Global DSGE Models*

Dan Cao†
Georgetown University

Wenlan Luo
Tsinghua University

Guangyu Nie
Shanghai University of Finance and Economics

June 3, 2021

Abstract

We introduce our GDSGE framework and a novel global solution method, called simultaneous transition and policy function iterations (STPFIs), for solving dynamic stochastic general equilibrium models. The framework encompasses many well-known incomplete markets models with highly nonlinear dynamics such as models of financial crises and models with rare disasters including the current COVID-19 pandemic. Using consistency equations, our method is most effective at solving models featuring endogenous state variables with implicit laws of motion such as wealth or consumption shares. Finally, we incorporate this method in a publicly available toolbox that solves many important models in the aforementioned topics, and in many cases, more efficiently or accurately than their original algorithms.

Keywords: nonlinear DSGE models, incomplete markets, financial crises, rare disasters, portfolio choices, occasionally binding constraints

JEL Classifications: C68, E00, F00, G00

*First version: 02/15/2020. An online compiler server of the toolbox is deployed on the toolbox's website: http://www.gdsge.com. Currently the toolbox and libraries are only accessible through the website and we have been soliciting and collecting user feedback to improve the toolbox. But once we have finalized the toolbox design and implementation, we will make the whole toolbox (including compiler, C++ libraries and their source codes) publicly available. For useful comments and discussions, we thank Martin Evans, Luca Guerrieri, Fatih Guvenen, Mark Huggett, Toshi Mukoyama, John Rust, Gianluca Violante, Tim Landvoigt, Ivan Wérning, 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.

†Department of Economics, Georgetown University, 3700 and O Streets, N.W., Washington DC 20057, USA (e-mail: dan.cao@georgetown.edu)

1 Introduction

Dynamic stochastic general equilibrium (DSGE) models are an important tool in the study of business cycles and monetary and fiscal policies. The introduction of a general framework and local solution methods in Blanchard and Kahn (1980), Uhlig (1999), and Sims (2002) among others, and the toolbox Dynare, which implements these local methods, have made it easy to solve and estimate DSGE models and have enabled a large number of important academic studies and policy applications. 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 during financial crises, in closed or open economies such as Mendoza (2010), Bianchi (2011), Brunnermeier and Sannikov (2014), and Cao et al. (2019);
- implications of rare disasters such as Barro (2006), and Guerrieri et al. (2020) (this last paper studies the impact of the current COVID-19 pandemic);
- portfolio choices and their implications such as Heaton and Lucas (1996) and Guvenen (2009);
- models with occasionally binding constraints (e.g, borrowing constraints and the zero lower bound on monetary policy) such as Gust et al. (2017), Guerrieri and Iacoviello (2017), and Cao et al. (2019);
- international finance models with endogenous capital accumulation and/or portfolio choices such as Bocola (2016), Coeurdacier et al. (2019), and Cao et al. (2020);
- and many more.

Despite these important developments, there has not been a unified framework and solution method for the global solutions of DSGE models. This paper offers such a framework and method.

First, we develop a general framework that encompasses many recent well-known models and their extensions. The framework consists of state variables, policy variables, and short-run equilibrium conditions, e.g., market clearing conditions and Euler equations,[1] that fully describe sequential equilibrium. A crucial difference between this

[1] We use an important result in constrained optimization that for *globally* concave optimization problems, short-run first-order conditions such as the Euler equations and complementary-slackness conditions are necessary and sufficient for optimality. Duffie et al. (1994, Proposition 3.2) provide a proof for such a result in the context of an incomplete markets DSGE model.

framework and the standard DSGE framework is that the state variables and their *global* domain need to be specified, hence we name it G(lobal) DSGE. In the framework, a recursive equilibrium is a mapping from current state variables to current policy variables (policy function) and future state variables (transition function). In this framework, we develop a new and general algorithm to to robustly and efficiently solve for recursive equilibria. The main idea of the algorithm is to solve jointly for policy and transition functions over the iterations. Hence, we call it *Simultaneous Transition and Policy Function Iterations (STPFI)* to differentiate it from the standard Policy Function Iteration (PFI) algorithms in the existing literature (e.g., Coleman (1990, 1991) and Judd (1992)), which we discuss below.

Second, we develop a toolbox that implements the algorithm. The toolbox is similar to Dynare in that it allows users to write models using intuitive and simple scripts, despite requiring users to specify the state and policy variables and the ranges for state variables explicitly, due to the nature of global solutions. One of the challenges associated with the implementation of our algorithm is to solve for a large number of nonlinear equations with many unknowns, including future endogenous state variables in the STPFI algorithm and unknowns that may need to satisfy inequality constraints (e.g. the Lagrangian multiplier that enters a complementarity-slackness condition). The toolbox addresses this challenge by using an efficient equation solver that is able to explicitly enforce bounds of unknowns,[2] combined with an automatic differentiation method that is able to compute the Jacobian matrix up to machine precision with small costs.[3] Besides,

[2]We use a constrained dogleg algorithm developed by Bellavia et al. (2012), which is tailored for nonlinear systems with simple bound constraints imposed on unknowns. The algorithm modifies a conventional trust-region-dogleg search step (Powell, 1970) by an affine-scaling transformation (Heinkenschloss et al., 1999), so that the search always stays within the predetermined bounds of unknowns. Adopting such an algorithm avoids using a general constrained optimization routine to deal with the inequality constraints, which is usually less efficient than the dogleg method for solving equations, and avoids using ad-hoc transformations to convert constrained equations into unconstrained ones (as done in Cao (2018) and Elenev et al. (2021), for example, $x = (\sin(y))^2$ if $0 \leq x \leq 1$).

