Dynamic Model

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 solving five forms of the dynamic model:

1. Steady states
2. Open economy transition path baselines
3. Open economy transition path counterfactuals
4. Closed economy transition path baselines
5. Closed economy transition path counterfactuals

Multiple instances of the method can be executed in parallel using a shared file system. Further parallelization is employed within the method when executed on multiprocessor 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. Scenario-independent parameters are loaded and derived.

### Internal parallelization 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

*For transition path counterfactuals only*

### Baseline execution

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

### Aggregate generation

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

### Additional aggregate calculation

Additional static aggregates are calculated based on the aggregates generated by generate\_aggregates.

## Dynamic aggregate generation

### Steady state execution

*For transition paths only*

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

### Open economy execution

*For closed economy transition paths only*

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

### Solution iterations

Iterations are performed to identify a solution—a set of market conditions—that satisfies a market clearing criterion.

#### Market condition definitions

Market conditions, including prices, are initialized or derived from the dynamic aggregates generated in the previous iteration.

#### Aggregate generation

generate\_aggregates is called to generate dynamic aggregates using the market conditions.

#### Additional aggregate calculation

Additional dynamic aggregates are calculated based on the aggregates generated by generate\_aggregates.

#### Market clearing evaluation

The market clearing criterion is evaluated by juxtaposing the dynamic aggregates with the market conditions.

## Elasticity calculation

*For steady states only*

### 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 in conjunction with marginally deviated aggregates generated by calling generate\_aggregates with prices marginally deviated from the steady state values.

# 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 a precomputed set of optimal labor values for optimal decision value calculation and a precomputed population distribution for use in aggregate generation. For dynamic aggregates, the function calculates optimal decision values using backward induction and generates a population distribution using forward propagation 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.

## Dynamic optimization

### Terminal utility determination

*For dynamic aggregates only*

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 dynamic optimization

*For transition paths only*

For static aggregates, optimal decision values for each cohort are derived from the precomputed optimal labor values. For dynamic aggregates, optimal decision values are calculated by solving the cohort’s dynamic optimization problem through backward induction from the cohort’s terminal utility values over the modeling period. Resource availability and utility are calculated using supporting functions that model agent preferences and taxation.

### Optimal decision value collation

Optimal decision values from all cohorts are collated to produce the complete set of optimal decision values for the population over the modeling period.

## Distribution generation

*For dynamic aggregates only*

### Initialization

For steady states, the population distribution is initialized with an arbitrary uniform distribution. For transition paths, the population distribution is initialized with the corresponding steady state distribution.

### Forward propagation

The population distribution is forward propagated by year using the optimal decision values found above. Population growth and decline due to births, deaths, and immigration flows are represented as explicit distributional changes. For steady states, forward propagation continues until the normalized distribution over ages satisfies an invariance criterion. For transition paths, forward propagation is performed across the years of the modeling period.

## Aggregate generation

Aggregates are calculated using the optimal decision values and population distribution found above. For steady states, the aggregates are scalar values. For transition paths, the aggregates are time series spanning the modeling period.