Global DSGE Models*

Dan Cao
Georgetown University

Wenlan Luo
Tsinghua University

Guangyu Nie
Shanghai University of Finance and Economics

March 2020

Abstract

In this paper, we introduce our GDSGE framework and MATLAB toolbox for solving dynamic stochastic general equilibrium models with a novel global solution method. The framework encompasses many well-known incomplete markets models with highly nonlinear dynamics such as models on financial crises, models with rare disasters, with many financial assets and portfolio choices, and with occasionally binding constraints. The toolbox allows users to input a simple and intuitive model description script similar to Dynare, and returns a convenient MATLAB interface for accessing efficient computations implemented in C++. The toolbox is most effective in solving models featuring endogenous state variables with implicit law-of-motion such as wealth shares or consumption shares. The toolbox solves many recent important models more efficiently and accurately compared to their original solution algorithms.

Keywords: nonlinear DSGE models, global solution method, computation toolbox, implicit law-of-motions, consistency equations

*An online compiler server of the toolbox is deployed at the toolbox's website: http://www.gdsge.com. For useful comments and discussions, we thank Toshi Mukoyama, John Rust, and generations of students in the Advanced Macro classes at Georgetown University and Tsinghua University. Dan thanks Shanghai University of Finance and Economics for hospitality during the completion of the paper and Georgetown Center for Economics Research (GCER) for financial support. Cristián Cuevas provided superb research assistance.

1 Introduction

The Dynamic Stochastic General Equilibrium (DSGE) models are an important tool in the study of business cycles and monetary and fiscal policies. The introduction of the toolbox Dynare has made it easy to solve and estimate DSGE models and has enabled a large number of important academic studies and policy applications. Dynare uses local algorithms to solve the models. However, recent developments in macroeconomics highlight the importance of solving these models using global methods. These developments include studies on

- financial crises and highly nonlinear dynamics of the economy around the crises in close or open economies such as Mendoza (2010), Bianchi (2011), He and Krishnamurthy (2011), Brunnermeier and Sannikov (2014), and Cao et al. (2019);
- implications of rare disasters such as Barro (2006), Gourio (2012), and Barro et al. (2017);
- portfolio choices and their implications such as Heaton and Lucas (1996), Guvenen (2009), and Cao (2018);
- models with occasionally binding constraints (e.g, borrowing constraints and monetary policy zero lower bound) such as Gust et al. (2017), Guerrieri and Iacoviello (2017), Cao and Nie (2017), and Cao et al. (2019);
- international finance models with endogenous capital accumulation and/or portfolio choices such as Caballero et al. (2008), Maggiori (2017), Coeurdacier et al. (2019), and Cao et al. (2020);
- and many more.

Yet, despite these important developments, there has not been an unified framework and a toolbox like Dynare for the global solutions of DSGE models. This paper offers such a framework and toolbox.

In this paper, we first develop a general framework that encompasses many recent well-known models and their extensions. The framework allows us to design a general algorithm to solve these models robustly and efficiently using policy-function iterations. We then develop a toolbox that implements the algorithm. The toolbox is similar to Dynare in that it allows users to write models in intuitive and simple scripts, i.e., gmod files (gmod standing for global model), despite that users need to specify the state and policy variables explicitly, due to the nature of global solutions.

The algorithm is based on policy function iteration, collocation, and global projection. One well-known challenge for global solution methods, including ours, is that the equilibrium equation system needs to be solved for a large number of collocation points across the state space, requiring researchers to turn to a compiled language such as C++ or Fortran to make computations feasible. The toolbox addresses this challenge by compiling the model description file into a C++ library that implements the actual computations with high efficiency, while returning a convenient MATLAB interface to users. The low-level implementation takes care of details such as interfacing to multiple equation solvers, dense/sparse grid function approximation methods, automatic differentiation, and parallel computation, while remains flexible by allowing users to specify options and generate model output via the MATLAB interface.

We provide many examples of existing seminal applications that can be solved very easily using the toolbox. The examples in the paper include Heaton and Lucas (1996), Guvenen (2009), Bianchi (2011), and Barro et al. (2017). Each of the examples listed can be implemented within 200 lines of toolbox codes and execute in a minute on a regular laptop. The toolbox solves these examples more efficiently and accurately compared to their original solution methods. We provide many more examples on the toolbox's website.

The toolbox demonstrates the most of its power, relative to other methods, for models with endogenous state variables with implicit state-transition equations, such as wealth shares or consumption shares. As we make clear in the applications, these endogenous state variables help reduce the number of state variables to be kept track of in models with many assets such as Heaton and Lucas (1996), Kubler and Schmedders (2003), and Cao (2018), or get around multiple equilibria issues such as Cao et al. (2019). The key insight which allows us to integrate these models in our framework is to include the vectors of future realization of endogenous state variables in the vector of policy variables. The additional equations in the system of equations and unknowns, to be solved in each collocation point over the iterations, are the *consistency equations* that impose the future endogenous state variables to be consistent with current policy variables.

Our approach to solving models with endogenous state variables is different from existing approaches in the literature. For example, Kubler and Schmedders (2003) use wealth shares as endogenous state variables. They solve for future wealth shares using consistency equations as an additional fixed-point problem for each guess for current policy variables. The solution to the fixed-point problem is then used to formulate a system of equations and unknowns for current policy variables. By contrast, we directly include future wealth shares and consistency equations among the policy variables and equilibrium conditions. This allows us to solve for equilibrium at the current state variables in a single-step and facilitate the implementation of the toolbox.

An earlier attempt in providing a general, unified framework for global solution of DSGE models is Winschel and Kratzig (2010). Our framework is more general and allows for endogenous state variables with implicit state-transition equations. We also provide a Dynare like toolbox which only requires users to provide model files. Users do not need to code up their model in specific programming languages like Java, Fortran, or MATLAB.

The framework is more readily applicable to solving GDSGE models with a finite number of agents, or more precisely a finite number of agent-types.[1] Cao (2020) shows that incomplete markets models with finite agent types are useful special cases of fully-heterogeneous-agent, incomplete markets model with both idiosyncratic and aggregate shocks à la Krusell and Smith (1998). In particular, the former corresponds to the latter in which idiosyncratic shocks are perfectly persistent. We provide an explicit comparison between the two models in the toolbox's website. In addition, the toolbox can be used to solve the agents' decision problem and to simulate in the latter given conjectured laws of motion of the aggregate variables. Then, with an additional fixed-point iteration on these laws of motion, which can be coded up simply in MATLAB, the toolbox solution can be used to solve for the DSGE in the latter. In the last section of the paper, we show how this idea can be used to solve Krusell and Smith's model in less 100 lines of toolbox code and 100 lines of MATLAB code.

The remainder of the paper is organized as follows. In Section 2, we present the leading example for our toolbox. In Section 3 and Section 4, we provide the general framework and the design of the toolbox. A wide range of examples is presented in Section 5. In Section 6 we discuss the application of our toolbox to heterogenous agent models with both idiosyncratic and aggregate shocks. Section 7 concludes.

2 A Leading Example

We use the benchmark model in Heaton and Lucas (1996) as the first illustration for how to write models in our framework and solve them using the toolbox. We follow closely the notation in the original paper.

This is an incomplete market model with two representative agents $i \in \mathcal{I} = \{1,2\}$ who trade in equity shares and bonds. The aggregate state $z \in \boldsymbol{Z}$, which consists of

[1]There is a continuum of price-taking agents within each type and they make identical decisions in equilibrium.

capital income share, agents' income share, and aggregate endowment growth, follows a first-order Markov process. $p_t^s(z^t)$ and $p_t^b(z^t)$ denote share price and bond price at time t and in shock history $z^t = \{z_0, z_1, \ldots, z_t\}$. To simplify the notations, we omit the explicit dependence on shock history.

Agent i takes the share and bond prices as given and maximizes her inter-temporal expected utility

$$\mathcal{U}_t^i = \mathbb{E}_t \left[\sum_{\tau=0}^{\infty} \beta^\tau \frac{\left(c_{t+\tau}^i\right)^{1-\gamma}}{1-\gamma} \right]$$

subject to

$$c_t^i + p_t^s s_{t+1}^i + p_t^b b_{t+1}^i \leq (p_t^s + d_t) s_t^i + b_t^i + Y_t^i$$

and

$$\begin{aligned} s_{t+1}^i &\geq 0 \\ b_{t+1}^i &\geq K_t^b, \end{aligned}$$

where Y_t denotes the aggregate income. $d_t = \delta_t Y_t^a$ is total dividend (capital income) and $Y_t^i = \eta_t^i Y_t^a$ is labor income of agent i. Aggregate income grows at a stochastic rate $\gamma_t = \frac{Y_t^a}{Y_{t-1}^a}$. $z_t = \{\gamma_t^a, \delta_t, \eta_t^1\}$ follows a first-order Markov process estimated using U.S. data. The borrowing limit is set to be a constant fraction of per capita income, i.e., $K_t^b = \bar{K}^b Y_t$.

In equilibrium, prices are determined such that markets clear in each shock history:

$$\begin{aligned} s_t^1 + s_t^2 &= 1, \\ b_t^1 + b_t^2 &= 0. \end{aligned}$$

As in Kubler and Schmedders (2003) and Cao (2010, 2018), we use the normalized financial wealth share

$$\omega_t^i = \frac{(p_t^s + d_t) s_t^i + b_t^i}{p_t^s + d_t}$$

as an endogenous state variable. In equilibrium, the market clearing conditions imply that $\omega_t^1 + \omega_t^2 = 1$.

For any variable x_t, let $\hat{x}_t$ denote the normalized variable: $\hat{x}_t = \frac{x_t}{Y_t}$ (except b_t^i for which $\hat{b}_t^i = \frac{b_t^i}{Y_{t-1}^a}$). Using this normalization, agent i's budget constraint can be rewritten as

$$\hat{c}_t^i + \hat{p}_t^s s_{t+1}^i + p_t^b \hat{b}_{t+1}^i \leq \left(\hat{p}_t^s + \hat{d}_t\right) \omega_t^i + \hat{Y}_t^i.$$

The wealth share is rewritten as

$$\omega_t^i = \frac{(\hat{p}_t^s + \hat{d}_t)s_t^i + \frac{\hat{b}_t^i}{\gamma_t^a}}{\hat{p}_t^s + \hat{d}_t}.$$

The optimality of agent i's consumption and asset choices are captured by first-order conditions in s_{t+1}^i and b_{t+1}^i:

$$\begin{aligned}
1 &= \beta \mathbb{E}_t \left[\left(\frac{\hat{c}_{t+1}^i}{\hat{c}_t^i} \right)^{-\gamma} \left(\gamma_{t+1}^a\right)^{1-\gamma} \frac{\hat{p}_{t+1}^s + \hat{d}_{t+1}}{\hat{p}_t^s} \right] + \hat{\mu}_t^{i,s} \\
1 &= \beta \mathbb{E}_t \left[\left(\frac{\hat{c}_{t+1}^i}{c_t^i} \right)^{-\gamma} \left(\gamma_{t+1}^a\right)^{-\gamma} \frac{1}{p_t^b} \right] + \hat{\mu}_t^{i,b},
\end{aligned}$$

where $\hat{\mu}_t^{i,s}$ and $\mu_t^{i,b}$ are the Lagrangian multipliers on agent i's no short sale constraint and borrowing constraint, respectively. The multipliers and portfolio choices satisfy the complementary-slackness conditions:

$$\begin{aligned}
0 &= \hat{\mu}_t^{i,s} s_t^i \\
0 &= \hat{\mu}_t^{i,b} (\hat{b}_t^i + \bar{K}^b).
\end{aligned}$$

Because the optimization problems of the agents are concave optimization problems. The first-order conditions are necessary and sufficient for optimality.

