Dynamic Model

Core Design

# dynamicSolver.solve

The solve method of the dynamicSolver class is the core execution engine of the Penn Wharton Budget Model dynamic model. The method is capable of performing five modes of dynamic model execution:

1. Steady state baseline scenario
2. Open economy transition path baseline scenario
3. Open economy transition path counterfactual scenario
4. Closed economy transition path baseline scenario
5. Closed economy transition path counterfactual scenario

Shared storage parallel execution is supported and further parallelization is utilized within the method when executed on compatible platforms. A summary of the method’s internal operations follows.

## Initialization

### Parameter management

Parameters defining both the baseline scenario and optional counterfactual scenario are unpacked. Default values are set for unspecified parameters. Basic parametrically derived values are computed.

### Parallel execution setup

On multiprocessor platforms, execution environment setup is performed to allow for cohort‑level parallelization within generate­\_aggregates when used to generate static and dynamic transition path aggregates.

## Static aggregate generation

*Only for transition path counterfactuals*

### Baseline execution

A recursive call to dynamicSolver.solve is placed in order to generate the corresponding baseline transition path solution if it is not already available.

### Aggregate generation

generate\_aggregates is called to generate static aggregates using the baseline optimal decision values and population distributions as a precomputed solution.

### Additional aggregate calculations

Additional static aggregates are calculated.

## Dynamic aggregate generation

### Steady state execution

*Only for transition paths*

A recursive call to dynamicSolver.solve is placed in order to generate the corresponding steady state solution if it is not already available. The steady state solution is used to initialize the transition path.

### Open economy execution

*Only for closed economy transition paths*

A recursive call to dynamicSolver.solve is placed in order to generate the corresponding open economy transition path solution if it is not already available. The open economy solution is necessary to compute the government expenditure adjustments for the closed economy transition path.

### Solution iterations

Iterations are performed to identify a solution that satisfies the market clearing condition.

#### Price definitions

Prices are defined based on either initial values or the dynamic aggregates generated in the previous iteration.

#### Aggregate generation

generate\_aggregates is called to generate dynamic aggregates using the prices defined above.

#### Additional aggregate calculations

Additional dynamic aggregates are calculated.

#### Market clearing check

The market clearing condition is checked to determine whether a satisfactory solution has been reached or if further iterations should be performed.

## Elasticity calculation

*Only for steady state*

### Capital to output ratio calculation

The capital to output ratio is calculated using the dynamic aggregates generated above.

### Labor elasticity calculation

Labor elasticity is calculated using the dynamic aggregates generated above.

### Savings elasticity calculation

Savings elasticity is calculated using the dynamic aggregates generated above as well as perturbed aggregates generated with generate\_aggregates and marginally perturbed prices.

# generate\_aggregates

The generate\_aggregates function nested within dynamicSolver.solve performs the key operation of generating the static and dynamic aggregates. For static aggregates, the function accepts sets of precomputed optimal decision values and population distributions for use in aggregate generation. For dynamic aggregates, the function calculates optimal decision values and corresponding population distributions before performing aggregate generation. On multiprocessor platforms, the function makes use of cohort‑level parallelization for transition paths. A summary of the function’s internal operations follows.

## Initialization

*Only for dynamic aggregates*

### Initial distribution determination

Initial population distributions for all cohorts are determined, with transition path distributions drawing from the corresponding steady state distributions.

### Terminal utility determination

Terminal utility values for all cohorts are determined by solving the dynamic optimization problem for the steady state / post‑transition path cohort through backward induction from a zero utility terminal state over a full cohort lifetime.

## Cohort aggregate generation

### Dynamic optimization

*Only for dynamic aggregates*

Optimal decision values for each cohort are calculated by solving the cohort’s dynamic optimization problem through backward induction from the cohort’s terminal utility values over the modeling time period. Supporting functions that model preferences, taxation, and other elements are called to calculate available resources and utility.

### Property calculation

Population properties, such as tax on income, are calculated for each cohort using the cohort’s optimal decision values.

### Distribution generation

*Only for dynamic aggregates*

Population distributions for each cohort are generated through forward propagation from the cohort’s initial distribution using stochastic state transition probabilities and the cohort’s optimal decision values.

### Aggregate generation

Aggregates for each cohort are generated using the cohort’s population properties weighted by the cohort’s population distributions.

## Cross-cohort aggregation

Cohort aggregates are aggregated to produce economy‑wide static or dynamic aggregates for the modeling time period.