[3]Automatic differentiation (AD) is different from the finite-difference method or analytical differentiation method that are more familiar to economists. AD exploits the fact that every computer program, no matter how complicated, executes a sequence of elementary arithmetic operations, of which the derivatives can be evaluated analytically. By applying the chain rule repeatedly to these operations, the derivative of a complex function can be computed along calculating the value of the function. Because at each step of the chain rule the derivative is evaluated analytically, AD can calculate gradients up to the machine precision by using only a *constant* factor more operations than evaluating the original function, and is thus more accurate and much faster than the finite-difference method which requires perturbing the function along each dimension of the arguments with a finite step size. Moreover, because the chain rules are applied automatically, AD saves the hassle of taking analytical derivatives of a complex function, and applies to functions whose analytical derivatives are state-dependent (e.g., the max or min operator which enters a complementarity constraint). Our implementation of AD further utilizes the expression template feature of C++, so many of the calculations are done at compile time, which significantly increases the efficiency of gradient evaluations at run time.

the toolbox implements all actual computations in C++ to maximize the performance, and provides a MATLAB interface that allows users to specify options and generate model output conveniently. With such a design, the toolbox is able to solve many classical models that are well-known for computational challenges with a few lines of toolbox codes and in a few minutes on a regular laptop (more details on the run-time is provided in the paper).

The algorithm demonstrates its greatest power, relative to other methods, for models with endogenous state variables with implicit laws of motion, 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 multiple assets such as Heaton and Lucas (1996), Kubler and Schmedders (2003), and Cao (2018), or help simplify the feasible region of the endogenous state space in models with a collateral constraint such as Mendoza (2010) and Cao and Nie (2017). The endogenous state variables also help circumvent multiple equilibria issues as demonstrated in Cao et al. (2019). Our STPFI algorithm includes the vectors of future realizations of endogenous state variables in the vector of unknowns to be solved at each collocation point over the iterations. The additional equations in the system of equations at each collocation point are the *consistency equations* that impose the future endogenous state variables to be consistent with current policy variables.

We provide many examples of existing seminal applications that can be solved easily using the toolbox. The examples in the paper include Heaton and Lucas (1996), Guvenen (2009), Mendoza (2010), Bianchi (2011), Barro et al. (2017), and a dynamic stochastic extension of Guerrieri et al. (2020). Each of the examples listed can be implemented within 200 lines of toolbox code and can be executed in a few minutes on regular laptops. In many cases, the toolbox algorithm is fundamentally different from the original solution methods and is either more efficient and faster or more accurate, or both. For example, in Guvenen (2009), his solution method is based on the algorithm in Krusell and Smith (1998) using fixed point iterations over the pricing and aggregate state transition functions with nested value function iterations. Our algorithm recognizes that because the agents' optimization problems are *globally* concave problems, the first-order conditions are sufficient for optimality (without having to solve the agents' Bellman equation). Therefore, we can directly use STPFIs to solve jointly for agents' optimization problems and market clearing conditions. As a result, our toolbox's solution method is significantly faster and more stable than the original one in Guvenen (2009). The baseline model in Guvenen (2009) can be solved in less 3 minutes on a regular laptop (MacBook Air 2.2 GHz Intel Core i7) while it takes about 9 hours 15 minutes (on a MacProWorkstation with 8-core 3.2 GHz Xeon processors) using the original algorithm in Guvenen (2009) as reported in Appendix A in his paper.

In another example, Barro et al. (2017) solve an incomplete markets model with rare disasters and heterogenous risk-aversion using a mixture of projection and perturbation methods developed in Fernández-Villaverde and Levintal (2018) – while our toolbox's algorithm is a purely projection method using wealth share as an endogenous state variable with an implicit law of motion accommodated by STPFI. As Barro et al. (2017) discuss in their paper, their solution method is not sufficiently accurate for large values of risk-aversion coefficients (up to 10). We show that our method can tackle cases with risk-aversion coefficients as high as 100 effectively and uncover new economic insights in these cases. We provide many more examples on the toolbox's website.

Since our toolbox solves explicitly for policy functions and transition functions, we can conveniently use them to carry out numerical accuracy analyses in terms of Euler equation errors along the lines in Judd (1992), Judd et al. (2011), and Guerrieri and Iacoviello (2015). To illustrate this feature, we provide the toolbox code for these analyses for our leading example from Heaton and Lucas (1996). The toolbox can also automatically generates Monte Carlo simulations. These simulations are useful to understand the transmission mechanisms in the models using impulse response functions and to eventually estimate the models. The code to generate impulse response functions from the dynamic extension of Guerrieri et al. (2020) is available on the toolbox's website.

The examples mentioned above are highly nonlinear models which require solutions with high accuracy but are of small scale (up to three continuous state variables). The policy and transition functions in these models could be well approximated using spline approximations. Cao et al. (2019) is a New Keynesian application with five continuous state variables using piece-wise linear approximation. Cubic spline or adaptive sparse grid approximation does not work well for the model because the policy functions exhibit sharp changes in slopes in regions where the zero lower bound or collateral constraint switches from non-binding to binding and vice-versa. However, in principle, our toolbox can accommodate medium scale DSGE models of up to 40 continuous state variables such as Smets and Wouters (2007) which features around 20 continuous state variables.[4] For these models with larger numbers of state variables, we provide the option of using the adaptive sparse grid method developed by Ma and Zabaras (2009) and introduced to economics by Brumm and Scheidegger (2017). This method works well when the nonlinearity in the models is not as extreme as in Cao et al. (2019).