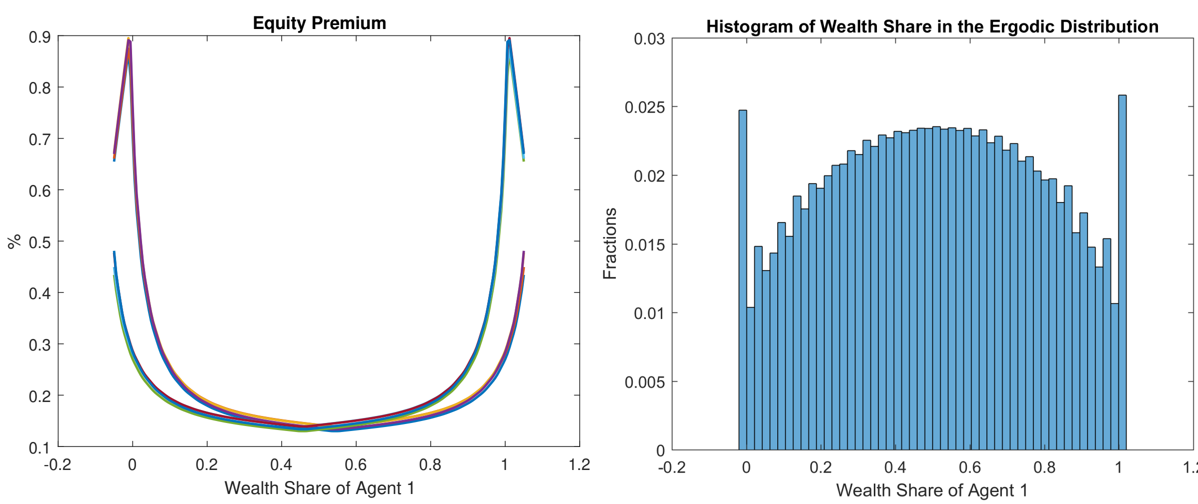
We solve the model using policy function iterations: we look for pricing, allocation, and Lagrange multiplier functions over wealth share which satisfy the market clearing conditions and first-order conditions. The GDSGE code for the model and implements our algorithm is given in the next page.

```
% Parameters
parameters beta gamma Kb;
beta = 0.95;  % discount factor
gamma = 1.5;  % CRRA
Kb = -0.05;   % borrowing limit in ratio of aggregate output
% Shock variables
var_shock g d eta1;
% Shocks and transition matrix
shock_num = 8;
g = [.9904 1.0470 .9904 1.0470 .9904 1.0470 .9904 1.0470];
d = [.1402 .1437 .1561 .1599 .1402 .1437 .1561 .1599];
eta1 = [.3772 .3772 .3772 .3772 .6228 .6228 .6228 .6228];
shock_trans = [
  0.3932 0.2245 0.0793 0.0453 0.1365 0.0779 0.0275 0.0157
  0.3044 0.3470 0.0425 0.0484 0.1057 0.1205 0.0147 0.0168
  0.0484 0.0425 0.3470 0.3044 0.0168 0.0147 0.1205 0.1057
  0.0453 0.0793 0.2245 0.3932 0.0157 0.0275 0.0779 0.1365
  0.1365 0.0779 0.0275 0.0157 0.3932 0.2245 0.0793 0.0453
  0.1057 0.1205 0.0147 0.0168 0.3044 0.3470 0.0425 0.0484
  0.0168 0.0147 0.1205 0.1057 0.0484 0.0425 0.3470 0.3044
  0.0157 0.0275 0.0779 0.1365 0.0453 0.0793 0.2245 0.3932
  ];
shock_trans = shock_trans ./ sum(shock_trans,2); % Normalize
% State variables
var_state w1;  % wealth share
w1 = linspace(-0.05,1.05,201);

% Endogenous variables, bounds, and initial values
var_policy c1 c2 s1p nb1p nb2p ms1 ms2 mb1 mb2 ps pb w1n[8];
inbound c1 1e-12 1;
inbound c2 1e-12 1;
inbound s1p 0.0 1.0;
inbound nb1p 0.0 1.0;   % nb1p=b1p-Kb
inbound nb2p 0.0 1.0;
inbound ms1 0 1;        % Multilier for constraints
inbound ms2 0 1;
inbound mb1 0 1;
inbound mb2 0 1;
inbound ps 0 10;
inbound pb 0 10;
inbound w1n 0 1;
% Extra output variables
var_aux equity_premium;
% Interpolation objects
var_interp ps_future pb_future c1_future c2_future;
initial ps_future 0.0;
initial pb_future 0.0;
initial c1_future w1.*d+eta1;
initial c2_future (1-w1).*d+1-eta1;
ps_future = ps;
pb_future = pb;
c1_future = c1;
c2_future = c2;

model;
  % Interpolation
  [psn',pbn',c1n',c2n'] = GDSGE_INTERP_VEC'(w1n');
  % Expectations in Euler Equations
  es1 = GDSGE_EXPECT{g'^(1-gamma)*(c1n'/c1)^(-gamma)*(psn'+d')/ps};
  es2 = GDSGE_EXPECT{g'^(1-gamma)*(c2n'/c2)^(-gamma)*(psn'+d')/ps};
  eb1 = GDSGE_EXPECT{g'^(-gamma)*(c1n'/c1)^(-gamma)/pb};
  eb2 = GDSGE_EXPECT{g'^(-gamma)*(c2n'/c2)^(-gamma)/pb};
  % b transformation
  b1p = nb1p + Kb;  % Transform bond back
  b2p = nb2p + Kb;
  s2p = 1-s1p;      % Market clear of shares
  % Budget constraint
  budget_1 = w1*(ps+d+1) - c1 - ps*s1p - pb*b1p;
  budget_2 = (1-w1)*(ps+d+1) - c2 - ps*s2p - pb*b2p;
  % Consistency
  w1_consis' = (s1p*(psn'+d') + b1p/g' + eta1')/(psn'+d'+1) - w1n';
  % Extra output
  equity_premium = GDSGE_EXPECT{(psn'+d')/ps*g'} - 1/pb;
  equations;
    -1+beta*es1+ms1;
    -1+beta*es2+ms2;
    -1+beta*eb1+mb1;
    -1+beta*eb2+mb2;
    ms1*s1p;
    ms2*s2p;
    mb1*nb1p;
    mb2*nb2p;
    b1p+b2p;
    budget_1;
    budget_2;
    w1_consis';
  end;
end;

simulate;
  num_periods = 50000;
  num_samples = 6;
  initial w1 0.5;
  initial shock 1;
  var_simu c1 c2 ps pb equity_premium;
  w1' = w1n';
end;
```

The GDSGE code solves for the equilibrium prices and allocation as functions of exogenous, z_t and endogenous state variables ω_t. A key innovation in our algorithm that enables the implementation using the toolbox is that we incorporate consistency equations (line 71 in the GDSGE code) into the system of equations and unknowns. These equations require that the conjectured future endogenous state variables are consistent with the current portfolio choices and future prices:

$$\omega_{t+1}^1 = \frac{(\hat{q}_{t+1}(z_{t+1}, \omega_{t+1}^1) + d_{t+1})k_{t+1}^1 + \hat{b}_{t+1}^1 / g_{t+1}}{\hat{q}_{t+1}((z_{t+1}, \omega_{t+1}^1) + d_{t+1}}.$$

The code produces the policy functions including equilibrium prices and allocation as functions of the endogenous state variable, wealth share ω^1, and exogenous state variable z. Panel (a) in Figure 1 shows the equity premium (the difference between expected stock and bond returns) as a function of wealth share and for different combination of exogenous state variables. The kinks in the equity premium function appear at points where the borrowing and short-sale constraints switch from being binding to non-binding, or vice versa, as ω_t increases. Panel (b) in Figure 1 shows the ergodic distribution of the endogenous state variable, ω^1.



(a) The Policy Function for Equity Premium (b) Ergodic Distribution of the Endogenous State

Figure 1: Ergodic Distribution and Policy Functions

Note: The model is solved with 8 realizations of exogenous states, 201 fixed grid points for the endogenous state. The histogram is based on 6 sample paths, 50,000 period simulations per sample path, with the first 10,000 periods dropped (burn-in periods).

The model can also be solved using consumption share instead of wealth share, as in Bernard and Lyasoff (2012). In this case, the consistency equations correspond to agents' future budget constraints: future consumption shares should be consistent with current portfolio choices and future portfolio choices, which in turn depend on future consumption shares. Bernard and Lyasoff (2012) call these equations "marketability conditions." Our algorithm is more general and does not rely on their "kernel conditions" which are derived by assuming the agents' Euler equations hold exactly. Our algorithm allows for deviation from the Euler equations due to binding portfolio constraints, such as borrowing constraint or short-selling constraint. The details of our implementation using GDSGE toolbox are provided in the toolbox's website.

3 General Environment

In this section we provide the general framework and the solution algorithm to compute recursive equilibrium in this framework. In the next section, Section 4, we present the design of the toolbox to implement the algorithm. In Section 5, we show that many recent important models fit exactly in the framework and, hence, can be solved using the toolbox. The toolbox's algorithm is different from the algorithms in their original papers.

3.1 Recursive Equilibrium and Solution Algorithm

We work with models for which the sequential competitive equilibrium of the economy can be characterized by a system of short-run equilibrium conditions:

$$\boldsymbol{F}(s, x, z, \{s'(z'), x'(z')\}_{z' \in \mathcal{Z}}) = 0 \tag{1}$$

where

$$z \in \mathcal{Z} \subset \mathbb{R}^{d_z}$$

is a vector of exogenous shocks;

$$s \in \mathcal{S} \subset \mathbb{R}^{d_s}$$

is a vector of endogenous states variables; and

$$x \in \mathcal{X} \subset \mathbb{R}^{d_x}$$

is a vector of endogenous policy variables. The function

$$\boldsymbol{F} : \mathbb{R}^{d_s+d_x+d_z} \times \left(\mathbb{R}^{d_s} \times \mathbb{R}^{d_x}\right)^Z \rightrightarrows \mathbb{R}^{d_s+d_x+d_z} \times \left(\mathbb{R}^{d_s} \times \mathbb{R}^{d_x}\right)^Z,$$

where Z is the cardinality of $\mathcal{Z}$, consists of optimality conditions, market clearing conditions, and laws of motion for state variables. The laws of motion can be explicit or implicit, as we discuss below.

Notice that the framework allow for general dependence on the future variables, instead of through common expectations as in Winschel and Kratzig (2010). This generality is important in allowing for non-rational expectations models such as model with belief heterogeneity such as Sandroni (2000), Blume and Easley (2006), Simsek (2013), and Cao (2018). It is also necessary to capture nonlinear forms of borrowing constraint such as the collateral constraints in Kiyotaki and Moore (1997), Geanakoplos (2010), and Cao and Nie (2017).[2]

Models with inequality constraints also fit into the general formulation (1) by adding additional endogenous policy functions. Indeed, if a recursive model has both equality and inequality conditions (such as the borrowing constraints in Heaton and Lucas (1996)):

$$\begin{aligned} \boldsymbol{F}\left(s, x, z, \left\{s'(z'), x'(z')\right\}_{z' \in \mathcal{Z}}\right) &= 0 \\ \boldsymbol{G}\left(s, x, z, \left\{s'(z'), x'(z')\right\}_{z' \in \mathcal{Z}}\right) &\geq 0, \end{aligned}$$

we can use

$$\hat{\boldsymbol{F}} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{G} - \eta \end{pmatrix}$$

with $\eta \geq 0$, and

$$\hat{x} = (x, \eta),$$

to write the system with inequality constraint in form (1) using $\hat{\boldsymbol{F}}$ and $\hat{x}$.

Definition A *recursive equilibrium* is a solution to (1) under the form

$$x = \mathcal{P}(z, s)$$

and

$$s'(z') = \mathcal{T}(z, z', s)$$

where $\mathcal{P}$ and $\mathcal{T}$ are equilibrium policy and transition functions, respectively.

[2] Collateral constraints might involve nonlinear functions of future asset prices (as random variables), beyond simple functions of expected prices such as the minimum of the price realizations over all possible future states. Cao and Nie (2017) provide a detailed comparison for different forms of collateral constraints.

A Colocation Policy Function Iteration Algorithm We solve for a recursive equilibrium of (1) using policy function iteration as follows.[3] The algorithm starts with an initial guess of policy and transition functions

$$\left\{\mathcal{P}^{(0)}(.,.), \mathcal{T}^{(0)}(.,.,.)\right\}$$

Given $\mathcal{P}^{(n)}$ and $\mathcal{T}^{(n)}$, $\mathcal{P}^{(n+1)}$ and $\mathcal{T}^{(n+1)}$ are determined by solving the following system of equations

$$\boldsymbol{F}\left(s, x, z, \left\{s'(z'), \mathcal{P}^{(n)}\left(z', s'(z')\right)\right\}_{z' \in \mathcal{Z}}\right) = 0. \tag{2}$$

with unknowns x and $\{s'(z')\}_{z' \in \mathcal{Z}}$ for each

$$(s, z) \in \mathcal{C}^{(n)} \subset \mathcal{Z} \times \mathcal{S}.$$

The set $\mathcal{C}^{(n)}$, which we call the set of collocation points, is a subset of $\mathcal{Z} \times \mathcal{S}$. We keep track of a distance between $\mathcal{P}^{(n)}, \mathcal{T}^{(n)}$ and $\mathcal{P}^{(n+1)}, \mathcal{T}^{(n+1)}$ over the iterations and stop when the distance falls below a preset threshold.

The typical initial guess for $\mathcal{P}^{(0)}$ that we use corresponds to the equilibrium in the 1-period economy. So the solution for $\mathcal{P}^{(n)}$ corresponds to the equilibrium values of the first period in the (n+1)-period economy. So the numerical limit of $\left\{\mathcal{P}^{(n)}\right\}$ corresponds to the finite-horizon limit. This limit is shown to be the equilibrium in the infinite horizon economies in existence proofs for infinite-horizon incomplete markets economy such as Duffie et al. (1994), Magill and Quinzii (1994), and Cao (2020).

Example For the model in Heaton and Lucas (1996) described above

$$z = (\gamma^a, \delta, \eta),$$

and

$$s = (\omega^1),$$

and

$$x = (\hat{c}^1, s^1, \hat{b}^1, \hat{c}^2, s^2, \hat{b}^2, p^s, p^b).$$

[3] Earlier work using policy-function iterations for DSGE economies includes Coleman (1990), Coleman (1991), and Judd et al. (2000).

3.2 More Detailed Representations

The system of equations in (1) represents different type of equilibrium conditions, including laws of motion for state variables and Euler-type first order conditions relating current and next period choices. These equations can be written more explicitly, as in Winschel and Kratzig (2010), for clarity. In some cases, they can be used to reduce the number of equations to be solved in each policy function iteration step.

3.2.1 Explicit and Implicit State Transitions

The state variables s may consist of state variables $\bar{s}$ which have explicit transition equations (law-of-motions), and state variables $\bar{\bar{s}}$ which consists of state variables with implicit transition equations: $s = (\bar{s}, \bar{\bar{s}})$. For $\bar{s}$, the law of motion can be written explicitly:

$$\bar{s}' = \bar{g}(s, x, z, z').$$

This is the specification in Winschel and Kratzig (2010). In our framework, we also allow for state variables $\bar{\bar{s}}$ with implicit laws of motion:

$$0 = \bar{\bar{g}}\left(s, x, z, \bar{\bar{s}}'(z'), x'(z'), z'\right).$$

Examples of state variables with implicit state transition includes wealth shares, as in Section 2 for Heaton and Lucas (1996), or consumption shares.

