 估计

6 随机模拟：EP

7 参考文献

完美预期算法的 ELB

- 在 Dynare 的 model-block 通过 max 算子来实施 ELB

```
[name = '货币政策']
ino=rhoi*ino(-1)+(1-rhoi)*(phipi*pi+phiy*y)+epsi;

[name = '可观测利率']
ino = max(ino,ilb);
```

- 注意：max 算子不可微，这对依赖于稀疏 Jacobian 的 Newton 解带来挑战

模拟 1% 需求冲击

```
shockssequence = -0.01;

shocks;
    var uc;
    periods 1;
    values(shockssequence);
ends

perfect_foresight_setup(periods=50);
perfect_foresight_solver;
```

- 重要：设置足够长的时期，回到稳态
- 完整代码见 NK_det.mod

1 概述

2 NK 模型

3 确定性模拟：完美预期
max/min 算子
MCP

4 随机模拟：OccBin

5 OccBin 估计

6 随机模拟：EP

7 参考文献

完美预期算法：MCP

- 备择方法：用专业的 MCP 算法：
  - 用 lmmcp 或者 perfect_foresight_solver(stack_solve_algo=7, solve_algo=10) 命令来诱发 Levenberg-Marquardt 混合互补问题算法（Kanzow and Petra，2004）
  - PATH 解（Ferris and Munson，1999）：perfect_foresight_solver(stack_solve _algo=7, solve_algo=11)
- 上述解方法要求特定类型的设定：

$$LB = X \Rightarrow F(X) > 0 \tag{7}$$

$$LB \leq X \leq UB \Rightarrow F(X) = 0 \tag{8}$$

$$X = UB \Rightarrow F(X) < 0 \tag{9}$$

- OBC 的方程余值的符号特别重要
- Dynare 传统：LHS-RHS

Dynare 中的 MCP

- mcp-tag 指示着取代的方程（HRS 表达式并没有被解析）

```
[name = ' 无约束利率规则',mcp = 'ino>-0.0055']
ino = inomnot;

[name = ' 无约束利率规则']
inomnot = rhoi*inomnot(-1)+(1-rhoi)*(phipi*pi+phiy*y)+epsi;
```

- 如果利率下界 binding：

$$residual = i_t - i_t^{no} > 0 \tag{10}$$

模拟 1% 需求冲击

```
shockssequence = -0.01;

shocks;
    var uc;
    periods 1;
    values(shockssequence);
ends

perfect_foresight_setup(periods=50);
perfect_foresight_solver(lmmcp);
```

- 关键词 lmmcp 诱发专业的算法
- 完整代码见 NK_mcp.mod

1. 概述
2. NK 模型
3. 确定性模拟：完美预期
4. 随机模拟：OccBin
5. OccBin 估计
6. 随机模拟：EP
7. 参考文献

概述

- Dynare 用扰动方法来解随机模型：泰勒展开
- OBCs 引入了不可微特征
- OccBin（Guerrieri and Iacoviello，2015）：用 piecewise linear 方法来应对 OBCs
- OBCs 在同一模型里不同的模式，例如，
  - 不受 ELB 约束的经济
  - 受 ELB 约束的经济
- 由于反馈环路，模型解具有高度的非线性特征：
  - 一种模式的动态依赖于该模式的预期持续期
  - 而预期持续性依赖于状态

优势和劣势

- 与线性模型解相似，这个方法不能解释不确定性（预防性行为）
- 解并没有解释约束在未来的 binding（因为冲击还未实现）
- 但是，它非常快，可以用于估计
- 可以应用于大规模模型

两种模式的表达式

- 基准模式（ELB 无约束）可以表达成下列线性系统：

$$A_1 E_t\left[X_{t+1}\right] + A_0 X_t + A_{-1} X_{t-1} + E u_t = 0 \tag{11}$$

其中，X_t 表示偏离稳态的模型变量
- 备择模式（ELB 约束）可以用一个常数 D^* 来刻画

$$A_1^* E_t\left[X_{t+1}\right] + A_0^* X_t + A_{-1}^* X_{t-1} + D^* + E^* u_t \tag{12}$$