[4]Notice that, in general, highly nonlinear models with fewer state variables could potentially be more difficult to solve than less nonlinear model with more state variables.

Our approach to solving models with endogenous state variables with implicit laws of motion is different from the existing approaches in the literature. There are two main existing approaches. One approach is the nested-fixed point algorithm proposed by Kubler and Schmedders (2003). The algorithm is based on the existing PFI algorithm. In each iteration, the authors solve for future endogenous state variables using the consistency equations as an additional fixed point problem. The solution to the fixed-point problem is then used to formulate a system of equations and unknowns for current policy variables. This nested-fixed point procedure might be unstable [5] and is not amenable to a simple and automated toolbox implementation. Another approach to tackle the implicit laws of motion is, in each policy function iteration, to use the transition function from the previous iteration and only solve for the policy function, instead of solving for both simultaneously as in our toolbox. Combining the transition function with the policy function from the previous iteration, one can obtain the forecasts of future variables. This approach is used in more recent papers such as Elenev et al. (2021). We call this method *transition function iterations* (TFI).[6] The advantage of our approach is that the equilibrium in each iteration corresponds to the equilibrium in a finite-horizon economy. For many models, this property guarantees the existence of solution to the short-run equilibrium systems. The convergence of the policy functions over the iterations thus reflects the convergence of equilibria in finite-horizon economies to those of the infinite-horizon economies, consistent with the theoretical proofs for equilibrium existence in infinite-horizon incomplete markets economies such as Duffie et al. (1994), Magill and Quinzii (1994), and more recently Cao (2020). The TFI approach does not always work for the examples considered in the present paper.[7]

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

[5]For example, the nested fixed point problems might not have a solution in early iterations when the initial guess for policy functions is far from the converged functions.

[6]Recently, Mendoza and Villalvazo (2020) show that this method yields significant speed improvement for models with explicit laws of motion, including Mendoza (2010)'s model with natural state variables.

[7]This is probably because the approach does not allow for feedback from future variables on current policy variables within each iteration. Some of the feedback effects can be introduced by manually setting dampening parameters. However, it is possible that for other models, our approach does not work and the alternative approach does. Therefore, in the toolbox, we allow for the option to use the alternative approach (The details are provided in Online Appendix A.1, in particular footnote 28).

(2010)'s algorithm uses PFIs based on earlier work using PFIs for DSGE economies such as Coleman (1990, 1991) and Judd (1992).

Our framework is more readily applicable to solving, globally, DSGE models with a finite number of agents or, more precisely, a finite number of agent types.[8] 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 on 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 state variables. Then, with an additional fixed-point iteration on these laws of motion, which can be coded 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 fewer than 200 lines of code.

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

2 A Leading Example

We use the benchmark model in Heaton and Lucas (1996) as the first illustration of 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 markets 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 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 notation, we omit the explicit dependence on the shock history, e.g., p_t^s stands for $p_t^s(z^t)$.

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

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

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

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

where Y_t^a 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 $\gamma_t^a = \frac{Y_t^a}{Y_{t-1}^a}$. $z_t = \{\gamma_t^a, \delta_t, \eta_t^1\}$ 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., $K_t^b = \bar{K}^b Y_t^a$.

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

$$s_t^1 + s_t^2 = 1, \text{ and } b_t^1 + b_t^2 = 0.$$

As in Kubler and Schmedders (2003), Brunnermeier and Sannikov (2014), and Cao (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: $\hat{x}_t = \frac{x_t}{Y_t^a}$ (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 is captured by first-order conditions in s^i_{t+1} and b^i_{t+1}:

$$1 = \beta \mathbb{E}_t \left[\left(\frac{\hat{c}^i_{t+1}}{\hat{c}^i_t} \right)^{-\gamma} (\gamma^a_{t+1})^{1-\gamma} \frac{\hat{p}^s_{t+1} + \hat{d}_{t+1}}{\hat{p}^s_t} \right] + \hat{\mu}^{i,s}_t,$$
$$1 = \beta \mathbb{E}_t \left[\left(\frac{\hat{c}^i_{t+1}}{c^i_t} \right)^{-\gamma} (\gamma^a_{t+1})^{-\gamma} \frac{1}{p^b_t} \right] + \hat{\mu}^{i,b}_t,$$

where $\hat{\mu}^{i,s}_t \geq 0$ and $\mu^{i,b}_t \geq 0$ are the Lagrangian multipliers on agent i's short-sale constraint and borrowing constraint, respectively. The multipliers and portfolio choices satisfy the complementary-slackness conditions:

$$0 = \hat{\mu}^{i,s}_t s^i_{t+1},$$
$$0 = \hat{\mu}^{i,b}_t (\hat{b}^i_{t+1} - \bar{K}^b).$$