In this case, system of equation (1) can be written as

$$\boldsymbol{F}(s, x, z, \{s'(z'), x'(z')\}_{z' \in \mathcal{Z}}) = \begin{pmatrix} f\left(s, x, z, \{s'(z'), x'(z')\}_{z' \in \mathcal{Z}}\right) \\ \bar{s}' - \bar{g}(s, x, z, z') \\ \bar{\bar{g}}\left(s, x, z, \bar{\bar{s}}'(z), x'(z'), z'\right) \end{pmatrix}$$

In a recursive equilibrium, the last equation becomes

$$0 = \bar{\bar{g}}\left(s, x, z, \bar{\bar{s}}'(z'), \mathcal{P}(z', (\bar{g}(s, x, z, z'), \bar{\bar{s}}'(z'))), z'\right). \tag{3}$$

We call these equations *consistency equations*. It requires future state variables $\bar{\bar{s}}'(z')$ to be consistent with current policies and future policies implied by these future state variables and the policy function $\mathcal{P}$.

The state variables with explicit state transition allow us to reduce the number of equations and unknowns in each step of the policy function iteration algorithm described above. Indeed, in the policy function iteration algorithm we can work with $\bar{\boldsymbol{F}}$ which only takes the first and third components from $\boldsymbol{F}$:

$$\bar{\boldsymbol{F}}\left(s, x, z, \left\{\bar{\bar{s}}'(z'), \mathcal{P}^{(n)}\left(z', \left(\bar{g}(s, x, z, z'), \bar{\bar{s}}'(z')\right)\right)\right\}_{z' \in \mathcal{Z}}\right) = 0.$$

In this case, we solve for unknowns x and $\{\bar{\bar{s}}'(z')\}_{z' \in \mathcal{Z}}$ given future policy function $\mathcal{P}^{(n)}$. Consistency equations (3) become

$$\bar{\bar{g}}\left(s, x, z, \bar{\bar{s}}'(z), \mathcal{P}^{(n)}\left(z', \left(\bar{g}(s, x, z, z'), \bar{\bar{s}}'(z')\right)\right), z'\right) = 0.$$

3.2.2 Expectation Variables

Some of the policy functions include the expectation of the futures

$$x_t = (\bar{x}_t, e_t)$$

where

$$\begin{aligned} e_t &= \mathbb{E}_t h(s_t, \bar{x}_t, z_t, s_{t+1}, \bar{x}_{t+1}, z_{t+1}) \\ &= \sum_{z_{t+1}|z_t} \Pr(z_{t+1}|z_t) h(s_t, \bar{x}_t, z_t, s_{t+1}, \bar{x}_{t+1}, z_{t+1}). \end{aligned} \tag{4}$$

For example, in Section 2 for Heaton and Lucas (1996), e_t includes the expectation of asset returns weighted by agents' marginal utilities.

In this case, the system of equation, (1) can be more explicitly written as

$$\boldsymbol{F}(s, x, z, \{s'(z'), x'(z')\}_{z' \in \mathcal{Z}}) = \begin{pmatrix} \bar{\boldsymbol{F}}\left(s, (\bar{x}, e), z, \{s'(z'), (\bar{x}'(z'), e'(z'))\}_{z' \in \mathcal{Z}}\right) \\ e - \sum_{z' \in \mathcal{Z}} \Pr(z'|z') h(s, \bar{x}, z, s', \bar{x}', z') \end{pmatrix}.$$

In the policy function iteration algorithm we work with $\bar{\boldsymbol{F}}$ which takes the first component from $\boldsymbol{F}$:

$$\bar{\boldsymbol{F}}\left(s, (\bar{x}, e), z, \left\{s'(z'), \mathcal{P}^{(n)}\left(z', \left(\bar{g}(s, \bar{x}, e, z, z'), s'(z')\right)\right)\right\}_{z' \in \mathcal{Z}}\right) = 0$$

which consists of a fewer number of equations and unknowns than the original system. In the policy function iteration steps, we only need to solve for unknowns $\bar{x}$ and $\{s'(z')\}_{z' \in \mathcal{Z}}$.

4 The Design of the Toolbox

In this section, we described in detail how the toolbox is designed and implemented. The design of the toolbox is depicted in Figure 2. Users create and edit their own gmod file that describes the dynamic equilibrium of their model in the general form (1) of the general framework. Gmod stands for global model. The structure of the gmod file is given in Subsection 4.1. The gmod files can be uploaded to the toolbox's website and the toolbox compiles the files into MATLAB and C++ files which solve for recursive equilibria using policy function iterations and simulate the equilibrium dynamics. The functions of the complied files, which consist of solving system of equations, discretizing and interpolating policy functions, are described in Subsection 4.2

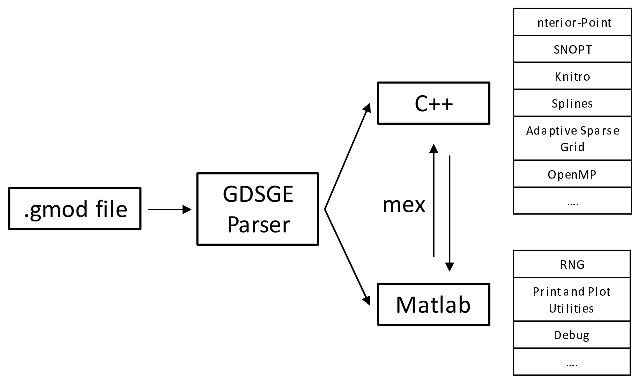


Figure 2: Toolbox Design and Implementations

4.1 User Inputs: the gmod Files

The toolbox asks users to provide gmod files which contain the equilibrium system (1) of their models. The gmod file, after processed by the toolbox, will return the policy and state transition functions from converged time iterations and the Monte-Carlo simulation samples. In this section, we provide the description for a minimum gmod file such as the one for the leading example in Section 2, and refer readers to the appendix and the toolbox's website for a detailed user manual. A minimum gmod file should contain the following components:

parameters. Exogenous parameters that do not vary across states or over time.

var_shock. Exogenous state variables z in system (1). These states need to be specified as discretized points.[4]

[4]To accommodate exogenous continuous shocks such as AR(1) processes, treat continuous shocks as endogenous state variables and approximate the shock processes with discretized innovations as exogenous states.

shock_num. The number of discretized points for *var_shock*. For multi-dimension *var_shock*, this should be the size of the Cartesian set across all dimensions.

shock_trans. The Markov transition matrix for exogenous state variables.

var_state. Endogenous state variables s in system (1). The toolbox requires users to specify the grid for each of these variables.[5]

var_policy. Policy variables x in system (1). For state variables with implicit laws of motion, we include vectors of these variables in future states among the policy variables.

var_aux. Some policy variables can be directly computed as relatively simple, explicit functions of other variables in x, s, x', s'. We use the keyword *var_aux* for these variables. We exclude them from the var_policy in order to reduce the number of equations and unknowns to be solved in each policy function iteration.

var_interp. These are policy variables x that appear in equilibrium system (1) as future states $x'(z')$. Even though the general formulation allows any policy variable in x to appear as a future state, in practice not all of them do. Here we only include those variables which need to be interpolated in the policy function iteration steps. When the time iteration converges, *var_interp* also delivers the state transition functions.

The updates of each *var_interp* after each time iteration should be specified after declaring the *var_interp*'s. The updates can use functions of solutions of policy variables in *var_policy* or *var_aux*, combining any parameters or exogenous states.

The model block. The model definition is enclosed in a block starting with *model;* and ending with *end;*. The model block should include an *equations* block in which each line represents one equation of the equilibrium system (1) to be solved. Other variables required to be evaluated in these equations should be put into the model block preceding the *equations* block. A variable followed by a prime (′) indicates that the variable is a vector of length *shock_num*, and it is usually used to represent future states z', or s' as in the general framework notations. The model block can use the following utility functions.

GDSGE_EXPECT. Calculate the conditional expectation of the object, such as e_t in equation (4), using the default transition matrix specified in *shock_trans*. This function can also accommodate a different transition matrix than *shock_trans* so that the toolbox can be used to solve models with heterogeneous beliefs (see Cao (2018) and the associated gmod file in the toolbox's website for an example).

GDSGE_INTERP_VEC. Evaluate function approximations specified in *var_interp*. This function, when followed by a prime (′), indicates that the approximation is evaluated for

[5]For fixed-grid-based function approximations such as splines, the grids will directly used; for adaptive grid method, the two end points of the grids will be used as the range of the state variable.

a vector of arguments of length *shock_num*; accordingly, the input and output variables in this case should also be followed by a prime. The output is thus a vector corresponding to $s'(z')$ or $x'(z')$ in system (1) for all possible realizations of exogenous states z'.

The simulate block. This optional block specifies the Monte Carlo simulations after the convergence of time iterations. It should specify *num_samples* for the number of sample paths, *num_periods* for the number of simulation periods of each path, *initial* for initial values of endogenous and exogenous states, *var_simu* for the variables to be recorded in the simulation, and the transitions for each endogenous state (the transition for exogenous states are handled automatically by the toolbox).

By default, the simulation resolves the system of equations (with $s'(z')$ and $x'(z')$ given by the converged policy and state transition functions) at each time step. This ensures the numerical error is minimum within a time step. We also implement a conventional fast albeit less accurate simulation method based on interpolating the policy and state transition functions directly. To use this method, the users should specify *SIMU_INTERP*=1 and declare interpolated variables in *var_output*. See the user manual in the appendix for details.

These simulations are important to compute stationary recursive equilibria, i.e., recursive equilibria with an ergodic distribution over the state variables, from which the model moments are calculated (the rigorous definition is provided in Duffie et al. (1994) and Cao (2020)). They can also be used to calculate nonlinear impulse response functions (see Cao and Nie (2017) and Cao et al. (2020) for examples) to understand the transmission mechanisms, or to estimate the models.

4.2 Implementations

Once a gmod file is uploaded by an user to the toolbox's website, the toolbox reads the file and returns MATLAB files that can be run locally in the user' computer to solve and simulate their model.

General Implementations The gmod file is first parsed into an internal model structure, based on which the toolbox generates the C++ and MATLAB source codes. The toolbox then compiles the C++ source code to a dynamic library that MATLAB can call. All the real computations are implemented in the native C++ code to achieve maximum performance and contained in the dynamic library, while the MATLAB file provides a convenient interface to print, debug, and specify options. To reach maximum computation efficiency, our implementation takes care of miscellaneous designs covering equation solver, interpolation, automatic differentiation, and parallel computation, which we discuss below each of them in details.[6]

Equation Solver The time iteration step requires solving systems of equations for each discretized point in the state space. Since evaluating the function to be solved is rather costly, it is crucial that we design an efficient equation solver. We implement the Powell's dogleg algorithm augmented with an interior-point method to respect the box constraints (Powell, 1970; Coleman and Li, 1996; Bellavia et al., 2012). We also provide interfaces to commercial optimization software SNOPT and Knitro for users with licenses.[7]

Automatic Differentiation Since we use a gradient-based equation solver and the function evaluation is expensive, it is crucial to calculate the gradients efficiently. We use a reverse-mode automatic differentiation method implemented by Adept (Hogan, 2014). This library utilizes the expression template feature of C++, so much of the differentiation is taken care of at compile time, bringing the computation cost on par with evaluating analytical gradients.

Interpolation The time iteration step (2) involves function approximations because $(z', s'(z'))$ might fall outside $\mathcal{C}^{(n)}$. The default option is multi-dimensional linear interpolation or splines. We also implement a multi-dimensional adaptive sparse grid method with hierarchical hat basis functions developed in Ma and Zabaras (2009) and recently applied in economic applications by Brumm and Scheidegger (2017). We provide analytical gradients to these approximation procedures, which complement the automatic differentiation method to achieve maximum performance.

Parallel Computation Within a time iteration, the problems are independent of each other while they share a large chunk of data for function approximations. To utilize this structure, we use multi-threaded parallel computation so all problems share a same block of memory for function approximation parameters, minimizing the overhead for data communications; when evaluating the interpolations with splines or the adaptive sparse grid method, we design the data structure such that it can exploit the single-instruction-multiple-data (SIMD) CPU instructions. This design of parallelism turns out to be efficient—the program executes fast on a single processor and scales well with the number of CPU cores.

[6]For each of the implementation details, we also provide a separate library when possible so that they can be used independently of the toolbox.

[7]Our own implementation of the algorithm turns out to be more efficient both in terms of number of function calls and overhead, for a large class of test problems. This is partly because the algorithm we implement is designed for solving equations, while these commercial softwares target a more general class of optimization problems. Besides, the equation solver we implement targets small to medium scale problems (less than 1000 unknowns), which are adequate for most applications in economics while these commercial softwares accommodate much larger problems and thus incurs more overhead.

5 Applications

In this section, we provide examples of how well-known models can be solved using our toolbox. The gmod files for these models are provided in the appendix. The toolbox algorithm is different from the algorithm provided in the original papers. These examples could be read independently and the notation follows closely from the notation in the original papers. We also refer readers to the original papers for the important economic motivation of these models.

5.1 Asset Pricing with Heterogeneous IES by Guvenen (2009)

Guvenen (2009) constructs a two-agent model to explain several salient features of asset pricing moments, such as high risk premium, low and relatively smooth interest rate, and countercyclical movements in risk premium and Sharpe ratio. Two key ingredients of his model are limited stock market participation and heterogeneity in the elasticity of intertemporal substitution in consumption (EIS).