- 两种模式（11）和（12）都是在同一个点处线性化（D^* 解释了潜在不同的稳态）
- 重要：基准模式的经济没有任何冲击

模式和解

- 基准模式（11）：
  - 解可以写成：

$$X_t = \mathcal{P} X_{t-1} + C u_t \tag{13}$$

  - 转移矩阵 $\mathcal{P}$ 和 C 是常数
  - BK 条件满足
  - 提供线性化点
- 备择模式（12）：
  - 解可以写成：

$$X_t = \begin{cases} \mathcal{P}_t X_{t-1} + R_t + C_t u_t & \text{for } t = 1 \\ \mathcal{P}_t X_{t-1} + R_t & \text{for } t \geq 2 \end{cases} \tag{14}$$

  - 时变矩阵依赖于模式的预期长度
- 假设确定性等价：系统不依赖于 u_{t+j}，人们预期未来冲击等于他们的预期值

解法：后看 + 猜测验证

- 我们知道终值条件：如果未来所有时期 $u_t = 0$，经济系统就会回到基准模式（11）
- 需要检查我们是否从 T 期回到了（11）
- 给定状态 X_{t-1}，对于 $t < T$，模式（12）(潜在冲击 u_t)
- 猜测验证来找出模式的持续期：
    1. 在 t_0 猜测 T，以使得对于 $t \geq T$，我们会回到模式（11）
    2. 检查我们猜测是否与模型动态一直
    3. 如果不一致，就继续猜测

后看

- 对于任意的 $t \geq T$，预期 $u_t = 0$，模式（11）的解就意味着：

$$X_t = PX_{t-1} \tag{15}$$

- 那么，T-1 的预期：

$$E_{T-1}[X_T] = PX_{T-1} \tag{16}$$

- 这允许消除线性系统（12）的前看部分：

$$A_1^* E_{T-1}\left[X_T\right] + A_0^* X_{T-1} + A_{-1}^* X_{T-2} + D^* = A_1^* \mathcal{P} X_{T-1} + A_0^* X_{T-1} + A_{-1}^* \tag{17}$$

- 给定状态 X_{T-2}，有一个未知的矩阵方程 X_{T-1}

后看解法：T-1 的解

- 给定 X_{T-2}，解 X_{T-1} 可以得到决策规则：

$$X_{T-1} = -\left(A_1^* \mathcal{P} + A_0^*\right)^{-1} \left(A_{-1}^* X_{T-2} + D^*\right) \tag{18}$$

- 线性：

$$X_{T-1} = \underbrace{P_{T-1}}_{-\left(A_1^* \mathcal{P}+A_0^*\right)^{-1} A_{-1}^*} X_{T-2} + \underbrace{R_{T-1}}_{-\left(A_1^* \mathcal{P}+A_0^*\right)^{-1} D^*} \tag{19}$$

