ECON 413 — Project: Policy Lab

gEcon Directions (Calibration → Solution → IRFs)

Business Cycles and Growth

Presentation in class: Tuesday, December 2

Goal and deliverables

Choose **one** sample DSGE/RBC model from the gEcon "Sample models" page (*do not* use the basic RBC). Calibrate parameters (no estimation), solve, and generate impulse response functions (IRFs) for a shock of your choice. Your deliverables:

1. A GitHub repo with a runnable script (`code/run_all.R`), your `.gcn` model file, saved IRF figures, and a one-page `README.md`.
2. A 10-minute slide deck (PDF) presented on **Tue, Dec 2**.

Your talk must explain: (i) why this model vs. the baseline RBC; (ii) *one* parameter not in the baseline RBC and how you calibrated it; (iii) the key IRFs.

Recommended model choices (examples)

From the gEcon models page: RBC with installation costs (`rbc_ic.gcn`), capacity utilization (`rbc_cu.gcn`), habit formation (`rbc_hf.gcn`), monopolistic competition (`rbc_mc.gcn`), two sectors (`rbc_ts.gcn`), home production (`home_production.gcn`), time-to-build (`ttb.gcn`), two-country risk sharing (`tc_rs.gcn`), Smets–Wouters '03 (`SW_03.gcn`). *Avoid* `rbc.gcn`.

Setup (one-time)

1. **Install gEcon (R-Forge)**: `install.packages("gEcon", repos="http://R-Forge.R-project.org")`.
2. **Download a model** (`.gcn` and its accompanying `.R`) from the gEcon "Sample models" page and place them in a `model/` folder in your project directory.

Repo layout (use this; ASCII only)

```
policy-lab-yourteam/
model/              # .gcn and any .model.log (do not track huge logs)
data/               # optional
code/
    run_all.R       # main script (calibrate -> solve -> IRFs)
    helpers.R       # optional utilities
plots/              # IRF figures auto-saved here
README.md           # 1 page: what/why/how + how to run
```

Minimal pipeline (edit and run `code/run_all.R`)

Important: This project uses *calibration only* (no estimation).

```
# ---- Packages: install & load quietly ----
req <- c("MASS","Matrix","Rcpp","nleqslv")
if (is.null(getOption("repos")) || getOption("repos")["CRAN"] %in% c(NA,"@CRAN@")) {
  options(repos = c(CRAN = "https://cran.r-project.org"))
}
for (p in req) if (!requireNamespace(p, quietly = TRUE)) install.packages(p, quiet = TRUE
    )
if (!requireNamespace("gEcon", quietly = TRUE)) {
  install.packages("gEcon", repos = "http://R-Forge.R-project.org", quiet = TRUE)
}
suppressPackageStartupMessages({
  library(gEcon)
  library(MASS); library(Matrix); library(Rcpp); library(nleqslv)
})

# ---- Paths & model choice (EDIT THESE) ----
model_file <- file.path("model","rbc_hf.gcn")  # e.g., habit formation; choose your model
irf_vars    <- c("Y","C","I","K_s","L_s")       # EDIT to match your model's variable
    names
irf_horizon <- 40                                # IRF length in periods
shock_name  <- "epsilon_Z"                       # EDIT to one present in your model

# ---- 1) Load model & inspect names ----
mod <- make_model(model_file)
# Optional: see variables/parameters/shocks available:
# var_info(mod); par_info(mod); shock_info(mod, all = TRUE)

# ---- 2) Calibrate (set free parameters) ----
# Example: habit parameter h; replace with a parameter from YOUR model
# Check the model PDF on the sample-models page for parameter names.
# mod <- set_free_par(mod, list(h = 0.70))

# ---- 3) Steady state & linear solution ----
mod <- steady_state(mod)         # nonlinear steady state solver
mod <- solve_pert(mod, loglin = TRUE)  # 1st-order (log-)linear solution
check_bk(mod)                    # check Blanchard--Kahn conditions

# ---- 4) Choose shock variance/covariance ----
# Set sd for your target shock. Use shock_info(mod, all=TRUE) to list names.
mod <- set_shock_distr_par(
  model = mod,
  distr_par = setNames(list(0.10), sprintf("sd(%s)", shock_name)) # sd = 0.10 as example
)

# ---- 5) Compute & save IRFs ----
irf_obj <- compute_irf(model = mod, variables = irf_vars, sim_length = irf_horizon)
dir.create("plots", showWarnings = FALSE)
plot_simulation(irf_obj, to_eps = TRUE)  # saves EPS to a /plots subfolder near your .gcn

# ---- 6) (Optional) Model statistics for your README ----
```

```
# mod <- compute_model_stats(model = mod, n_leadlags = 6, ref_var = "Y")
# get_model_stats(mod)
```

Notes.

- `var_info(mod)` and the model's own PDF (from the sample models page) tell you the correct variable names to plot.
- Use `set_free_par()` to calibrate a parameter that does not exist in the baseline RBC (e.g., habits, installation cost curvature, Calvo stickiness, mark-up elasticity). Briefly justify your chosen value in the README and your slides (e.g., a target moment or a citation).

Troubleshooting

- **Steady state fails:** provide better guesses with `initval_var()` (variables) and `initval_calibr_par()` (calibrated parameters); then rerun `steady_state()`.
- **BK fails:** run `check_bk(mod)` after `solve_pert()` to see eigenvalue counts; re-check timing or parameter choices (e.g., too extreme persistence).
- **Shock names:** `shock_info(mod, all=TRUE)` lists them; set variances via `set_shock_distr_par()`.
- **IRFs did not save:** ensure you called `plot_simulation(irf_obj, to_eps=TRUE)` and that the model's `/plots` subfolder exists and is writable.

Submission checklist

1. `code/run_all.R` runs end-to-end without edits on a clean machine.
2. `plots/` contains exported IRFs with axes labels and clear legends.
3. `README.md` (one page): model choice, what it adds vs. RBC, one new parameter (definition + calibration choice), how to run, and what the figures show.