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

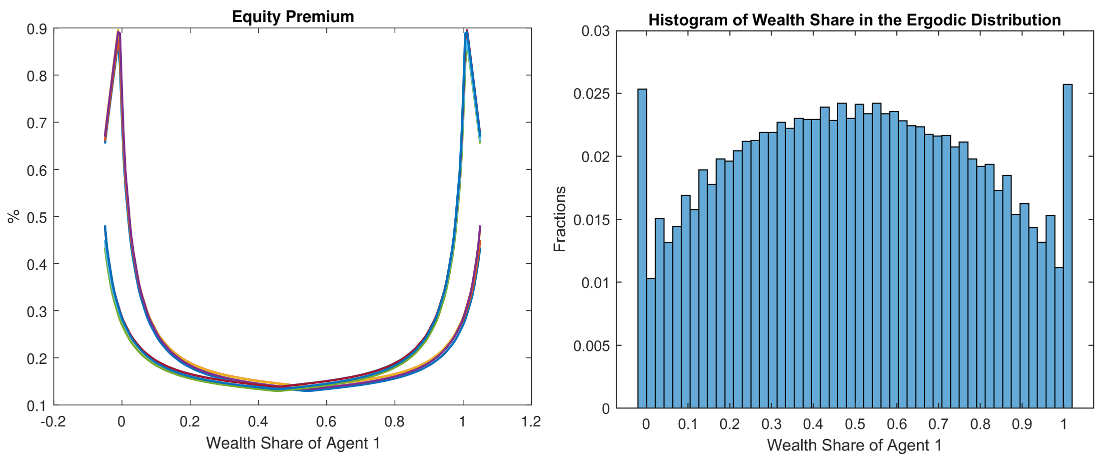
We solve the model using simultaneous policy and transition function iterations that we describe in more detail in Section 3: In each iteration, we take the policy functions solved from the last iteration as given, and look for pricing, allocation, and Lagrange multipliers as functions of wealth shares and exogenous states that satisfy the market clearing conditions and first-order conditions. The main challenge of doing so is immediately clear because to evaluate future prices and quantities, which show up in the first-order conditions, one needs to know the transition of the endogenous state, ω^1_{t+1} in the current example. However, ω^1_{t+1} depends on future prices and allocations which are themselves functions of the endogenous state variable, or in other words, the law of motion for the endogenous state variable is implicit. In dealing with such an endogenous state variable featuring an implicit law of motion, the literature has developed different methods such as the nested fixed point algorithm (Kubler and Schmedders, 2003) and the transition function iteration algorithm (Elenev et al., 2021) as surveyed in the introduction. A key innovation in our algorithm is that we incorporate consistency equations into the system of equations and unknowns so the transition of the endogenous state variable ω^1_{t+1} is solved jointly with policy variables. These equations require that the conjectured future endogenous state variables are consistent with the current portfolio choices and future prices:

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

We discuss these equations in more detail in Subsection 3.2 in the context of our general framework. The GDSGE code for the model that implements our algorithm is given in GDSGE code below (the consistency equations correspond to line 68 in the GDSGE code).

```
1  % Parameters
2  parameters beta gamma Kb;
3  beta = 0.95;  % discount factor
4  gamma = 1.5;  % CRRA coefficient
5  Kb = -0.05;   % borrowing limit in ratio of aggregate output
6  % Exogenous state variables
7  var_shock g d eta1;
8  % Enumerate exogenous states and transition matrix
9  shock_num = 8;
10 g = [.9904 1.0470 .9904 1.0470 .9904 1.0470 .9904 1.0470];
11 d = [.1402 .1437 .1561 .1599 .1402 .1437 .1561 .1599];
12 eta1 = [.3772 .3772 .3772 .3772 .6228 .6228 .6228 .6228];
13 shock_trans = [
14     0.3932 0.2245 0.0793 0.0453 0.1365 0.0779 0.0275 0.0158
15     0.3044 0.3470 0.0425 0.0484 0.1057 0.1205 0.0147 0.0168
16     0.0484 0.0425 0.3470 0.3044 0.0168 0.0147 0.1205 0.1057
17     0.0453 0.0793 0.2245 0.3932 0.0157 0.0275 0.0779 0.1366
18     0.1366 0.0779 0.0275 0.0157 0.3932 0.2245 0.0793 0.0453
19     0.1057 0.1205 0.0147 0.0168 0.3044 0.3470 0.0425 0.0484
20     0.0168 0.0147 0.1205 0.1057 0.0484 0.0425 0.3470 0.3044
21     0.0158 0.0275 0.0779 0.1365 0.0453 0.0793 0.2245 0.3932
22     ];
23 % Endogenous state variables
24 var_state w1;  % wealth share
25 w1 = linspace(-0.05,1.05,201);
26 % Policy variables and bounds that enter the equations
27 var_policy c1 c2 s1p nb1p nb2p ms1 ms2 mb1 mb2 ps pb w1n[8];
28 inbound c1 0 1;
29 inbound c2 0 1;
30 inbound s1p 0.0 1.0;
31 inbound nb1p 0.0 1.0;   % nb1p=b1p-Kb
32 inbound nb2p 0.0 1.0;
33 inbound ms1 0 1;        % Multipliers for constraints
34 inbound ms2 0 1;
35 inbound mb1 0 1;
36 inbound mb2 0 1;
37 inbound ps 0 1 adaptive(1.5);
38 inbound pb 0 1 adaptive(1.5);
39 inbound w1n -0.5 1.5;
40 % Other policy variables
41 var_aux equity_premium;
42 % Interpolation variables for state transitions
43 var_interp ps_future c1_future c2_future;
44 initial ps_future 0.0;
45 initial c1_future w1.*d+eta1;
46 initial c2_future (1-w1).*d+1-eta1;
47 ps_future = ps;
48 c1_future = c1;
49 c2_future = c2;
50
51 model;
52   % Evaluate interpolation
53   [psn',c1n',c2n'] = GDSGE_INTERP_VEC'(w1n');
54   % Calculate expectations that enter the Euler Equations
55   es1 = GDSGE_EXPECT{g'^(1-gamma)*(c1n'/c1)^(-gamma)*(psn'+d')/ps};
56   es2 = GDSGE_EXPECT{g'^(1-gamma)*(c2n'/c2)^(-gamma)*(psn'+d')/ps};
57   eb1 = GDSGE_EXPECT{g'^(-gamma)*(c1n'/c1)^(-gamma)/pb};
58   eb2 = GDSGE_EXPECT{g'^(-gamma)*(c2n'/c2)^(-gamma)/pb};
59   % Transform bond back
60   b1p = nb1p + Kb;
61   b2p = nb2p + Kb;
62   % Market clearing of shares
63   s2p = 1-s1p;
64   % Budget constraints
65   budget_1 = w1*(ps+d)+eta1 - c1 - ps*s1p - pb*b1p;
66   budget_2 = (1-w1)*(ps+d)+(1-eta1) - c2 - ps*s2p - pb*b2p;
67   % Consistency equations
68   w1_consis' = (s1p*(psn'+d') + b1p/g')/(psn'+d') - w1n';
69   % Other policy variables
70   equity_premium = GDSGE_EXPECT{(psn'+d')/ps*g'} - 1/pb;
71   equations;
72     -1+beta*es1+ms1;
73     -1+beta*es2+ms2;
74     -1+beta*eb1+mb1;
75     -1+beta*eb2+mb2;
76     ms1*s1p;
77     ms2*s2p;
78     mb1*nb1p;
79     mb2*nb2p;
80     b1p+b2p;
81     budget_1;
82     budget_2;
83     w1_consis';
84   end;
85 end;
86
87 simulate;
88   num_periods = 10000;
89   num_samples = 24;
90   initial w1 0.5;
91   initial shock 1;
92   var_simu c1 c2 ps pb equity_premium;
93   w1' = w1n';
94 end;
```