- 线性可以容易评价 T-2 期条件期望 $T_{T-2}[X_{T-1}]$

后看解法：直到第 1 期

- 继续上述过程到 T-2 期：

$$A_1^* P_{T-1} X_{T-2} + A_0^* X_{T-2} + A_{-1}^* X_{T-3} + D^* = 0 \tag{20}$$

- 给定 X_{T-3}，解 X_{T-2} 的决策规则：

$$X_{T-2} = P_{T-2} X_{T-3} + R_{T-2} \tag{21}$$

其中，

$$P_{T-2} = -\left(A_1^* P_{T-1} + A_0^*\right)^{-1} A_{-1}^* \tag{22}$$

$$R_{T-2} = -\left(A_1^* \mathcal{P}_{T-1} + A_0^*\right)^{-1} D^* \tag{23}$$

- P_t, R_t, C_t 是构成 (11) 和（12）模式结构矩阵的时变矩阵
- 非线性动态

后看解法：第 1 期

- 最后到第 1 期：

$$A_1^* P_2 X_1 + A_0^* X_1 + A_{-1}^* X_0 + D^* + E^* u_1 = 0 \tag{24}$$

- 给定 X_0 和当前冲击 u_1，解 X_1 的决策规则：

$$X_1 = P_1 X_0 + R_1 + C_1 u_1 \tag{25}$$

其中，

$$\mathcal{P}_1 = -\left(A_1^* \mathcal{P}_2 + A_0^*\right)^{-1} A_0^* \tag{26}$$

$$R_1 = -(A_1^* \mathcal{P}_2 + A_0^*)^{-1} D^* \tag{27}$$

$$C_1 = -\left(A_1^* \mathcal{P}_2 + A_0^*\right)^{-1} \mathcal{E}^* \tag{28}$$

- 到此，完成了（12）模式的计算

验证

- 我们现在获得了所有时期的时间依赖决策规则，但是有可能条件与潜在有误的猜测 T
- 给定冲击 ϵ_1，从 X_0 开始的经济行为与假设的决策规则一致吗？
- 计算 X_{t+j} 的时间路径，反推出基准模式 $\hat{T}$
- 如果 $\hat{T} = T$ 得到验证：解就找到了
- 如果不相等，就继续猜测模式序列
    - 利用计算的 X_t 路径来升级下一个猜测值
    - 其它的升级模式（例如，模式猜测步长，simul_curb_retrench）
- 多重解：多模式序列（Holden,2024）

- 方程的 name-tag 用于命名方程
- bind 和 relax-tag 将方程配置给备择模式和基准模式

```
[name='Notional rate Taylor rule']
inomnot=rhoi*inomnot(-1)+（1-rhoi）
*(phipi*pie+phiy*y)+epsi;

[name = 'Observed interest rate',relax='zlb']
inom = inomnot;

[name = 'Observed interest rate',bind='zlb']
inom = ilb;
```

- 声明模式转换条件

```
occbin_constraints;
name 'zlb'; bind inomnot<=ilb;
end;
```

等价于

```
occbin_constraints;
name 'zlb'; bind inomnot<=ilb; relax inomnot>ilb;
end;
```

当 bind 表达式不能在备择模式下评价时，就要声明 relax-tag

- 在每一个模拟期，Dynare 将会检查一个模式是否发生转变的条件
- 在模式之间的边界处就会出数值不精确
- 技巧：引入一个安全 buffer：

```
occbin_constraints;
name 'zlb'; bind inomnot<=ilb-1e-8; relax inomnot>ilb+1e-8;
end;
```

```
shockssequence = -0.01;

shocks(surprise);
    var u_c;
    periods 1;
    values(shockssequence);
end;
```

关键是冲击模块的 surprise，意味着使用完美预期命令，且未来的冲击不可预期

```
occbin_setup;
occbin_solver(simul_periods=20,
simul_check_ahead_periods=50);
occbin_graph y ino inomnot pi;
```

occbin_setup 是触发 OccBin 工具箱的命令

occbin_solver 触发模拟

- simul_periods：模拟期数
- simul_check_ahead_periods: 模式回到基准的 T 期的初始猜测值

代码见NK_occbin.mod

1 概述

2 NK 模型

3 确定性模拟：完美预期

4 随机模拟：OccBin

5 OccBin 估计
 概述
 三个应用

6 随机模拟：EP

7 参考文献

1 概述

2 NK 模型

3 确定性模拟：完美预期

4 随机模拟：OccBin

5 OccBin 估计
   概述
   三个应用

6 随机模拟：EP

7 参考文献

- 许多时候，人们都比较关心模型的经验拟合度
- Giovannini, Pfeiffer, Ratto(2021) 提出了一种一般性的，有效的，易于实施的方法来估计带来 OBCs 的 DSGE 模型：Piecewise Kalman Filter
- PKF 基于扰动方法和 OBCs 工具箱的模拟方法
- 当前 OBCs 的估计方法有些缺陷：
  - Inversion Filter(Guerrieri and Iacoviello, 2017, JME) 应用了 piecewise-linear 近似：从观测时间序列“倒推”冲击意味着冲击数量等于观测变量数量，排除了一些有趣的应用
  - Particle Filter(Cuba-Borda et al.,2019, JAE; Atkinson et al., 2019, JME) 状态变量较多，计算成本大，对测量误差选择较为敏感