The solution algorithm in Guvenen (2009) is quite different from ours. His is based on the algorithm in Krusell and Smith (1998): starting from a conjectured law of motion for state-variables and pricing functions, he solves the agents' Bellman equation and the agents' policy functions using standard value function iterations. Then he uses these policy functions and temporary market clearing conditions to obtain a new law of motions and new pricing functions. These functions are then used as conjectured functions to obtain new functions. He keeps iterating until the new functions are close enough to the conjectured functions.

Our algorithm recognizes that, because the agents' optimization problems are concave problems, the first-order conditions are sufficient for optimality (without having to solve the agents' Bellman equation). Therefore, we can directly use policy function iterations to solve jointly for agents' optimization problems and market clearing conditions.

5.1.1 Model Description

There are two types of infinitely-lived agents: stock*h*olders (h) with measure μ, and *n*on-stockholders (n) with measure $1-\mu$. Agents have Epstein-Zin utility functions

$$U_{i,t} = \left\{ (1-\beta)\, c_{i,t}^{1-\rho^i} + \beta \left[\mathbb{E}_t \left(U_{i,t+1}^{1-\alpha} \right) \right]^{\frac{1-\rho^i}{1-\alpha}} \right\}^{1/\left(1-\rho^i\right)}. \tag{5}$$

for $i = h, n$. Most importantly, $\rho^h < \rho^n$, i.e., the non-stockholders have lower EIS which is inversely proportional to ρ^i, and thus they have higher desire for consumption smoothness. Each agent has one unit of labor endowment.

Stockholders can trade stock s_t and bond $b_{h,t}$ at prices P_t^s and P_t^f respectively. Their budget constraint is

$$c_{h,t} + P_t^f b_{h,t+1} + P_t^s s_{t+1} \leq b_{h,t} + s_t \left(P_t^s + D_t\right) + W_t,$$

where W_t is the labor income and borrowing constraint is

$$b_{h,t+1} \geq -\underline{B},$$

and in calibration $\underline{B}$ is set to six times of the average monthly wage rate. The non-stockholders have the same constraints. In addition, they are restricted from trading stocks.

A representative firm produces the consumption good using capital K_t and labor L_t based on a Cobb-Douglas production function:

$$Y_t = Z_t K_t^{\theta} L_t^{1-\theta},$$

and the technology evolves according to an AR(1) process:

$$\ln Z_{t+1} = \phi \ln Z_t + \varepsilon_{t+1},\ \varepsilon \overset{i.i.d.}{\sim} N\left(0, \sigma_\varepsilon^2\right).$$

The firm maximizes its value P_t^s expressed as the sum of its future dividends $\left\{D_{t+j}\right\}_{j=1}^{\infty}$ discounted by the shareholders' marginal rate of substitution process:

$$P_t^s = \max_{\left\{I_{t+j}, L_{t+j}\right\}} \mathbb{E}_t \left[\sum_{j=1}^{\infty} \beta^j \frac{\Lambda_{h,t+j}}{\Lambda_{h,t}} D_{t+j}\right]. \tag{6}$$

The firm accumulates capital subject to a concave adjustment cost function in investment:

$$K_{t+1} = (1-\delta) K_t + \Phi\left(\frac{I_t}{K_t}\right) K_t. \tag{7}$$

Each period, the firm sells one-period bonds at price P_t^f. The bond supply is constant and equals to χ fraction of its average capital stock $\bar{K}$. Thus dividend D_t can be written as

$$D_t = Z_t K_t^{\theta} L_t^{1-\theta} - W_t L_t - I_t - \left(1 - P_t^f\right) \chi \bar{K}.$$

A sequential competitive equilibrium is given by sequences of allocations

$$\{c_{i,t}, b_{i,t+1}, s_{t+1}, I_t, K_{t+1}, L_t\}$$

$i = h, n$ and prices $\left\{P_t^s, P_t^f, W_t\right\}$ such that (i) given the price sequences, $\{c_{i,t}, b_{i,t+1}, s_{t+1}\}$ $i = h, n$ solve the stockholders' and non-stockholders' optimization problems; (ii) Given the wage sequence $\{W_t\}$ and the law of motion for capital (7), $\{L_t, I_t\}$ are optimal for the representative firm; (iii) all markets clear:

$$\mu b_{h,t+1} + (1-\mu)\, b_{n,t+1} = \chi \bar{K}, \tag{8}$$

$$\mu s_{t+1} = 1, \tag{9}$$

$$L_t = 1,$$

$$\mu c_{h,t} + (1-\mu)\, c_{n,t} + I_t = Y_t.$$

5.1.2 Computation

We use $\{K_t, B_t^n, Z_t\}$ as the aggregate state variables, where $B_t^n = (1-\mu)\, b_{n,t}$ is total bond holding by the non-stockholders. The optimization problems of the households and the representative firm are concave maximization problems, so the first-order conditions are necessary and sufficient for optimality. With this observation and the aforementioned state variables, the competitive equilibrium in this model can be represented by a system of short-run equilibrium conditions (1) required by the general framework. This system consists of 8 unknowns: $\{c_{h,t}, c_{n,t}, I_t, B_{t+1}^n, \lambda_{h,t}, \lambda_{n,t}, P_t^s, P_t^f\}$, and 8 equations:

1. Euler equations for bond holding:

$$P_t^f = \beta\,(1 + \lambda_{i,t})\, \mathbb{E}_t \left(\frac{\Lambda_{i,t+1}}{\Lambda_{i,t}}\right), \ \forall i = h, n.$$

2. Euler equations for the stockholders' demand of equity:

$$P_t^s = \beta \mathbb{E}_t \left[\frac{\Lambda_{h,t+1}}{\Lambda_{h,t}} \left(P_{t+1}^s + D_{t+1}\right)\right].$$

3. Slackness condition of borrowing limit:

$$\lambda_{i,t}\left(b_{i,t+1}+\underline{B}\right)=0,\ \forall i=h,n.$$

4. The budget constraints (imposing $s_{t+1}=1/\mu$):

$$c_{h,t}+P_t^f b_{h,t+1}+\frac{P_t^s}{\mu}=P_t^s+D_t+\frac{\chi\bar{K}-B_t^n}{\mu}+W_t,$$

$$c_{n,t}+P_t^f b_{n,t+1}=\frac{B_t^n}{1-\mu}+W_t.$$

5. Firm's optimal capital accumulation K_{t+1}:

$$q_t=\beta\mathbb{E}_t\left\{\frac{\Lambda_{h,t+1}}{\Lambda_{h,t}}\left[\theta Z_t K_t^{\theta-1}-\frac{I_{t+1}}{K_{t+1}}+q_{t+1}\left(1-\delta+\Phi\left(\frac{I_{t+1}}{K_{t+1}}\right)\right)\right]\right\},\qquad(10)$$

in which capital price q_t is the Lagrangian multiplier on the capital formation (7) and satisfies

$$q_t\Phi'\left(\frac{I_t}{K_t}\right)=1.\qquad(11)$$

The auxiliary variables can be determined by the utility function (5), market clearing conditions, (7) and the following two equations:

$$W_t=(1-\theta)\,Z_t\left(\frac{K_t}{L_t}\right)^{\theta},$$

$$\beta\frac{\Lambda_{i,t+1}}{\Lambda_{i,t}}=\beta^{\frac{1-\alpha}{1-\rho^i}}\left(\frac{c_{i,t+1}}{c_{i,t}}\right)^{-\rho^i}\left[\frac{\frac{U_{i,t+1}}{c_{i,t}}}{\left[\left(\frac{U_{i,t}}{c_{i,t}}\right)^{1-\rho^i}-(1-\beta)\right]^{1/\left(1-\rho^i\right)}}\right]^{\rho^i-\alpha}.$$

Having represented the equilibrium in the required form (1), we can then use the toolbox to solve for a recursive equilibrium. In period t, the 6 future variables in use: $c_{h,t+1}$, $c_{n,t+1}$, $P_{t+1}^s+D_{t+1}$, I_{t+1}/K_{t+1}, $U_{h,t+1}$ and $U_{n,t+1}$ are functions of $\left\{K_{t+1},B_{t+1}^n,Z_{t+1}\right\}$ and are solved from the previous iteration. Similar to Guvenen (2009), the initial guess for these functions are obtained by solving a version of the model with no leverage

($\chi = 0$, $\underline{B} = 0$).[8]

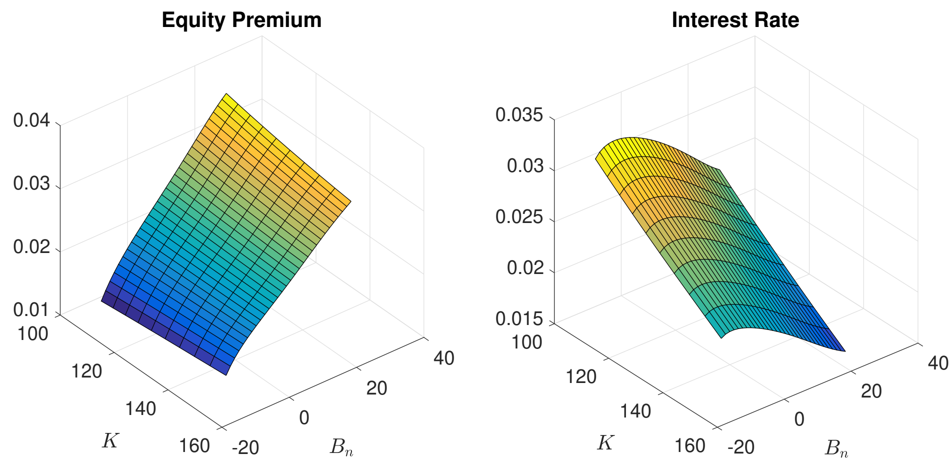


Figure 3: Asset Pricing Policy Functions in Guvenen (2009)
Note: The figure plots the annual equity premium and interest rate as functions of $\{K, B^n\}$. We use the same parameter values as in Table 1 of Guvenen (2009), and set $Z_t = 1$.

In Figure 3, we plot the annual equity premium and interest rate as functions of $\{K, B^n\}$ by fixing $Z_t = 1$. Figure 4 plots the ergodic distributions of capital and the financial wealth share of stockholders.

5.1.3 Mapping into the General Setup

For the model in Guvenen (2009) described above, the correspondence with our general setup of the toolbox is

$$z = (Z),$$

and

$$s = (K, B^n),$$

and

$$x = (c_h, c_n, I, B^{n\prime}, \lambda_h, \lambda_n, P^s, P^f, q, U_h, U_n).$$

[8]It is easy to implement this algorithm in the toolbox. Users can solve the no-leverage version first, and after convergence, use its policy functions as the initial conjecture for the benchmark case. The toolbox allows the users to provide their own initial conjectured functions by the "WarmUp" option, so they do not need to write separate codes for different cases. See the code available online for details. Furthermore, the functions provided can be defined on different grid points from the state variables, which offers the users much flexibility. For example, a user can solve a model with coarse grids for speed first and then uses its converged policy functions as the initial conjecture for the same model with finer grids.

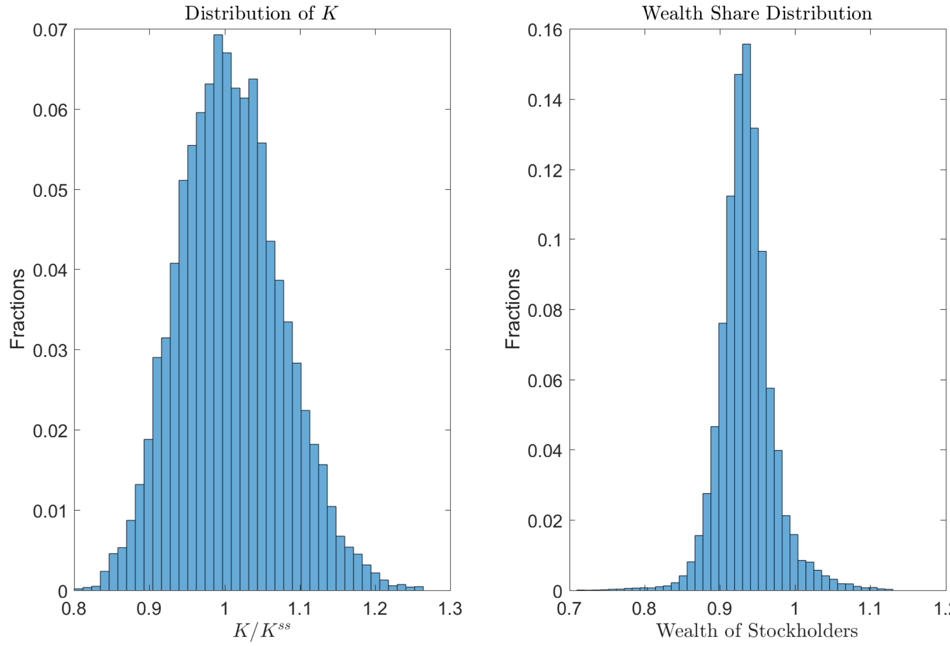


Figure 4: Ergodic Distributions of Capital and Wealth Share
Note: The Ergodic Distributions are generated by simulation. We use the same parameter values as in Table 1 of Guvenen (2009).

5.2 Sudden Stops in an Open Economy by Bianchi (2011)

Bianchi (2011) studies an incomplete-markets open economy model that can generate competitive equilibria featuring sudden stop episodes, mimicking those experienced by many emerging economies. A sudden stop episode features a large output drop and current account reversals, which are at odds with the prediction of a standard incomplete-markets model with precautionary saving motives. A key feature for the model in Bianchi (2011) is to introduce feedback of the price of non-tradable goods to the borrowing constraint: a negative external shock that lowers the equilibrium price of non-tradable goods tightens the borrowing constraint and forces reducing the consumption of tradable goods, which further lowers the price of non-tradable goods. The competitive equilibrium is inefficient since agents do not take into account the effects of non-tradable price on the borrowing constraint in the event of a sudden stop crisis. This leads to ex-ante over-borrowing and calls for policy interventions.