The code runs in less than a minute on a regular laptop (about 50 seconds on a MacBook Air with a 2.2 GHz Intel Core i7 CPU or 7 seconds on a desktop with Intel 2.5 GHz 12-Core CPU). It produces the policy functions, including equilibrium prices and allocation as functions of the endogenous state variable, agent 1's 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^1 increases. Panel (b) in Figure 1 shows the ergodic distribution of the endogenous state variable, ω^1.



(a) Policy Function for Equity Premium (b) Ergodic Distribution of Wealth Share

Figure 1: Policy Functions and Ergodic Distribution

Note: The model is solved with 8 realizations of the exogenous states and 201 grid points for the endogenous state. The histogram is based on 24 sample paths and 10,000 periods per sample path, with the first 1,000 periods dropped.

The model can also be solved using consumption share instead of wealth share as a state variable, as in Dumas 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. Dumas 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 that the agents' unconstrained Euler equations hold exactly. Our algorithm allows for deviation from the Euler equations due to binding portfolio constraints, such as borrowing constraints or short-sale constraints. The details of our implementation using our GDSGE toolbox are provided on the toolbox's website. In Section 4.1, we also show how to simplify the feasible region of the endogenous state-space in Mendoza (2010) using consumption as one endogenous state variable.

To assess the accuracy and speed of our new algorithm and the toolbox compared

Table 1: Accuracy and Speed of the Algorithm and Comparisons with Existing Methods

	Specification	number of collocation points	Running time (seconds)	Mean errors	Max errors
(1)	STPFI	1608	7	2.08E-05	3.40E-03
(2)	STPFI, dense grid	8040	24	9.09E-07	5.91E-04
(3)	STPFI, adaptive grid	1548	11	9.22E-06	5.78E-04
(4)	Transition Function Iteration	1608	47 (not converged)	2.57E-05	3.38E-03

Note: Running time is based on a desktop with an Intel 2.5 GHz 12-Core CPU. The mean and max Euler equation errors are calculated across 24 sample paths and 10,000 period per sample path, with the same random number draws across methods. All algorithms are implemented carefully so the performances can be fairly compared (see the text for detail). STFPI corresponds to the *Simultaneous Transition and Policy Function Iteration* algorithm developed in the current paper. All STFPI algorithms converge under the criterion that the metric between transition functions across two adjacent iterations is below 1E-6. Transition Function Iteration (TFI) corresponds to the algorithm proposed by Elenev et al. (2021), which at each time step fixes the transition function constructed from the previous iteration. The TFI algorithm can only converge under the convergence threshold of 2E-3 (compared to 1E-6 for STPFI) with an appropriate choice of dampening parameter. Algorithms (1) and (4) use the same grid for collocation points; (2) uses a denser grid; (3) uses an adaptive grid following Brumm and Scheidegger (2017) implemented by the toolbox.

with existing methods, we record the running time of solving the model with different specifications. We calculate the unit-free Euler equation errors for shares and bonds of agent $i \in \{1,2\}$ defined below, for a large sample drawn from the model's ergodic distribution:[9]

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

$$\mathcal{E}_t^{b,i} = -1 + \beta \mathbb{E}_t \left[\left(\frac{\hat{c}_{t+1}^i}{c_t^i} \right)^{-\gamma} \left(\gamma_{t+1}^a\right)^{-\gamma} \frac{1}{p_t^b} \right] + \hat{\mu}_t^{i,b}.$$

We then calculate the mean and max errors across all samples.