- Farmer(2021) 在非高斯状态空间模型中提出了一种似然函数的快速非线性近似
- Aruoba et al.(2020) 讨论了高阶解和粒子滤波的 OBCs 的 piecewise-linear 性质
- Boehl and Strobel(2020) 提出 Ensemble Kalman Filter 和 piecewise-linear 解算法结合
- 这些方法比较适合低维状态空间，但是对于许多状态变量的模型则不适用
- 对于 Markov-switching 模型，Maih(2015) 提供了一种有效率的扰动方法
- 而 PKF 则可以处理缺失数据和非平稳序列，类似于标准的 diffuse kalman filter

piecewise linear model

- 利用 Dynare 中的 OccBin 工具箱（Guerrieri and Iacoviello’s (2015, JME)），意味着时变决策规则

$$
\begin{aligned}
\mathrm{xt} &= \psi^{OccBin}(x_{t-1}, \varepsilon_t; \theta) \\
&= \mathrm{T}(x_{t-1}, \varepsilon_t; \theta) x_{t-1} + \mathrm{C}(x_{t-1}, \varepsilon_t; \theta) + \mathrm{R}(x_{t-1}, \varepsilon_t; \theta)\varepsilon_t,
\end{aligned} \tag{29}
$$

其中，状态变量 x_t，冲击 ϵ_t，深度参数 θ

- 观测方程：$y_t = Hx_t$
- $j = 1, \ldots, n_t$ 约束集：
$r_{j,t} = max\{a_j x_{t+1} + b_j x_t + c_j x_{t-1} + d_j \epsilon_t, \bar{r}_j\}$，其中，限制
$d_j = 0$

估计

- 目标：用 likelihood 函数估计模型的参数 θ
  - 观测到的数据集 y^T 的似然函数是参数 θ 的函数
  - 找到 θ 使得似然函数最大化，但是需要追踪估计的状态变量和估计量的方差
- 标准线性模型的 Kalman Filter：

$$x_t = T(\theta)x_{t-1} + R(\theta)\epsilon_t \tag{30}$$

很容易得到 likelihood 函数，并估计参数 θ

- 然而，T,C,R 是状态变量和冲击的函数

$$x_t = T(x_{t-1}, \epsilon_t; \theta)x_{t-1} + C(x_{t-1}, \epsilon_t; \theta) + R(x_{t-1}, \epsilon_t; \theta)\epsilon_t \tag{31}$$

- 直觉上说，决策规则依赖于经济的状态，例如，ZLB 的持续状态

OccBin

- 对于许多状态变量的模型，OccBin 用一种快速的算法找到这些矩阵（Guerrieri and Iacoviello, JME, 2015）
- 一种“猜测、验证”法，细节见上文
- Giovannini, Pfeiffer, Ratto(2021) 将这种算法引入了 Kalman Filter 来估计 OBCs 模型
- OBCs 估计算法细节见 Giovannini, Pfeiffer, Ratto(2021)，或者 2022 Dynare Workshop for Advanced users 的 slides