The borrowing constraint is occasionally binding in the equilibrium's ergodic set, and the equilibrium policy and state transition functions are highly non-linear when the borrowing constraint binds. Therefore, a global and non-linear solution is essential to capture the model's rich dynamics. We now describe how this class of models[9] can be solved by the toolbox robustly and efficiently, using the exact model in Bianchi (2011) as an example.

To compute the competitive equilibrium, Bianchi (2011) uses a policy function iteration algorithm. His algorithm treats cases with binding or non-binding constraint separately, while the toolbox uses the Lagrange multiplier on the constraint and the complementary slackness condition to write these cases with the same system of equations. This seemingly minor detail is important in allowing the model to be written and solved in the same framework as in other models.

5.2.1 Model Description

Small-open economy representative consumers derive utility from consumption of tradable goods c_t^T and of non-tradable goods c_t^N according to

$$\mathbb{E}\Big[\sum_{t=0}^{\infty}\beta^t\frac{c_t^{1-\sigma}}{1-\sigma}\Big] \tag{12}$$

with the composite consumption

$$c_t = A\left(c_t^T, c_t^N\right) \equiv [\omega(c_t^T)^{-\eta} + (1-\omega)(c_t^N)^{-\eta}]^{-\frac{1}{\eta}}, \tag{13}$$

where $\omega \in (0,1)$ and $\eta > -1$ are parameters. $\beta \in (0,1)$ is the discount factor and σ is the coefficient of relative risk-aversion. $\mathbb{E}$ is the expectation operator to integrate shocks below.

Borrowing is via a state non-contingent bond in tradable goods at a constant world interest r. The endowments of tradable goods y_t^T and non-tradable goods y_t^N follow exogenous stochastic processes. The consumer faces the following sequential budget constraint

$$b_{t+1} + c_t^T + p_t^N c_t^N = b_t(1+r) + y_t^T + p_t^N y_t,$$

where b_{t+1} is the bond-holding determined at period t. Tradable good is the numeraire and p_t^N is the equilibrium price of non-tradable goods, taken as given by consumers.

A key feature of the model is that the borrowing is subject to a borrowing constraint

[9]Other models in this literature that can be solved by the toolbox include Mendoza (2010) with endogenous capital accumulation and a borrowing constraint tied to asset instead of commodity price, which we include as an example in the toolbox's website.

tied to the non-tradable good price as below

$$b_{t+1} \geq -(\kappa^N p_t^N y_t^N + \kappa^T y_t^T)$$

which says that the borrowing cannot exceed the sum of κ^N fraction of the value of non-tradable goods, plus κ^T fraction of the value of tradable goods, with parameter $\kappa^N > 0$, $\kappa^T > 0$ determining the collaterability of the non-tradable and tradable endowments, respectively.

Equilibrium Definition. A sequential competitive equilibrium corresponds to stochastic processes $\{b_{t+1}, c_t^T, c_t^N, c_t, p_t^N\}_{t=0}^{\infty}$ such that $\{b_{t+1}, c_t^T, c_t^N\}$ solves the households optimization problem and markets clear:

$$\begin{aligned} c_t^N &= y_t^N \\ c_t^T &= y_t^T + b_t(1+r) - b_{t+1} \end{aligned}$$

Because the households' maximization problem is a concave problem, the first-order conditions are necessary and sufficient for optimality: there exists stochastic processes for the Lagrange multiplier, $\{\mu_t, \lambda_t\}$ such that, together with $\{b_{t+1}, c_t^T, c_t^N\}$ the following conditions are satisfied:

$$p_t^N = \left(\frac{1-\omega}{\omega}\right)\left(\frac{c_t^T}{c_t^N}\right)^{\eta+1} \tag{14}$$

$$\lambda_t = \beta(1+r)\mathbb{E}_t\lambda_{t+1} + \mu_t \tag{15}$$

$$\mu_t\left[b_{t+1} + (\kappa^N p_t^N y_t^N + \kappa^T y_t^T)\right] = 0 \tag{16}$$

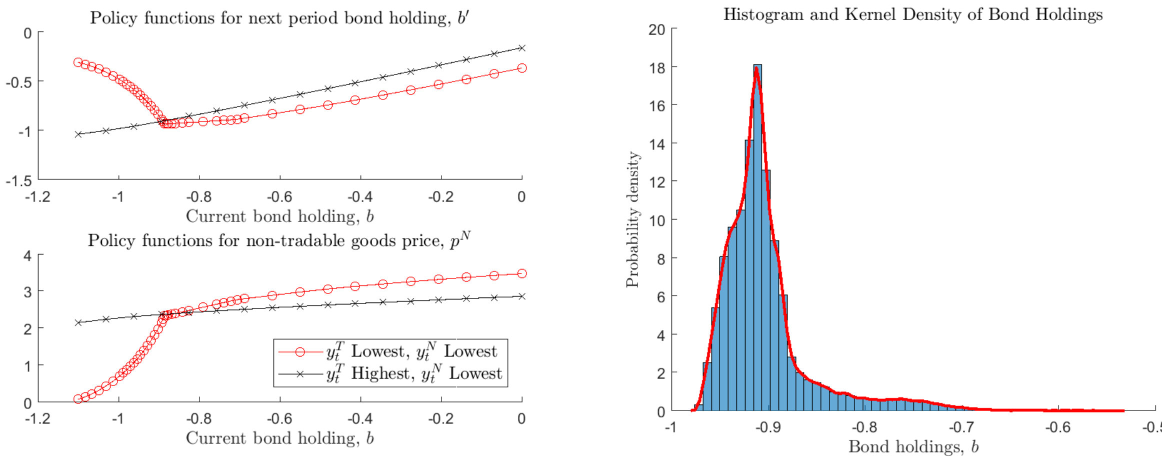
$$b_{t+1} + c_t^T + p_t^N c_t^N = b_t(1+r) + y_t^T + p_t^N y_t^N$$

where

$$\lambda_t = c_t^{-\sigma}\frac{\partial A(c_t^T, c_t^N)}{\partial c_t^T} = c_t^{-\sigma}[\omega(c_t^T)^{-\eta} + (1-\omega)(c_t^N)^{-\eta}]^{-\frac{1}{\eta}-1}\omega[c_t^T]^{-\eta-1}.$$

With these observations, the equilibrium in this economy can be represented in the form (1) required to apply the toolbox.

Parameterization. We use the exact parameters as in the benchmark calibration in Bianchi (2011).



(a) Policy Functions

(b) The ergodic Distribution of bond holdings

Figure 5: Ergodic Distribution and Policy Functions of Bianchi (2011)

Note: The policy functions are for exogenous states fixing y_t^N to be the lowest of the 4 realizations, and y_t^T to be the highest or lowest of the 4 realizations respectively. The markers indicate the grid points automatically generated by the adaptive-grid method. The histogram is based on 100 sample paths of 1000-period simulations, burning the first 500 periods of each path.

5.2.2 Computation

The equilibrium can be input into the toolbox by discretizing the exogenous endowments process y_t^N and y_t^T. Following the parameterization and discretization used by Bianchi (2011), we discretize the joint process of (y_t^N, y_t^T) to 16 states. The natural endogenous state variable of the economy is b_t.

Like previous examples, a time step of policy iterations is to solve the equilibrium system defined above, for each collocation point of exogenous and endogenous states, taking the state transition function implicitly defined in $\lambda_{t+1}(y_{t+1}^N, y_{t+1}^T, b_{t+1})$ as given. After each time step, $\lambda_t(y_t^N, y_t^T, b_t)$ is compared with $\lambda_{t+1}(y_{t+1}^N, y_{t+1}^T, b_{t+1})$ to check for convergence under certain criteria.

While it is possible to specify an exogenous discrete grid for b_t, since the model is highly non-linear, we illustrate the use of function approximations with adaptive-grid methods with the toolbox, which automatically place more points to the state space that features high non-linearity.[10] The equilibrium policy functions for p_t^N and b_{t+1}, and the ergodic distribution of b_t are presented in Figure 5.

[10] As described in the user manual in the appendix, we take care of implementation details and the user only needs to specify one option in the toolbox to switch to the adaptive grid method. The adaptive grid method is based on Ma and Zabaras (2009) and Brumm and Scheidegger (2017), and features sparsity for multi-dimensional problems and thus can accommodate models with high-dimension state space.

As shown in the left panel, the policy functions are highly nonlinear: when the borrowing constraint binds, the price of non-tradable goods declines sharply in the level of exist borrowing; future borrowing declines, instead of increasing, as the economy goes further in debt, implying current account reversals. If the borrowing constraint does not bind, then the price movement is much milder as we vary the level of existing debt, and current account reversals do not happen. The right panel displays the ergodic distribution of bond holdings, which show that the non-linear regions do exist in the ergodic set of the equilibrium and thus cannot be ignored, but due to precautionary motives, the frequency of the economy being in these regions cannot be determined ex-ante, highlighting the necessity of using a global solution method.

The markers on the policy functions indicate the grid points automatically placed by the adaptive-grid method, and show that the method adds more points to the state space where the policy and state transition functions become non-linear. Importantly, the method takes care that these non-linear regions can differ across exogenous states, as shown in the figure. This illustrates the effectiveness of the adaptive-grid method for this class of models, as these non-linear regions of state-space cannot be determined ex-ante, and require very dense exogenous grids or painful manual configurations.

5.2.3 Mapping into the General Setup

For the model in Bianchi (2011) described above, the correspondence with our general setup of the toolbox is

$$z = (y^T, y^N),$$

and

$$s = (b),$$

and

$$x = (b', c^T, c^N, c, \mu, \lambda, p^N).$$

5.3 Safe Assets by Barro et al (2017)

Barro et al. (2017) incorporate heterogeneous risk-aversion into the model with rare disasters in Barro (2006) to study the endogenous creation of safe-asset. Their model features incomplete markets: agents can only trade in a stock and a bond as in Heaton and Lucas (1996). They solve their model using a mixture of projection and perturbation method developed in Fernández-Villaverde and Levintal (2018). Our toolbox's algorithm is a purely a projection method. It uses wealth share as state variables and the normalization from Cao (2018) to deal with consumption being close to zero when some of the wealth share is close to zero. As Barro et al. (2017) discuss in their paper, their solution method is not sufficiently accurate for large values of risk-aversion coefficients.[11] We show below that our method can tackle these cases effectively and uncover new economic insights in these cases.

5.3.1 Model and Normalization

There are two groups of agents, $i = 1, 2$ in the economy. Agents have an Epstein and Zin (1989)-Weil (1990) utility function. The coefficients of risk aversion satisfy $\gamma_2 \geq \gamma_1 > 0$, i.e., agent 1 is less risk-averse than agent 2. The other parameters between these two groups are the same. There is a replacement rate v at which each type of agents move to a state that has a chance of μ_i of switching into type i. Taking the potential type shifting into consideration, their utility function can be written as

$$U_{i,t} = \left\{ \frac{\rho + v}{1+\rho} C_{i,t}^{1-\theta} + \frac{1-v}{1+\rho} \left[\mathbb{E}_t \left(U_{i,t+1}^{1-\gamma_i} \right) \right]^{\frac{1-\theta}{1-\gamma_i}} \right\}^{1/(1-\theta)}. \tag{17}$$

In this economy, there is a Lucas tree generating consumption good Y_t in period t consumed by both agents. Y_t is subject to identically and independently distributed rare-disaster shocks. With probability $1-p$, Y_t grows by the factor $1+g$; with a small probability p, Y_t grows by the factor $(1+g)(1-b)$. Thus the expected growth rate of Y_t in each period is $g^* \approx g - pb$. Denote agent i's holding of the tree as K_{it}. The supply of the Lucas tree is normalized to one, and denote its price as P_t. The gross return of holding equity is $R_t^e = \frac{Y_t + P_t}{P_{t-1}}$. Agents also trade a risk-free bond, B_{it}, whose net supply is zero, and the gross interest rate is R_t^f.