Table 1 reports the running time and mean and max Euler equation errors for different algorithms and compare them. Algorithm (1) corresponds our benchmark STFI algorithm which is implemented by the toolbox code listed before and produces the output in Figure 1. The algorithm uses a grid over ω_1 with 201 points and uses cubic splines to approximate the policy and transition functions. The mean and max unit-free Euler equation errors are at at the order of 2.1E-05 and 3.4E-03, which, transform

[9] To do so, we simulate 24 sample paths with 11,000 periods per sample path and drop the first 1000 periods of each sample path to collect a sample of size $24 \times 10,000$.

to \$0.14 and \$23 in \$10,000 consumption levels, respectively.[10] By increasing the number of grid points by 5 times, the Euler equation errors can be reduced by an order of magnitude, as shown in algorithm (2). However, this can be done more efficiently by using an adaptive-grid approach developed by Ma and Zabaras (2009) and Brumm and Scheidegger (2017). Our toolbox implements the algorithm efficiently and as shown in algorithm (3), with such method the (max) Euler equation errors are reduced by an order of magnitude, without increasing the number of grid points or increasing the running time significantly.[11]

We compare the performance of our algorithm and an alternative algorithm (4), TFI, used in Elenev et al. (2021). This algorithm is also based on time iterations, but unlike ours which solves the transition and policy functions jointly in each iteration, theirs fixes the state function transition implied by the policy functions solved from the previous iteration.[12] We implement this alternative algorithm with the same numerical routines (including the equation solver and functional approximation procedure) in C++, starting from the same initial transition function,[13] and fine-tuned for maximum numerical efficiency[14] so the performance of the two algorithms can be compared fairly. Even with dampened updating, algorithm (4) can only converge to the level of 2E-3 (measured by the metric between transition functions in adjacent iterations) compared to the level of 1E-6 for our algorithm. However, the Euler equation errors are comparable to ours under the final (non-converged) state transition function. The speed is significantly slower than STFPI, mainly because their algorithm fixes the state transition function and does not allow the future endogenous state (in the current example, future wealth share) to respond to current policy variables (share and bond choices etc.), which renders finding a solution to the equilibrium system more difficult.[15]

[10] The unit-free errors can be transformed to errors in consumption levels by multiplying them by $1/\gamma$.

[11] See Section 4.4 for another example which benefits even more from the adaptive-grid method due to the strong state-dependence of the model.

[12] Their algorithm can be also implemented using our toolbox. A detailed description of the algorithm and the implementation with the toolbox can be found in Online Appendix A.1, in particular footnote 28. We provide a more detailed comparison in the context of this Heaton and Lucas's model on the toolbox website: `https://www.gdsge.com/example/HL1996/HL1996TFIter.html`.

[13] The initial transition function is generated by the solution to the last-period problem of a finite-horizon economy.

[14] For example, for the alternative algorithm we pre-compute the expectation terms that do not depend on current policy variables under the fixed state transition functions. For the nested fixed point algorithm we use the solution of policy functions of the previous transition function iteration as "warm-ups" to increase the speed of convergence in the inner loop.

[15] Another reason why their algorithm is slower is because their algorithm needs to first evaluate the state transition function to get future states and then evaluate the policy functions to get future allocations and prices, whereas STFPI only needs to evaluate the policy functions as future endogenous states are solved simultaneously. However, the speed difference due to this reason is mitigated by that the future

In the next section we present the general framework beyond the specific example from Heaton and Lucas (1996).

3 General Environment

In this section, we provide the general framework and the solution algorithm (STPFI) to compute recursive equilibrium in this framework. In Online Appendix A, we present the design of the toolbox to implement the algorithm. In Section 4, 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 the 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 (aggregate productivity, aggregate growth rates, dividends, etc.), $s \in \mathcal{S} \subset \mathbb{R}^{d_s}$ is a vector of endogenous states variables (aggregate capital, wealth shares, consumption shares, etc.), and $x \in \mathcal{X} \subset \mathbb{R}^{d_x}$ is a vector of endogenous policy variables (asset prices, consumption, portfolio choices, Lagrange multipliers, etc.). 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.

For example, in the model in Heaton and Lucas (1996) described above $z = (\gamma^a, \delta, \eta)$, $s = (\omega^1)$, and $x = (\hat{c}^1, s^1, \hat{b}^1, \hat{c}^2, s^2, \hat{b}^2, p^s, p^b)$.

Notice that the framework allows for general dependence on the future variables, instead of through common expectations as in Winschel and Kratzig (2010). This generality is important to include models with state variables with implicit law of motion. We discuss this feature in more detail in Subsection 3.2. It also allows for non-rational expectations models including model with belief heterogeneity such as Sandroni (2000)

states and expectation terms can be pre-computed before solving the equilibrium system as they do not vary with current policy variables.

and Cao (2018). Lastly, it is necessary to capture nonlinear forms of borrowing constraints including the collateral constraints in Kiyotaki and Moore (1997) and Cao and Nie (2017).[16]

Models with inequality constraints 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)),

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

we can use $\hat{\boldsymbol{F}} = \begin{pmatrix} \boldsymbol{F} \\ \boldsymbol{G} - \eta \end{pmatrix}$ with $\eta \geq 0$,[17] and $\hat{x} = (x, \eta)$, to write the system with inequality constraint in form (1) for $\hat{\boldsymbol{F}}$ and $\hat{x}$.

In this general framework, we look for recursive equilibria defined below.

Definition A *recursive equilibrium* is a solution to (1) of 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.

The Simultaneous Transition and Policy Function Iteration (STPFI) Algorithm We solve for a recursive equilibrium of (1) using simultaneous transition and policy function iterations as follows. The algorithm starts with an initial guess for 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:

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

[16] 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 of different forms of collateral constraints.