1 概述

2 NK 模型

3 确定性模拟：完美预期

4 随机模拟：OccBin

5 OccBin 估计
   概述
   三个应用

6 随机模拟：EP

7 参考文献

1. 消费-储蓄模型中的借贷约束（Cuba-Borda et al.,2019,JAE）
   - 用不同近似误差的模型来作为 DGPs
   - 强调（速度）优势和劣势
2. 中等规模 NK 模型，刻画 ZLB（Atkinson et al., 2019,JME）
   - 在中等规模模型中，PKF 比其它非线性算法要好
3. 大型三国模型，刻画 ZLB 和向下工资粘性
   - PKF 在处理大规模模型时，速度较快，且有效率

借贷约束

- 首先，考察一个小规模模型——Cuba-Borda(2019)
- 这个模型刻画了内生借贷约束，这是现代非线性宏观模型的核心特征，例如，Mendoza (2006), Bocola (2016), and He and Krishnamurthy (2011)
- 消费者最大化效用

$$\max E_0 \sum_{t=0}^{\infty} \beta^t \frac{C^{1-\gamma}}{1-\gamma}$$

约束为

$$C_t + RB_{t-1} = Y_t + B_t$$

总利率 R，借款 B_t 的约束为

$$B_t \leq mY_t$$

- 收入是随机过程：

$$\log(Y_t) = \rho \log(Y_{t-1}) + \sigma\varepsilon_t$$

均衡

- 均衡系统
$$
\begin{aligned}
C_t + RB_{t-1} &= Y_t + B_t \\
ln(Y_t) &= \rho \ln(Y_{t-1}) + \sigma \epsilon_t \\
C_t^{-\gamma} &= \beta R E_t(C_{t+1}^{-\gamma}) + \lambda_t \\
\lambda_t(B_t - mY_t) &= 0
\end{aligned}
$$
  - 如果收入和资产较低：借贷约束很有可能 binds
  - 如果收入和资产较高：相较于未来消费，借贷也较高，及时仍低于最大值
- 对于高于平均收入水平，相较于最优储蓄的未来，当前消费足够高

ZLB 的 NK 模型

- Atkinson et al. (2019, JME) 的模型是用全局解算法解出来的，技术、风险溢价和货币冲击驱动的波动
- 关键的非线性特征：ZLB
- 模型设定见 Xu(2025) 讲稿

多国模型

- 三国模型：西班牙、欧元区和世界其他地区
- 两个关键的非线性特征：ZLB+DNWR
- 估计用了 40 个时间序列变量
- 这个模型用于研究 DNWR 在扩张经济周期中的作用（Burgert, Pfeiffer and Roeger, 2022）
- DNWR 会组织名义工资的调整，失业增加

DNWR 约束

- DNWR 是对名义工资的一个 OBC 约束

$$\frac{W_{jt}}{W_{jt-1}} \geq \gamma, \quad \gamma > 0$$

- 互补松弛（Kuhn-Tucker）条件

$$\lambda_t^W \left(\frac{W_{jt}}{W_{jt-1}} - \gamma \right) = 0$$

- λ_t^W 表示 DNWR 约束的拉格朗日乘数
  - 当约束不成立时，$\lambda_t^W = 0$
  - 当约束 bind 时，$\frac{W_{jt}}{W_{jt-1}} = \gamma$
- 代码见occbin_pk_est.mod

- 注意，当前的 PKF 并不能估计 OBCs 附近的不确定性
- PKF 不能处理 OBCs 诱发的随机奇异问题
- 观测变量必须包含关于约束处的模型行为信息

1. 概述
2. NK 模型
3. 确定性模拟：完美预期
4. 随机模拟：OccBin
5. OccBin 估计
6. 随机模拟：EP
7. 参考文献

- EP 算法已经有了，见 Adjemian and Juillard(2025):Stochastic Extended Path
- 计划在 Dynare 7.x 中呈现完整的界面

1. 概述
2. NK 模型
3. 确定性模拟：完美预期
4. 随机模拟：OccBin
5. OccBin 估计
6. 随机模拟：EP
7. 参考文献

参考文献

Johannes Pfeifer,2022,Solving Models with Occasionally Binding Constraints,Dynare Workshop for Advanced Users 2022

Thanks!