# Simulations

# Deterministic vs Stochastic Model

Dynare's general model frameworks

- deterministic model framework: $f(y_{t-1}, y_t, y_{t+1}, u_t | \theta) = 0$
- stochastic model framework: $E_t f(y_{t-1}, y_t, y_{t+1}, u_t | \theta) = 0,$ $u_{t+1} = \sigma \varepsilon_{t+1}$
  decision rule/policy function: $y_t = g(y_{t-1}, u_t, \sigma | \theta),$ $\varepsilon_t \sim N(0, \Sigma)$

Dynare computes the solution of

- deterministic models to arbitrary precision
- stochastic models based on perturbation approximation of policy function

# When to use which framework?

Deterministic simulation

- perfect foresight assumption
- useful to study: full implications of non-linearities, reaction to both contemporaneous and anticipated shocks, transition to new equilibrium

- Stochastic Simulation
  - 1st order: certainty equivalence: today's decisions don't depend on future uncertainty
  - higher-order: motive for precautionary savings or risk premia, as future uncertainty (future shocks) and nonlinear relationships are taken into account
  - Perturbation only valid in the vicinity of the steady-state, can be totally wrong otherwise
  - useful to study: transmission mechanisms of stochastic shocks, importance of shocks, estimation

# Deterministic Simulation in Dynare

- `initval`: for the initial steady state (followed by `steady`)
- `endval`: for the terminal steady state (followed by `steady`)
- `histval`: for initial or terminal conditions out of steady state
- `shocks`: for shocks along the simulation path
- `perfect_foresight_setup`: prepare the simulation
- `perfect_foresight_solver`: compute the simulation
- `simul`: old syntax, alias for perfect_foresight_setup + perfect_foresight_solver

# Deterministic Simulation in Dynare

Paths of exogenous and endogenous variables are stored in:

- $oo\_.endo\_simul = (y_0 \; y_1 \; \dots \; y_T \; y_{T+1})$
- $oo\_.exo\_simul = (\; u_1 \; \dots \; u_T \;)'$

Perfect_foresight_setup initializes those matrices, given the `shocks`, `initval`, `endval` and `histval` blocks

- $y_0$, $y_{T+1}$ and $u_1, \dots, u_T$ are the constraints of the problem
- $y_1, \dots, y_T$ are the initial guess for the Newton algorithm

Perfect_foresight_solver replaces $y_1, \dots, y_T$ in `oo_.endo_simul` by the solution

Initial guess for Newton algorithm can be manipulated after perfect_foresight_setup, but before perfect_foresight_solver

# Stochastic Simulation in Dynare

`shocks:` declare (co-)variance of Gaussian distribution

```
stoch_simul(order=1,irf=30,periods=0) y c iv;
```

approximate policy function at first order, compute impulse-response-function, theoretical moments and variance decomposition, print/plot only for y, c, iv

```
stoch_simul(order=3,irf=0,periods=300);
```

approximate policy function at third order, compute a simulation for 300 periods, empirical moments and variance decomposition, print/plot all variables