[17] In numerical implementations, we tackle this with an equation solver that respects box constraints of unknowns.

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.

Unlike policy function iterations for Bellman equations for which the Contraction Mapping Theorem applies, good initial guesses are critical in ensuring that the algorithm works. Otherwise one might run into situations in which (2) does not have solution for some $(s, z) \in \mathcal{C}^{(n)}$ and hence cannot iterate further, or the iterations do not converge. We find that good typical initial guesses for $\mathcal{P}^{(0)}$ correspond to equilibria in the 1-period economy versions of the models. In this case, the solution for $\mathcal{P}^{(n)}$ corresponds to the equilibrium values of the first period in the (n+1)-period economy. Thus, 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 economies such as Duffie et al. (1994), Magill and Quinzii (1994), and more recently Cao (2020).

3.2 Explicit and Implicit State Transitions

Our toolbox demonstrates its greatest power, relative to other methods, for models with endogenous state variables with implicit state transition equations (laws of motion), such as wealth shares or consumption shares in the leading Heaton and Lucas' example. Here we explain the idea in more detail in the context of the general framework.

The state variables s may consist of state variables $\bar{s}$ that have explicit transition equations (laws of motions), and state variables $\bar{\bar{s}}$ that consist 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 include wealth shares or consumption shares, as in Section 2 for Heaton and Lucas (1996).

In this case, the system of equations (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 $\mathcal{P}$.[18]

The state variables with explicit state transitions 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, by substituting $\bar{g}(s,x,z,z')$ for $\bar{s}'$, we can work with $\bar{\boldsymbol{F}}$ which only takes the first and third components from $\boldsymbol{F}$:

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

The state variables with implicit laws of motion make it difficult to apply standard policy function iteration methods. Because one cannot easily calculate future state variables, and hence future equilibrium variables, in order to evaluate inter-temporal equilibrium conditions. To tackle this issue, the key innovation in our algorithm is, in $(n+1)^{\text{th}}$-iteration, to simultaneously solve for the unknowns x and $\{\bar{\bar{s}}'(z')\}_{z'\in\mathcal{Z}}$ given future policy function $\mathcal{P}^{(n)}$. To do so, we add to the system of equations (2) the consistency equations (3):

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

As discussed in the introduction, our approach to solving models with endogenous state variables with implicit laws of motion is different from the existing approaches in the literature. For example, Kubler and Schmedders (2003) use wealth shares as en-

[18]These consistency equations are similar to the consistency equations in the heterogeneous agent incomplete markets literature, including Huggett (1993, 1997), Aiyagari (1994), Krusell and Smith (1998), Reiter (2009) and Cao (2020), in which future cross-sectional distributions are consistent with current distributions and current policy functions.

dogenous state variables. These authors 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. This nested-fixed point procedure might be unstable. For example, the nested fixed point problems might not have a solution in early iterations when the initial guess for policy functions is far from the converged functions. In addition, the procedure is not amenable to a simple, automated toolbox implementation because the system of equations representing the nested-fixed point problems are completely different from the original equilibrium systems of equations. Elenev et al. (2021)'s method iterates over transition functions. As described in Section 2, this method can be implemented in our toolbox but it might not be as accurate and robust as our method. One of the possible reasons for lower accuracy is that the approach does not allow for feedback from future variables on current policy variables within each iteration, even though some of the feedback effect can be introduced by manually setting dampening parameters.

One potential concern with our method is that if the number of possible realizations of future exogenous shocks z' is too large, including $\{\bar{\bar{s}}'(z')\}_{z'\in\mathcal{Z}}$ and consistency equations in the system of equations and unknowns to be solved leads to a system that is too large. For example, if the true exogenous shocks ζ follow a VAR process

$$\zeta' = A\zeta + \epsilon',$$

one needs to approximate this process with a discrete-Markov process z with many points. To deal with this issue, we include ζ among the *endogenous* state variables s and discretize the innovation process ϵ' instead. Discretizing the innovation process requires a smaller number of discretization points and hence a smaller number of consistency equations.[19]

In Online Appendix A, we present the design and implementation of our toolbox for the general framework described above. In the next section, we present several important applications. Besides Heaton and Lucas (1996), the models from Mendoza (2010) and Barro et al. (2017) in Subsections 4.1 and 4.2 are other examples featuring state variables with implicit laws of motion.

[19]See the RBC model with irreversible investment on the toolbox's website (`http://www.gdsge.com/example/rbc/rbcIrr.html`) for a concrete example.

4 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 Online Appendix B and on the toolbox's website. 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. Following up on the leading example from Heaton and Lucas (1996), the next two examples from Mendoza (2010) and Barro et al. (2017) feature state variables with implicit laws of motion and consistency equations.

4.1 Sudden Stops with Asset Price Deflation by Mendoza (2010)

The seminal work by Mendoza (2010) builds a model that generates crises featuring current account reversals and asset price collapses, resembling those experienced by many emerging economies. The model has two endogenous assets—capital and bond, with an occasionally binding borrowing constraint tied to the capital price behind its key mechanism. The model is highly non-linear and state-dependent, and calls for a global solution method to study its rich dynamics. Although it has become a workhorse model in the study of crises in open and closed economies, researchers still find solving and extending this type of model challenging. We show that the model in Mendoza (2010) can be easily represented in our current framework, and solved with the toolbox in 200 lines of codes and within a minute on a regular laptop (about 55 seconds on a MacBook Air 2.2 GHz Intel Core i7).