Denote the beginning-of-period wealth of agent i by A_{it}. Each agent's budget constraint is

$$C_{it} + P_t K_{it} + B_{it} = A_{it}.$$

Considering the type shifting shock, the law of motion of A_{it} is

$$A_{it} = (Y_t + P_t) \left[K_{it-1} - v \left(K_{it-1} - \mu_i \right) \right] + (1-v) R_t^f B_{it-1}.$$

As in Cao (2018, Appendix C.3, Extension 3), we normalize the utility U_{it} and consumption C_{it} by A_{it} and write equation (17) as follows:

[11] See Table 2 in their paper.

$$u_{it}^{1-\theta} = \frac{\rho + v}{1+\rho} c_{i,t}^{1-\theta} + \frac{1-v}{1+\rho} \left(1 - c_{it}\right)^{1-\theta} \left(\mathbb{E}_t \left[\left(R_{i,t+1} u_{it+1}\right)^{1-\gamma_i} \right]\right)^{\frac{1-\theta}{1-\gamma_i}}, \tag{18}$$

in which $u_{it} = U_{it}/A_{it}$, $c_{it} = C_{it}/A_{it}$, and

$$R_{i,t+1} = x_{it} R_{t+1}^{e} + \left(1 - x_{it}\right) R_{t+1}^{f}$$

is the average return of agent i's portfolio, and

$$x_{it} = \frac{P_t K_{it}}{P_t K_{it} + B_{it}}$$

is the equity share of agent i's portfolio holding. The FOCs for consumption and portfolio choices are

$$\left(\rho + v\right) c_{i,t}^{-\theta} = \left(1 - v\right) \left(1 - c_{it}\right)^{-\theta} \left[\mathbb{E}_t \left(R_{i,t+1} u_{it+1}\right)^{1-\gamma_i}\right]^{\frac{1-\theta}{1-\gamma_i}}, \tag{19}$$

and

$$\mathbb{E}_t \left[\frac{\left(R_{t+1}^{e} - R_{t+1}^{f}\right) u_{it+1}}{\left(R_{i,t+1} u_{it+1}\right)^{\gamma_i}} \right] = 0. \tag{20}$$

The choice of c_{it} and x_{it} are identical across agents of the same type i, and the portfolio choices of agent i is

$$K_{it} = x_{it} \left(1 - c_{it}\right) \left(1 + p_t\right) / p_t \omega_{it},$$
$$b_{it} = \left(1 - x_{it}\right) \left(1 - c_{it}\right) \left(1 + p_t\right) \omega_{it}.$$

In equilibrium, prices are determined such that markets clear:

$$C_{1t} + C_{2t} = Y_t, \tag{21}$$
$$K_{1t} + K_{2t} = 1, \tag{22}$$
$$B_{1t} + B_{2t} = 0. \tag{23}$$

To achieve stationarity, we normalize $\{B_{it}, P_t\}$variables by Y_t. We define the wealth share of agent i as

$$\omega_{it} = K_{it-1} - v\left(K_{it-1} - \mu_i\right) + \frac{\left(1 - v\right) R_t^{f} b_{it-1}}{\left(1 + p_t\right)\left(1 + g_t\right)}. \tag{24}$$

We see that given the market clearing conditions (22) and (23),

$$\omega_{1t} + \omega_{2t} = 1, \forall t.$$

5.3.2 Log Utility

For much of the analysis in Barro et al. (2017), the intertemporal elasticity of substitution θ is set at 1. In this case, agents consume a constant share of their wealth, and equation (19) is replaced by

$$c_{it} = \frac{\rho + v}{1 + \rho}.$$

Using this relationship for $i = 1, 2$, and use the market clearing conditions (21), (22) and (23), we have

$$p_t = \frac{1 - v}{\rho + v}.$$

The utility function (18) is replaced by

$$\begin{aligned} \ln u_{it} =& \frac{\rho + v}{1 + \rho} \ln c_{it} + \frac{1 - v}{1 + \rho} \ln (1 - c_{it}) \\ & + \frac{1 - v}{1 + \rho} \frac{1}{1 - \gamma_i} \ln \left[\mathbb{E}_t \left(R_{i,t+1} u_{it+1} \right)^{1 - \gamma_i} \right]. \end{aligned} \tag{25}$$

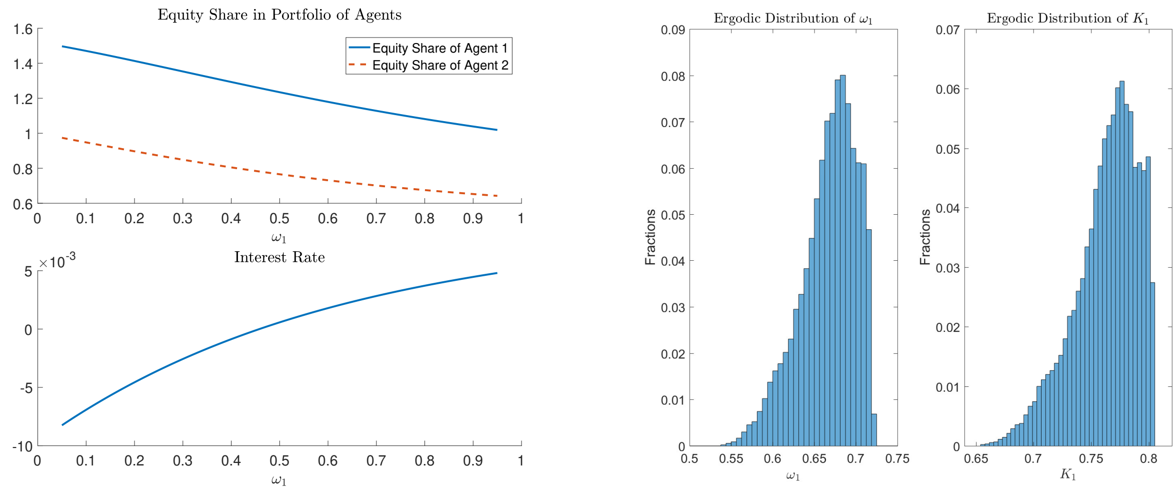
The state variable is ω_{1t}. The unknowns are $\left\{ x_{1t}, x_{2t}, R_t^f, \omega_{it+1} \left(z_{t+1} \right) \right\}$. We have 4 equations: (20) for $i = 1, 2$, the market clearing condition for bond (23) and the consistency equation (24) to solve the unknowns.

Since the growth shock is i.i.d., ω_1 is the only state variable. The policy functions and stationary distributions of $\omega_1$are given in Figure 6.

When the economy is at the steady state of normal times, the impulse responses after a one-time disaster shock in the first period are given in Figure 7.

In Table 2 of Barro et al. (2017), the values of risk aversion parameters γ_1 and γ_2 are adjusted to target an average annual interest rate $\bar{R}^f = 1.01$. The implicit reasoning is that, for each γ_1, $\bar{R}^f$ is decreasing in γ_2 and there exists a value of γ_2 such that $\bar{R}^f = 1$. In Table 2 of their paper displays γ_2 as a function of γ_1 following this procedure. However, when $\gamma_1 = 3.1$, the authors set $\gamma_2 = 10$ while acknowledging that their numerical solutions in this region were insufficiently accurate.

Using our toolbox, we can solve this problem for a wider range of γ_2. In Figure 8(a), we plot $\bar{R}^f$ corresponding to different values of γ_2 up to 100. In particular, we find that $\bar{R}^f$ is a non-monotone function of γ_2. In addition, $\bar{R}^f = 1.01$ cannot be reached when



(a) Policy Functions

(b) Ergodic Distributions

Figure 6: Ergodic Distribution and Policy Functions

Note: The figure is generated using the baseline parameters in Barro et al. (2017). For annual data, $\rho = 0.02$, $v = 0.02$, $\mu = 0.5$, $\gamma_1 = 3.3$, and $\gamma_2 = 5.6$. Growth rate in normal times is 0.025. Rare disaster happens with probability 4%, and once it happens, productivity drops by 32%. The model period is one quarter.

$\gamma_1 = 3.1$, since $\bar{R}^f$ is increasing in γ_2 when γ_2 is larger than 8.

The mechanism behind the non-monotonicity can be understood by looking at two opposing forces. First, as γ_2 gets larger, agent 2 becomes more risk-averse, and demand for more of the safe asset (bond). This pushes down $\bar{R}^f$. Second, an increase in γ_2 also leads agent 1 to borrow more and become more leveraged. Since the return of equity is higher than bond, the average wealth share of agent 1, ω_1 becomes larger. Larger ω_1 leads to more relative supply of safe asset and pushes up $\bar{R}^f$. Whether $\bar{R}^f$ decreases or increases in γ_2 depends on which force dominates. Figure 8 shows that when γ_2 is below 8 the first force dominates and $\bar{R}^f$ is decreasing in γ_2 as assumed in Barro et al. (2017). However, when γ_2 is larger than 8, the second force dominates and $\bar{R}^f$ is increasing in γ_2. When γ_2 is larger than 20, $\bar{R}^f$is not responsive to γ_2, since the wealth distribution ω_1 is almost degenerated to its upper limit. See Figure 8(b) as a comparison of two cases: $\gamma_2 = 8$ versus $\gamma_2 = 10$.

5.3.3 Mapping into the General Setup

For the model in Barro et al. (2017) described above, the correspondence with our general setup of the toolbox is

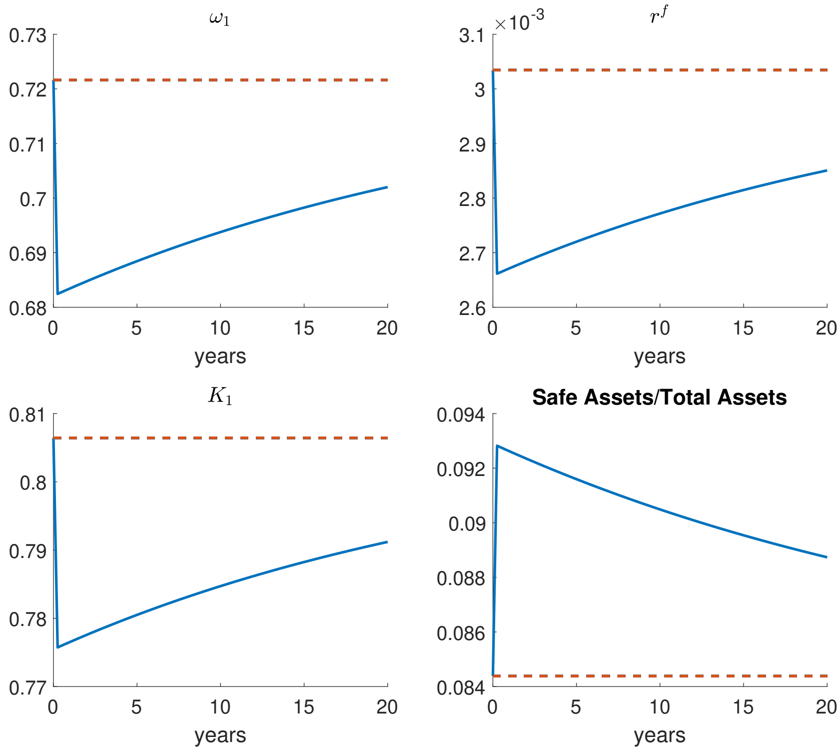


Figure 7: Dynamic Paths Following a Disaster

Note: The figure plots the dynamic paths after a one-time disaster using the baseline parameters in Barro et al. (2017). For annual data, $\rho = 0.02$, $v = 0.02$, $\mu = 0.5$, $\gamma_1 = 3.3$, and $\gamma_2 = 5.6$. Growth rate in normal times is 0.025. Rare disaster happens with probability 4%, and once it happens, productivity drops by 32%. The model period is one quarter.

$$z = (g),$$

and

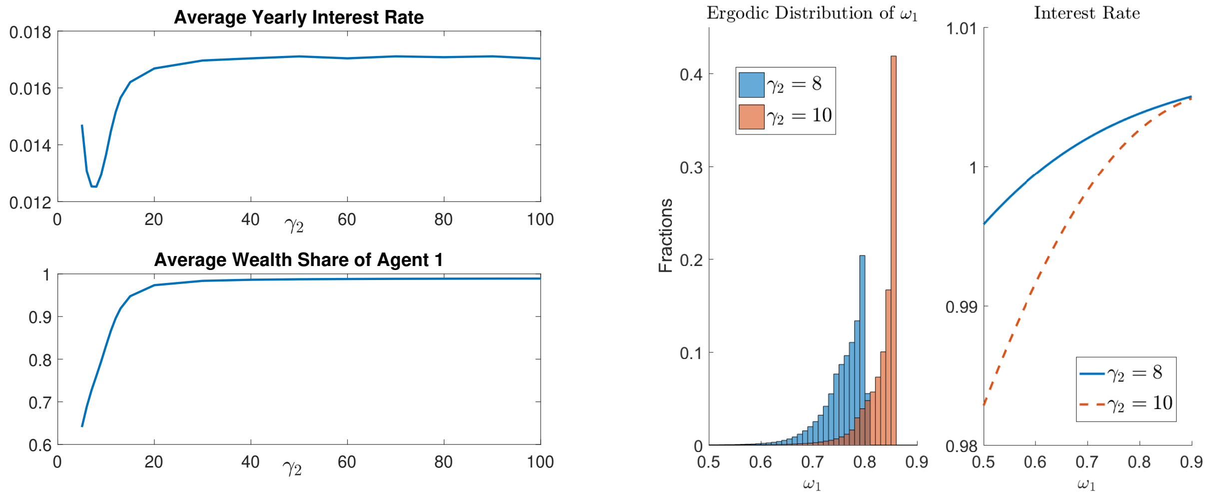
$$s = (\omega_1),$$

and

$$x = \left(c_1, c_2, x_1, x_2, R^f, K_1, b_1, p\right).$$

6 Heterogeneous Agent Models with Aggregate Shocks

The framework is more readily applicable to solving GDSGE models with a finite number of agents, or more precisely a finite number of agent-types. This is because in these models the equilibrium conditions can be represented as a system of a finite num-



(a) Interest Rate and Wealth Share of Agent 1

(b) Comparing Cases with Different γ_2

Figure 8: Interest Rate with Different γ_2

Note: The figure is generated using the baseline parameters in Barro et al. (2017). In particular, we fix $\gamma_1 = 3.1$ and change the value of γ_2 to generate the results. In Figure (a), we plot the average interest rate and wealth share of agent 1 correponding to different values of γ_2. In Figure (b), we compare the policy functions of R^f and ergodic distributions when $\gamma_2 = 8$ and 10.

ber of equations and unknowns. The solutions to these systems lie in finite-dimensional spaces. The policy and transition functions are mappings from finite-dimensional state-spaces to these finite-dimensional spaces. While in fully heterogeneous agent models à la Krusell and Smith (1998) with both idiosyncratic and aggregate shocks, both state spaces, such as spaces of wealth distributions, and equilibrium objects, such as policy and value functions, are infinite-dimensional objects, as emphasized in Cao (2020).

