Solving and Simulating DSGE Models in Julia

Course 3 — Short Macroeconomics Course Using Julia

November 26, 2025

Rational Expectations and Filtrations

- We work on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ with filtration $\{\mathcal{F}_t\}$.
- RE means agents' conditional beliefs match the model's law of motion given info $\mathcal{F}_t$.
- Law of iterated expectations: $E(x_t \mid \mathcal{F}_{t-1}) = E(E(x_t \mid \mathcal{F}_t) \mid \mathcal{F}_{t-1})$.

Scalar RE: First-Order Stochastic Linear Equation

Baseline: $E_t x_{t+1} = a x_t + b z_t$, with stable exogenous z_t.

- If $|a| < 1$: backward-looking solution.
- If $|a| > 1$: forward-looking solution.

Both can be derived via lag operator or undetermined coefficients.

Forward Solution (Sketch)

Using lag operator properties,

$$(L^{-1} - a)x_t = bz_t \;\Rightarrow\; x_t = -b\sum_{j\geq 0} a^{-(j+1)} E_t z_{t+j}.$$

Current x_t is a discounted sum of expected future z's if $|a| > 1$.

From Scalar to Multivariate RE Systems

General LRE form:

$$A\,E_t x_{t+1} = B x_t + C z_t, \qquad z_{t+1} = \Phi z_t + \Gamma \varepsilon_{t+1},\ E_t \varepsilon_{t+1} = 0.$$

Solution methods differ on: (i) handling singular A, (ii) treatment of predetermined vs jump variables.

BK, Klein & Sims method

Blanchard–Kahn: requires nonsingular A; uniqueness if #unstable roots = #jumps (order) and rank condition holds.
Klein: handles singular A via QZ on (A, B) with stability and invertibility blocks; conditions for unique stable solution.
Sims: stacks model as

$$\Gamma_0 y_t = \Gamma_1 y_{t-1} + C + \Psi z_t + \Pi \eta_t \tag{1}$$

and uses QZ; no explicit split of jumps/predetermined; unique solution test via partitioned Schur form.

Blanchard-Kahn method

There are two key characteristics of this approach: the matrix A is assumed to be non-singular. Furthermore, according to the method they proposed a variable is classified as predetermined when it satisfies the following definition.

Definition We say that a stochastic process $\{x_t\}_{t\geq 0}$ is predetermined if two conditions are met: x_0 is predetermined and x_{t+1} is measurable with respect to $\mathcal{F}_t$, where $\mathcal{F}_t$ is defined as a filtration. The second condition can be formally written as: $E[x_{t+1}|\mathcal{F}_t] = x_{t+1}$.

Following the results for the deterministic case, we can split again the vector x_t into a sub-vector y_t of non-predetermined variables of dimension n_y and a subvector k_t of predetermined variables of dimension $n - n_y$. Thus we would write: $x_t = [k_t', y_t']$.

Assuming that there are a number of n_u unstable eigenvalues and a number of $n_s = n - n_u$ stable eigenvalues, following the results for the deterministic case, we may formally write a forward looking solution as follows:

$$u_t = -J_u^{(-1)}(I - L^{(-1)}J^{(-1)})^{(-1)}C_u^* z_t = \tag{2}$$

$$-\sum_{j=0}^{\infty} J_u^{(-j-1)} C_u^* E_t z_{t+j} \tag{3}$$

Sims method

The representation above can be simplified conveniently by assuming that the matrix $C = 0$, if we use change of variables, and that z_t is serially correlated such that z_t is appended to y_t and replaced by a process ϵ_t. With these assumptions, the system becomes:

$$\Gamma_0 y_t = \Gamma_1 y_{t-1} + \Psi \epsilon_t + \Pi \eta_t \tag{4}$$

As in the method proposed by Klein, Sims' approach allows for a singular matrix A which, in his notation, is given by Γ_0. However, what really differentiates this approach is that the method does not distinct between predetermined and jump variables.
The method proposed by Sims is also based on the generalized Schur decomposition. Here, we apply it matrices (Γ_0, Γ_1) which are assumed to be regular. According to the generalized Schur Decomposition, there are unitary complex matrices Q, Z as well as upper triangular matrices S, T such that the following relationships hold: $Q\Gamma_0 Z = S$ and $Q\Gamma_1 Z = T$.

Model (Log-linear form)

Canonical RBC with AR(1) TFP (log-linearized):

$$E_t\tilde{c}_{t+1} = \tilde{c}_t + (1-\beta(1-\delta))(E_t\tilde{y}_{t+1} + E_t\tilde{k}_{t+1}) \tag{5}$$

$$(1+\psi)\tilde{h}_t + \tilde{c}_t - \tilde{y}_t = 0 \tag{6}$$

$$\tilde{y}_t - (1-\alpha)\tilde{h}_t - \alpha\tilde{k}_t - \tilde{a}_t = 0 \tag{7}$$

$$\tilde{y}_t = \frac{\bar{c}}{\bar{y}}\tilde{c}_t + \frac{\bar{i}}{\bar{y}}\tilde{i}_t \tag{8}$$

$$\tilde{k}_{t+1} = \delta\tilde{i}_t + (1-\delta)\tilde{k}_t \tag{9}$$

$$\tilde{a}_t = \rho\tilde{a}_{t-1} - \tilde{\epsilon}_t \tag{10}$$

Euler, intratemporal FOC, production fct, resource constraint, capital law of motion, shock dynamics. We group variables into y_t (dynamic block) and x_t (static block), i.e. $y_t = \{\tilde{k}_{t+1}, \tilde{a}_t, E_t\tilde{c}_{t+1}\}$ and $x_t = \{\tilde{y}, \tilde{c}_t, \tilde{i}_t, \tilde{h}_t\}$

Sims Representation

Using these definitions, we can write the system as a set of two equations, with the first one grouping the static equations and the second one collecting the dynamics ones, as seen below:

$$\begin{aligned} \Gamma_x x_t &= \Gamma_y y_{t-1} + \Gamma_\epsilon \epsilon_t + \Gamma_\eta \eta_t \\ \Gamma_y^0 y_t + \Gamma_x^0 E_t x_{t+1} &= \Gamma_y^1 y_{t-1} + \Gamma_x^1 x_t + \Gamma_\epsilon \epsilon_t + \Gamma_\eta \eta_t \end{aligned} \tag{11}$$

Sims Representation

Using the first equation, we can solve for x_t and get:

$$x_t = \Pi_y y_{t-1} + \Pi_\epsilon \epsilon_t + \Pi_\eta \eta_t \tag{12}$$

$$\tag{13}$$

Here, I denote through $\Pi_j = (\Gamma_x)^{-1}\Gamma_j$, with $j = \{y, \epsilon, \eta\}$. We also know that the expected value of the errors is zero, that is $E_t\epsilon_{t+1} = E_t\eta_{t+1}$ which implies that the above equation can be written as $E_t x_{t+1} = \Pi_y y_t$. We can thus rewrite the second equation now as:

$$A_0 y_t = A_1 y_{t+1} + B\epsilon_t + C\eta_t \tag{14}$$

Here, the coefficient matrices are defined as follows: $A_0 = \Gamma_y^0 + \Gamma_x^0\Pi_y$, $A_1 = \Gamma_y^1 + \Gamma_x^1\Pi_y$, $B = \Gamma_\epsilon + \Gamma_x^0\Pi_\epsilon$ and $C = \Gamma_\eta + \Gamma_x^0\Pi_\eta$.

RBC Calibration (Example Numbers)

```
alpha = 0.40;  delta = 0.025;  beta = 0.988;
rho   = 0.95;  hs    = 0.31;   sz   = 0.002
# SS ratios (log-linearization uses these):
ysk = (1 - beta*(1 - delta)) / (alpha*beta)
ksy = 1/ysk
si  = delta/ysk
sc  = 1 - si
```

These are the same steady-state helpers used before the Sims block construction.

Build the Coefficient Blocks

```
# Dimensions
ny, nx = 3, 4      # y_t = [k̂_{t+1}, â_t, E_t ĉ_{t+1}],
# x_t = [ŷ, ĉ, î, ĥ]
ne, nn = 1, 1      # eps, expect. error

# Initialize (zeros then fill)
UX = zeros(nx, nx); UY = zeros(nx, ny)
UE = zeros(nx, ne); UN = zeros(nx, nn)

G0Y = zeros(ny, ny); G1Y = zeros(ny, ny)
G0X = zeros(ny, nx); G1X = zeros(ny, nx)
GE  = zeros(ny, ne)

# ...fill from linearized equations (omitted here for brevity) ...
```

From Static Block to Reduced Form

```
# Solve static block: x_t = Y_y y_{t-1} + Y_ε ε_t + Y_η η_t
Y_y = UX \ UY;
Y_e = UX \ UE;   Y_n = UX \ UN
# Expected x_{t+1} uses E_t y_{t+1} mapping
# Plug into dynamic block: A0 y_t = A1 y_{t-1} + B ε_t + C η_t
A0 = G0Y + G0X * Y_y
A1 = G1Y + G1X * Y_y
B  = GE  + G1X * Y_e
C  =       G1X * Y_n
```

QZ and Unique Stable Solution

Compute QZ of (A_0, A_1), order stable generalized eigenvalues first, then form the policy matrices. Sims' uniqueness hinges on the partitioned Schur factors (S, T, Q, Z) and the construction removing explosive u_t via forward iteration.