However, Cao (2020) shows that incomplete markets models with finite agent types are useful special cases of fully heterogeneous agent, incomplete markets models with both idiosyncratic and aggregate shocks a la Krusell and Smith (1998). In particular, the former corresponds to the latter in which idiosyncratic shocks are perfectly persistent. We provide an explicit comparison between the two models in the toolbox's website. The dynamics of the aggregate variables in the two models are similar. Therefore, in general, the solution of the finite-agent models can be useful in understanding the properties of the fully heterogenous agent models and can be solved at low cost using the toolbox.

In addition, the toolbox can be used to solve the agents' decision problem and to simulate in the latter given conjectured laws of motion of the aggregate variables. Then, with an additional fixed-point iteration on these laws of motion, which can be coded up simply in MATLAB, the toolbox solution can be used to solve for the DSGE in the latter.

In the last section of the paper, we show how this idea can be used to solve Krusell and Smith's baseline model in less 100 lines of toolbox code and 100 lines of MATLAB code. We also provide these codes, as well as the codes for heterogenous discount factors, in the toolbox's website.

7 Conclusion

We provide a unified framework and a toolbox for solving DSGE models using global methods. The toolbox proves to work efficiently and robustly for a large class of highly nonlinear models, covering macro-finance, international finance, and asset pricing models.

In principle, any dynamic problems characterized by systems of equations and state transition functions can readily fit in the toolbox, such as the decision rules in heterogeneous agent models (Huggett, 1993; Aiyagari, 1994; Krusell and Smith, 1998). The equilibrium systems of many models with discrete choices such as sovereign default can be transformed to equation systems by introducing preference or technology shocks (Chatterjee and Eyigungor, 2015; Arellano et al., 2020), and thus also fits in the toolbox. The toolbox uses a policy iteration method and thus can be used to solve stochastic transition paths such as Storesletten et al. (2019).

The toolbox also allows researchers to define model estimations in a unified way, which we leave for future development.

References

Aiyagari, S. R. (1994). Uninsured Idiosyncratic Risk and Aggregate Saving. *The Quarterly Journal of Economics 109*(3), 659–684.

Arellano, C., Y. Bai, and G. P. Mihalache (2020, January). Monetary Policy and Sovereign Risk in Emerging Economies (NK-Default). Working Paper 26671, National Bureau of Economic Research.

Barro, R. J. (2006, 08). Rare Disasters and Asset Markets in the Twentieth Century*. *The Quarterly Journal of Economics 121*(3), 823–866.

Barro, R. J., J. Fernández-Villaverde, O. Levintal, and A. Mollerus (2017). Safe assets. Technical report, National Bureau of Economic Research.

Bellavia, S., M. Macconi, and S. Pieraccini (2012). Constrained dogleg methods for nonlinear systems with simple bounds. *Computational Optimization and Applications 53*(3), 771–794.

Bernard, D. and A. L. Lyasoff (2012). Incomplete-market equilibria solved recursively on an event tree. *The Journal of Finance 67*(5), 1897–1941.

Bianchi, J. (2011). Overborrowing and Systemic Externalities in the Business Cycle. *The American Economic Review 101*(7), 3400–3426.

Blume, L. and D. Easley (2006). If you're so smart, why aren't you rich? belief selection in complete and incomplete markets. *Econometrica 74*(4), 929–966.

Brumm, J. and S. Scheidegger (2017). Using adaptive sparse grids to solve high-dimensional dynamic models. *Econometrica 85*(5), 1575–1612.

Brunnermeier, M. and Y. Sannikov (2014). A macroeconomic model with a financial sector. *American Economic Reviews 104*(2), 379–421.

Caballero, R. J., E. Farhi, and P.-O. Gourinchas (2008, March). An equilibrium model of "global imbalances" and low interest rates. *American Economic Review 98*(1), 358–93.

Cao, D. (2010). Collateral shortages, asset price and investment volatility with heterogeneous beliefs. Georgetown University Working Paper.

Cao, D. (2018). Speculation and financial wealth distribution under belief heterogeneity. *The Economic Journal 218*, 2258–81.

Cao, D. (2020). Recursive equilibrium in Krusell and Smith (1998). *Journal of Economic Theory 186*.

Cao, D., M. Evans, and W. Luo (2020). Exchange rate dynamics beyond business cycles. Technical report. SSRN 3552189.

Cao, D., W. Luo, and G. Nie (2019). Fisherian asset price deflation and the zero lower bound. Technical report. SSRN 3531341.

Cao, D. and G. Nie (2017). Amplification and asymmetric effects without collateral constraints. *American Economic Journal: Macroeconomics*.

Chatterjee, S. and B. Eyigungor (2015, December). A Seniority Arrangement for Sovereign Debt. *American Economic Review 105*(12), 3740–3765.

Coeurdacier, N., H. Rey, and P. Winant (2019). Financial integration and growth in a risky world. *Journal of Monetary Economics*.

Coleman, T. F. and Y. Li (1996). An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on optimization 6*(2), 418–445.

Coleman, W. J. (1990). Solving the stochastic growth model by policy-function iteration. *Journal of Business & Economic Statistics 8*(1), 27–29.

Coleman, W. J. (1991). Equilibrium in a production economy with an income tax. *Econometrica 59*(4), 1091–1104.

Duffie, D., J. Geanakoplos, A. Mas-Colell, and A. McLennan (1994). Stationary markov equilibria. *Econometrica 62*(4), 745–781.

Epstein, L. G. and S. E. Zin (1989). Substitution, risk aversion, and the temporal behavior of consumption and asset returns: a theoretical framework. *Econometrica 57*, 937–969.

Fernández-Villaverde, J. and O. Levintal (2018). Solution methods for models with rare disasters. *Quantitative Economics 9*(2), 903–944.

Geanakoplos, J. (2010). Leverage cycles. In K. R. Daron Acemoglu and M. Woodford (Eds.), *NBER Macroeconomics Annual 2009*, Volume 24, pp. 1–65. University of Chicago Press.

Gourio, F. (2012, May). Disaster risk and business cycles. *American Economic Review 102*(6), 2734–66.

Guerrieri, L. and M. Iacoviello (2017). Collateral constraints and macroeconomic asymmetries. *Journal of Monetary Economics 90*, 28 – 49.

Gust, C., E. Herbst, D. López-Salido, and M. E. Smith (2017, July). The Empirical Implications of the Interest-Rate Lower Bound. *American Economic Review 107*(7), 1971–2006.

Guvenen, F. (2009). A parsimonious macroeconomic model for asset pricing. *Econometrica 77*(6), 1711–1750.

He, Z. and A. Krishnamurthy (2011, 09). A Model of Capital and Crises. *The Review of Economic Studies 79*(2), 735–777.

Heaton, J. and D. Lucas (1996). Evaluating the effects of incomplete markets on risk sharing and asset pricing. *Journal of Political Economy 104*(3), 443–87.

Hogan, R. J. (2014). Fast reverse-mode automatic differentiation using expression templates in C++. *ACM Transactions on Mathematical Software (TOMS) 40*(4), 1–16.

Huggett, M. (1993, September). The risk-free rate in heterogeneous-agent incomplete-insurance economies. *Journal of Economic Dynamics and Control 17*(5–6), 953–969.

Judd, K. L., F. Kubler, and K. Schmedders (2000). Computing equilibria in infinite-horizon finance economies: The case of one asset. *Journal of Economic Dynamics and Control 24*(5), 1047 – 1078.

Kiyotaki, N. and J. Moore (1997). Credit cycles. *Journal of Political Economy 105*(2), 211–248.

Krusell, P. and J. Smith, Anthony A. (1998). Income and wealth heterogeneity in the macroeconomy. *The Journal of Political Economy 106*(5), 867–896.

Kubler, F. and K. Schmedders (2003). Stationary equilibria in asset-pricing models with incomplete markets and collateral. *Econometrica 71*(6), 1767–1793.

Ma, X. and N. Zabaras (2009). An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *Journal of Computational Physics 228*(8), 3084–3113.

Maggiori, M. (2017, October). Financial intermediation, international risk sharing, and reserve currencies. *American Economic Review 107*(10), 3038–71.

Magill, M. and M. Quinzii (1994). Infinite horizon incomplete markets. *Econometrica 62*(4), 853–880.

Mendoza, E. G. (2010, December). Sudden Stops, Financial Crises, and Leverage. *American Economic Review 100*(5), 1941–1966.

Powell, M. J. D. (1970). A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations. In *Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, ed.*

Sandroni, A. (2000). Do markets favor agents able to make accurate predicitions? *Econometrica 68*(6), 1303–1342.

Simsek, A. (2013). Belief disagreements and collateral constraints. *Econometrica 81*(1), 1–53.

Storesletten, K., B. Zhao, and F. Zilibotti (2019, August). Business Cycle during Structural Change: Arthur Lewis' Theory from a Neoclassical Perspective. SSRN Scholarly Paper ID 3447532, Social Science Research Network, Rochester, NY.

Weil, P. (1990). Nonexpected utility in macroeconomics. *The Quarterly Journal of Economics 105*(1), 29–42.

Winschel, V. and M. Kratzig (2010). Solving, estimating, and selecting nonlinear dynamic models without the curse of dimensionality. *Econometrica 78*(2), 803–821.

Appendix A Example Toolbox Codes

In this appendix, we provide the gmod files for the models discussed in Section 5. These codes can also be downloaded from the toolbox's website, together with the gmod codes for many other models.