IRFs (Skeleton Code)

```
using Random, LinearAlgebra
Random.seed!(123)

nrep = 20
YS   = zeros(ny, nrep)  # [k̂_{t+1}, â_t, E_t ĉ_{t+1}]
XS   = zeros(nx, nrep)  # [ŷ, ĉ, î, ĥ]

# Suppose we computed MY (state transition for y) and
# ME, PIY, PIE (impact/measurement blocks)
Shock = 1.0
YS[:,1] .= ME * Shock
XS[:,1] .= PIE
for t in 2:nrep
YS[:,t] .= MY  * YS[:,t-1]
XS[:,t] .= PIY * YS[:,t-1]
end
```

Economic Reading of RBC IRFs

TFP shock raises output and investment on impact; consumption smooths; hours rise if intratemporal substitution dominates. Persistence in a_t drives hump shapes via capital dynamics and expectations.

Notes on Numerical Methods

- Scale variables (log-deviations) and check residuals of linear equations.
- Verify eigenvalue counts before policy recovery; ensure correct ordering.
- Seed RNG for reproducible IRFs; test horizons and shock sizing.

Baseline NK System (Log-linear)

The log-linearized version of the model, consisting in an IS curve, a New Keynesian Phillips curve, a Taylor rule for monetary policy as well as a AR process for technological shocks, is outlined below:

$$\tilde{x}_t = E_t\tilde{x}_{t+1} - \sigma^{-1}(\tilde{i}_t - E_t\tilde{\pi}_{t+1}) \tag{15}$$

$$\tilde{\pi}_t = E_t\tilde{\pi}_{t+1} + \kappa\tilde{x}_t \tag{16}$$

$$\tilde{i}_t = \delta\tilde{\pi}_t + v_t \tag{17}$$

$$v_{t+1} = \rho v_t + \epsilon_{t+1} \tag{18}$$

LRE Form and BK

Here, the parameter κ is given by $\frac{(1-\omega)(1-\beta\omega)}{\alpha\omega}$. Equations (3.146) and (3.148) and can grouped together and the model rewritten in a more convenient manner as follows:

$$E_t\tilde{x}_{t+1} + \sigma^{-1}E_t\tilde{\pi}_{t+1} = \tilde{x}_t + \sigma^{-1}\delta\tilde{\pi}_t + \sigma^{-1}v_t \tag{19}$$

$$E_t\tilde{\pi}_{t+1} = \tilde{\pi}_t - \kappa\tilde{x}_t \tag{20}$$

$$v_t = \rho v_{t-1} + \epsilon_{t+1} \tag{21}$$

This can be represented using the already familiar representation of DSGE models from (3.76):

$$A_0E_tX_{t+1} = A_1X_t + B_0v_{t+1} \tag{22}$$

BK/QZ Solution Sketch (NK)

```
using LinearAlgebra

# Build A,B from linearized NK (controls first or states first
# consistently with your BK routine). Then QZ:
S,T,Q,Z = qz(A, B)          # generalized Schur
λ = diag(T)./diag(S)
# Order eigenvalues so |λ|<1 first, then partition
# Recover policy x_t = P s_t, s_{t+1} = M s_t + N ε_{t+1}
# (algebra as in your BK helper)
```

Policy and Taylor Principle

With $\phi_\pi > 1$, the interest rule leans against inflation enough to push the unstable root count up to the number of jumps—delivering determinacy. Failing $\phi_\pi > 1$ typically yields indeterminacy (provided there is no interest rate smoothing).

Minimal NK Code Skeleton

```
# Parameters
β   = 0.99;  σ = 1.0;  κ = 0.1
π   = 1.5;   ϕx = 0.5
ρr  = 0.8;   σr = 0.002

# State: s_t = [r^n_t], Control: x_t = [x_t, π_t, i_t]
# Build A,B,C from the 3 equations; then solve via BK/QZ
# ... (assemble matrices) ...
S,T,Q,Z = qz(A,B)
# order, partition, recover (P,M,N)
```

Simulating NK IRFs

```
T = 40
s = zeros(1, T)          # r^n shock state
x = zeros(3, T)          # [x, π, i]
s[1] = σr                # one-std shock
for t in 1:T-1
x[:,t] .= P * s[:,t]
s[:,t+1] .= M * s[:,t]  # AR(1) embedded in M
end
# plot x[1,:], x[2,:], x[3,:]
```

Economic Reading of NK IRFs

Taylor rule lifts i_t more than one-for-one if $\phi_\pi > 1$, output gap adjusts; forward-looking terms shape the persistence via expectations.

When BK Fails (Quick Diagnostics)

- **Too few unstable roots**: likely indeterminacy (e.g., $\phi_\pi \leq 1$).
- **Too many unstable roots**: no stable solution given states.
- Check block invertibility (rank condition).

Numerical Tips (NK)

- Keep a consistent state/control ordering across A,B blocks.
- Use small helper functions for stacking equations.
- Validate with a known calibration before expanding shocks/rules.

Wrap-Up

We reviewed scalar and multivariate stochastic difference equations, solved a RBC model with Sims' QZ framework, and a basic NK with BK approach