A.1 Guvenen (2009)

```
% Parameters
parameters beta gamma thetah thetan alpha delta mu xsi chi a1 a2 Kss Bbar bn_shr_lb bn_shr_ub varianceScale;

beta = 0.9966;      % discount factor
gamma = 6;          % risk aversion
thetah = 1/.3;        % inv IES for stockholders
thetan = 1/.1;        % inv IES for non-stockholders
alpha = .3;         % capital share
delta = .0066;      % depreciation rate
mu = .2;             % participation rate
xsi = .4;           % adjustment cost coefficient
chi = .005;        % portion of bonds
a1 = (((delta^(1/xsi))*xsi)/(xsi-1));
a2 = (delta/(1-xsi));
Kss = ((1/beta-1+delta)/alpha)^(1/(alpha-1));
Bbar = -0.6*(1-alpha)*Kss^alpha; %borrowing constraint
varianceScale = 1e4;

TolEq = 1e-4;
INTERP_ORDER = 4; ExtrapOrder = 4;
PrintFreq = 100;
SaveFreq = inf;

% Shocks
var_shock Z;
shock_num = 15;
phi_z = 0.984;  % productivity AR(1)
mu_z = 0;
sigma_e = 0.015/(1+phi_z^2+phi_z^4).^0.5;
[z,shock_trans,~]=tauchen(shock_num,mu_z,phi_z,sigma_e,2);
Z = exp(z);

% States
var_state K bn_shr;
K_pts = 10;
K = exp(linspace(log(.84*Kss),log(1.2*Kss),K_pts));

bn_shr_lb = (1-mu)*Bbar/(chi*Kss);
bn_shr_ub = (chi*Kss - mu*Bbar)/(chi*Kss);
b_pts = 30;
bn_shr = linspace(bn_shr_lb,bn_shr_ub,b_pts);

% Last period
var_policy_init c_h c_n;

inbound_init c_h 1e-6 100;
inbound_init c_n 1e-6 100;

var_aux_init Y W vh vn vhpow vnpow Ps Pf Div Eulerstock Eulerbondh Eulerbondn Inv dIdK Eulerf;

model_init;
  Y = Z*(K^alpha);
  W = (1-alpha)*Z*(K^alpha);
  resid1 = 1 - (W + (bn_shr*chi*Kss/(1-mu)))/c_n; % c_n: individual consumption
  resid2 = 1 - (W + (Div/mu) + ((1-bn_shr)*chi*Kss/mu))/c_h; % c_h: individual consumption
  vh = ((1-beta)*(c_h^(1-thetah)))^(1/(1-thetah));
  vn = ((1-beta)*(c_n^(1-thetan)))^(1/(1-thetan));
  vhpow = vh^(1-gamma);
  vnpow = vn^(1-gamma);
```

```
  Pf = 0;
  Ps = 0;
  Div = Y - W - (1-Pf)*chi*Kss;  % investment is zero

  Eulerstock = (vh^(thetah-gamma))*(c_h^-thetah)*(Ps + Div);
  Eulerbondh = (vh^(thetah-gamma))*(c_h^-thetah);
  Eulerbondn = (vn^(thetan-gamma))*(c_n^-thetan);

  Inv = 0;
  Knext = 0;
  dIdK = (Inv/K) - (1/a1)*(xsi/(xsi-1))*(Inv/(K*((1/a1)*((Knext/K)-(1-delta)-a2))))*(Knext/K);
  Eulerf = (vh^(thetah-gamma))*(c_h^-thetah)*(alpha*Z*(K^(alpha-1)) - dIdK);

  equations;
     resid1;
     resid2;
  end;
end;

var_interp EEulerstock_interp EEulerbondh_interp EEulerbondn_interp EEulerf_interp Evh_interp Evn_interp EPD_interp EPD_square_interp;
initial EEulerstock_interp shock_trans*reshape(Eulerstock,shock_num,[]);
initial EEulerbondh_interp shock_trans*reshape(Eulerbondh,shock_num,[]);
initial EEulerbondn_interp shock_trans*reshape(Eulerbondn,shock_num,[]);
initial EEulerf_interp shock_trans*reshape(Eulerf,shock_num,[]);
initial Evh_interp shock_trans*reshape(vhpow,shock_num,[]);
initial Evn_interp shock_trans*reshape(vnpow,shock_num,[]);
initial EPD_interp shock_trans*reshape(Div,shock_num,[]);
initial EPD_square_interp shock_trans*reshape(Div.^2,shock_num,[]) / varianceScale;

EEulerstock_interp = shock_trans*Eulerstock;
EEulerbondh_interp = shock_trans*Eulerbondh;
EEulerbondn_interp = shock_trans*Eulerbondn;
EEulerf_interp = shock_trans*Eulerf;
Evh_interp = shock_trans*vhpow;
Evn_interp = shock_trans*vnpow;
EPD_interp = shock_trans*(Ps+Div);
EPD_square_interp = shock_trans*(Ps+Div).^2 / varianceScale;

% Endogenous variables, bounds, and initial values
var_policy c_h c_n Ps Pf Inv bn_shr_next lambdah lambdan;

inbound c_h 1e-3 100;
inbound c_n 1e-3 100;
inbound Ps 1e-3 500;
inbound Pf 1e-3 10;
inbound Inv 1e-9 100;
inbound bn_shr_next bn_shr_lb bn_shr_ub;
inbound lambdah 0 2;
inbound lambdan 0 2;

% Other equilibrium variables
var_aux Y W b_h b_n Div dIdKp Eulerstock Eulerbondh Eulerbondn dIdK Eulerf vhpow vnpow omega PDratio Rs R_ep vh vn Knext SharpeRatio;

model;
  Y = Z*(K^alpha);               % output
  W = (1-alpha)*Z*(K^alpha);     % Wage = F_l
  Div = Y - W - Inv - (1-Pf)*chi*Kss;             % dividends

  Knext = (1-delta)*K + (a1*((Inv/K)^((xsi-1)/xsi))+a2)*K;
  dIdKp = (1/a1)*(xsi/(xsi-1))*(Inv/(K*((1/a1)*((Knext/K)-(1-delta)-a2))));

  b_h = (1-bn_shr)*chi*Kss/mu;
  b_n = bn_shr*chi*Kss/(1-mu);

  [EEulerstock_future,EEulerbondh_future,EEulerbondn_future,EEulerf_future,Evh_future,Evn_future,EPD_future,EPD_square_future] =
       GNDSGE_INTERP_VEC(shock,Knext,bn_shr_next);
  EPD_square_future = EPD_square_future*varianceScale;

  vh =  ((1-beta)*(c_h^(1-thetah)) + beta*(Evh_future^((1-thetah)/(1-gamma))))^(1/(1-thetah));
  vn =  ((1-beta)*(c_n^(1-thetan)) + beta*(Evn_future^((1-thetan)/(1-gamma))))^(1/(1-thetan));

  Eulerstock = (vh^(thetah-gamma))*(c_h^-thetah)*(Ps + Div);
  Eulerbondh = (vh^(thetah-gamma))*(c_h^-thetah);
  Eulerbondn = (vn^(thetan-gamma))*(c_n^-thetan);
```

```
  dIdK = (Inv/K) - (1/a1)*(xsi/(xsi-1))*(Inv/(K*((1/a1)*((Knext/K)-(1-delta)-a2))))*(Knext/K);
  Eulerf = (vh^(thetah-gamma))*(c_h^-thetah)*(alpha*Z*(K^(alpha-1)) - dIdK);

  vhpow = vh^(1-gamma);
  vnpow = vn^(1-gamma);

  omega = (Ps+Div+ mu*b_h)/(Ps+Div+chi*Kss);
  PDratio = Ps/Div;
  Rs = EPD_future/Ps;
  R_ep = Rs - 1/Pf;
  % The following inline implements
  % std_ExcessR = (GNDSGE_EXPECT{(PD_future'/Ps - Rs)^2})^0.5;
  std_ExcessR = (EPD_square_future/(Ps^2) + Rs^2 - 2*EPD_future*Rs/Ps)^0.5;
  SharpeRatio = R_ep/std_ExcessR;

  % Equations:
  err_bdgt_h = 1 - (W + (Div/mu) + b_h - Pf*(chi*Kss*(1-bn_shr_next)/mu))/c_h; % these are individual consumptions
  err_bdgt_n = 1 - (W + b_n - Pf*(bn_shr_next*chi*Kss/(1-mu)))/c_n;
  foc_stock = 1 - (beta*EEulerstock_future*(Evh_future^((gamma-thetah)/(1-gamma))))/((c_h^(-thetah))*Ps);
  foc_bondh = 1 - (beta*EEulerbondh_future*(Evh_future^((gamma-thetah)/(1-gamma))) + lambdah)/((c_h^(-thetah))*Pf);
  foc_bondn = 1 - (beta*EEulerbondn_future*(Evn_future^((gamma-thetan)/(1-gamma))) + lambdan)/((c_n^-thetan)*Pf);
  foc_f = 1 - (beta*EEulerf_future*(Evh_future^((gamma-thetah)/(1-gamma))))/((c_h^(-thetah))*dIdKp);

  slack_bn = lambdan*(bn_shr_next - bn_shr_lb);    %mun_lw*bn_shr_next;
  slack_bh = lambdah*(bn_shr_ub - bn_shr_next);    %mun_up*(1-bn_shr_next);

  equations;
    err_bdgt_h;
    err_bdgt_n;
    foc_stock;
    foc_bondh;
    foc_bondn;
    foc_f;
    slack_bn;
    slack_bh;
  end;

end;

simulate;
  num_periods = 3000;
  num_samples = 100;

  initial K Kss;
  initial bn_shr 0.5;
  initial shock 2;

  var_simu Y c_h c_n Inv Ps Div Pf bn_shr_next Knext omega PDratio Rs EPremium SharpeRatio;

  K' = Knext;
  bn_shr' = bn_shr_next;
end;
```

A.2 Bianchi (2011)

```
% Toolbox options
USE_ASG=1; USE_SPLINE=0;
AsgMaxLevel = 10;
AsgThreshold = 1e-4;

% Parameters
parameters r sigma eta kappaN kappaT omega beta;
r = 0.04;
sigma = 2;
eta = 1/0.83 - 1;
kappaN = 0.32;
kappaT = 0.32;
omega = 0.31;
beta = 0.91;

% States
```

```
var_state b;
bPts = 101;
bMin=-0.5;
bMax=0.0;
b=linspace(bMin,bMax,bPts);

% Shocks
var_shock yT yN;
yPts = 4;
shock_num=16;

yTEpsilonVar = 0.00219;
yNEpsilonVar = 0.00167;
rhoYT = 0.901;
rhoYN = 0.225;

[yTTrans,yT] = markovappr(rhoYT,yTEpsilonVar^0.5,1,yPts);
[yNTrans,yN] = markovappr(rhoYN,yNEpsilonVar^0.5,1,yPts);

shock_trans = kron(yNTrans,yTTrans);
[yT,yN] = ndgrid(yT,yN);
yT = exp(yT(:)');
yN = exp(yN(:)');

% Define the last-period problem
var_policy_init dummy;
inbound_init dummy -1.0 1.0;

var_aux_init c lambda;
model_init;
  cT = yT + b*(1+r);
  cN = yN;
  c = (omega*cT^(-eta) + (1-omega)*cN^(-eta))^(-1/eta);
  partial_c_partial_cT = (omega*cT^(-eta) + (1-omega)*cN^(-eta))^(-1/eta-1) * omega * cT^(-eta-1);
  lambda = c^(-sigma)*partial_c_partial_cT;

  equations;
    0;
  end;
end;

% Implicit state transition functions
var_interp lambda_interp;
initial lambda_interp lambda;
lambda_interp = lambda;

% Endogenous variables, bounds, and initial values
var_policy nbNext mu cT pN;
inbound nbNext 0.0 10.0;
inbound mu 0.0 1.0;
inbound cT 0.0 10.0;
inbound pN 0.0 10.0;

var_aux c lambda bNext;
var_output bNext pN;

model;
  % Non tradable market clear
  cN = yN;

  % Transform variables
  bNext = nbNext - (kappaN*pN*yN + kappaT*yT);
  % Interp future values
  lambdaFuture' = lambda_interp'(bNext);

  % Calculate Euler residuals
  c = (omega*cT^(-eta) + (1-omega)*cN^(-eta))^(-1/eta);
  partial_c_partial_cT = (omega*cT^(-eta) + (1-omega)*cN^(-eta))^(-1/eta-1) * omega * cT^(-eta-1);
  lambda = c^(-sigma)*partial_c_partial_cT;
  euler_residual = 1 - beta*(1+r) * GDSGE_EXPECT{lambdaFuture'}/lambda - mu;

  % Price consistent
  price_consistency = pN - ((1-omega)/omega)*(cT/cN)^(eta+1);

```

```
  % budget constraint
  budget_residual = b*(1+r)+yT+pN*yN - (bNext+cT+pN*cN);

  equations;
    euler_residual;
    mu*nbNext;
    price_consistency;
    budget_residual;
  end;
end;

simulate;
  num_periods = 1000;
  num_samples = 100;
  initial b 0.0
  initial shock 1;
  var_simu c pN;
  b' = bNext;
end;
```

A.3 Barro et al. (2017)

```
% Parameters
parameters rho nu mu gamma1 gamma2;
period_length=0.25;         % a quarter
rho = 0.02*period_length;   % time preference
nu = 0.02*period_length;    % replacement rate
mu = 0.5;                   % population share of agent 1
P = 1-exp(-.04*period_length);   % disaster probability
B = -log(1-.32);            % disaster size
g = 0.025*period_length;    % growth rate
gamma1 = 3.1;
gamma2 = 50;

% Shocks
var_shock yn;
shock_num = 2;
shock_trans = [1-P,P;
               1-P,P];
yn = exp([g,g-B]);

% States
var_state omega1;
Ngrid = 501;
omega1 = [linspace(0,0.03,200),linspace(0.031,0.94,100),linspace(0.942,0.995,Ngrid-300)];

p = (1-nu)/(rho+nu);
pn = p;
Re_n = (1+pn)*yn/p;
% Endogenous variables, bounds, and initial values
var_policy shr_x1 Rf omega1n[2]
inbound shr_x1 0 1;             % agent 1's equity share
inbound Rf Re_n(2) Re_n(1);     % risk-free rate
inbound omega1n 0 1.02;         % state next period

% Other equilibrium variables
var_aux x1 x2 K1 b1 c1 c2 log_u1 log_u2 expectedRe;

% Implicit state transition functions
var_interp log_u1future log_u2future;
log_u1future = log_u1;
log_u2future = log_u2;
initial log_u1future (rho+nu)/(1+rho)*log((rho+nu)/(1+rho)) + (1-nu)/(1+rho)*log((1-nu)/(1+rho));
initial log_u2future (rho+nu)/(1+rho)*log((rho+nu)/(1+rho)) + (1-nu)/(1+rho)*log((1-nu)/(1+rho));

model;
  c1 = (rho+nu)/(1+rho);
  c2 = (rho+nu)/(1+rho);
  p = (1-nu)/(rho+nu);
  pn = p;

  log_u1n' = log_u1future'(omega1n');
```

```
  log_u2n' = log_u2future'(omega1n');
  u1n' = exp(log_u1n');
  u2n' = exp(log_u2n');

  Re_n' = (1+pn)*yn'/p;
  x1 = shr_x1*(Rf/(Rf - Re_n(2)));

  % Market clearing for bonds:
  b1 = omega1*(1-x1)*(1-c1)*(1+p);
  b2 = -b1;
  x2 = 1 - b2/((1-omega1)*(1-c2)*(1+p));
  K1 = x1*(1-c1)*omega1*(1+p)/p;
  K2 = x2*(1-c2)*(1-omega1)*(1+p)/p;

  R1n' = x1*Re_n' + (1-x1)*Rf;
  R2n' = x2*Re_n' + (1-x2)*Rf;

  % Agent 1's FOC wrt equity share:
  eq1 = GDSGE_EXPECT{Re_n'*u1n'^(1-gamma1)*R1n'^(-gamma1)} / GDSGE_EXPECT{Rf*u1n'^(1-gamma1)*R1n'^(-gamma1)} - 1;

  % Agent 2's FOC wrt equity share:
  log_u2n_R2n_gamma' = log_u2n'*(1-gamma2) - log(R2n')*gamma2;
  min_log_u2n_R2n_gamma = GDSGE_MIN{log_u2n_R2n_gamma'};
  log_u2n_R2n_gamma_shifted' = log_u2n_R2n_gamma' - min_log_u2n_R2n_gamma;
  eq2 = GDSGE_EXPECT{Re_n'*exp(log_u2n_R2n_gamma_shifted')} / GDSGE_EXPECT{Rf*exp(log_u2n_R2n_gamma_shifted')} - 1;

  % Consistency for omega:
  omega_future_consis' = K1 - nu*(K1-mu) + (1-nu)*Rf*b1/(yn'*(1+pn)) - omega1n';

  % Update the utility functions:
  ucons1 = ((rho+nu)/(1+rho))*log(c1) + ((1-nu)/(1+rho))*log(1-c1);
  ucons2 = ((rho+nu)/(1+rho))*log(c2) + ((1-nu)/(1+rho))*log(1-c2);
  log_u1 = ucons1 + (1-nu)/(1+rho)/(1-gamma1)*log(GDSGE_EXPECT{(R1n'*u1n')^(1-gamma1)});
  log_u2 = ucons2 + (1-nu)/(1+rho)/(1-gamma2)*( log(GDSGE_EXPECT{R2n'*exp(log_u2n_R2n_gamma_shifted')}) + min_log_u2n_R2n_gamma );

  expectedRe = GDSGE_EXPECT{Re_n'};

  equations;
    eq1;
    eq2;
    omega_future_consis';
  end;
end;

simulate;
  num_periods = 10000;
  num_samples = 50;
  initial omega1 .67;
  initial shock 1;

  var_simu Rf K1 b1 expectedRe;

  omega1' = omega1n';
end;
```

Appendix B User Manual