Even though with the benchmark parameterization in Mendoza (2010), the model can be solved with capital and bond as the two endogenous states directly, one of the challenges researchers have frequently countered is that due to the borrowing constraint tied to the capital price, the feasible state space of capital and bond is not a rectangle and hard to determine ex-ante.[20] Our framework and toolbox addresses this challenge by transforming the endogenous states to those that have simple boundary conditions, and use *consistency equations* to specify the implicit transitions of these transformed states. This strategy frees researchers from trial-and-error guesses for the feasible space of states, a

[20] In particular, the maximum sustainable debt level that ensures positive consumption is increasing in capital stock, rendering the feasible space of capital and bond non-rectangle. Furthermore, this boundary cannot be determined ex-ante since the sustainable debt level depends on the new debt that the economy can finance, which depends on capital price and other *endogenous* variables determined in equilibrium.

procedure that is necessary but usually painful for global solution methods. We first briefly describe the model and then its representation in our framework. As for other examples, the toolbox code is provided in Online Appendix B and can be downloaded from the GDSGE website.

4.1.1 Model Description

Representative consumers in a small-open economy value consumption and leisure, with preferences represented by

$$\mathbb{E}_0 \sum_{t=0}^{\infty} \beta^t [u\left(c_t - N\left(L_t\right)\right)],$$

where $\beta \in (0,1)$ is the discount factor, $u(\cdot)$ is the GHH (Greenwood et al., 1988) utility function, c_t is consumption, L_t is the labor supply, and $N(\cdot)$ is a convex function representing the disutility from labor supply. $\mathbb{E}_0$ is the expectation operator integrating aggregate shocks introduced below.

Output is produced by representative firms combining capital k_t, labor L_t, and imported inputs v_t according to a constant-return-to-scale production function

$$\exp(\varepsilon_t^A)F(k_t, L_t, v_t),$$

where ε_t^A is the aggregate productivity shock. Investment is subject to convex adjustment costs

$$i_t = k_{t+1} - (1-\delta)k_t + \frac{1}{2}a\frac{(k_{t+1}-k_t)^2}{k_t}, \tag{4}$$

where $a > 0$ is a parameter determining the level of adjustment cost. The end-of-period capital price is competitively determined as

$$q_t = \frac{\partial i_t}{\partial k_{t+1}} = 1 + a\frac{k_{t+1}-k_t}{k_t}. \tag{5}$$

A fraction of the cost of labor and imported inputs needs to be financed by with-in-period loans obtained from foreign lenders, at the world gross interest rate of $R_t = R\exp(\varepsilon_t^R)$, where R is the mean interest rate and ε_t^R is an exogenous shock. The price of imported goods is $p_t = p\exp(\varepsilon_t^p)$, where p is the mean price and ε_t is an exogenous shock. $(\varepsilon_t^A, \varepsilon_t^R, \varepsilon_t^p)$ are the shocks in the economy and follow first-order Markov processes.

Consumers can trade a one-period state non-contingent bond b_{t+1} (a negative value of b_{t+1} denotes borrowing) with the rest of the world, at price $q_t^b = \frac{1}{R_t}$. Therefore, the budget constraint of the consumers reads:

$$c_t + i_t = \exp(\varepsilon_t^A)F(k_t, L_t, v_t) - p_t v_t - \phi\left(R_t - 1\right)\left(w_t L_t + p_t v_t\right) - q_t^b b_{t+1} + b_t,$$

of which the left hand side is domestic consumption and investment, and the terms on the right hand side are in order (1) domestic gross output, (2) minus costs of imported goods, (3) minus interest payments from working capital loans, (4) minus costs of bond holding into the future period, and (5) plus current bond payments. It is assumed that the inter-temporal borrowing is subject to a collateral constraint:

$$q_t^b b_{t+1} - \phi R_t \left(w_t L_t + p_t v_t\right) \geq -\kappa q_t k_{t+1},$$

which says that total debt, including debt in one–period bonds and working capital loans, does not exceed κ fraction of the market value of end-of-period capital. The representative consumers maximize the utility specified earlier, subject to the budget and collateral constraints described above. We next represent the model equilibrium with a system of equations, which are enabled by that the optimality conditions are replaced with first-order and complementary slackness conditions.

A sequential competitive equilibrium is stochastic sequences $\{\mu_t, c_t, w_t, L_t, v_t, b_{t+1}, k_{t+1}\}_{t=0}^{\infty}$ such that

$$\begin{aligned}
&c_t + i_t = \exp(\varepsilon_t^A)F(k_t, L_t, v_t) - p_t v_t - \phi\left(R_t - 1\right)\left(w_t L_t + p_t v_t\right) - q_t^b b_{t+1} + b_t,\\
&-q_t^b \lambda_t + \mu_t q_t^b + \beta \mathbb{E}_t \lambda_{t+1} = 0,\\
&\beta \mathbb{E}_t \lambda_{t+1} \exp(\varepsilon_{t+1}^A) F_1(k_{t+1}, L_{t+1}, v_{t+1}) - \beta \mathbb{E}_t \lambda_{t+1} \frac{\partial i_{t+1}}{\partial k_{t+1}} - \lambda_t q_t + \mu_t \kappa q_t = 0,\\
&[q_t^b b_{t+1} - \phi R_t \left(w_t L_t + p_t v_t\right) + \kappa q_t k_{t+1}]\mu_t = 0, \mu_t \geq 0,\\
&\exp(\varepsilon_t^A) F_2(k_t, L_t, v_t) = w_t(1 + \phi(R_t - 1) + \frac{\mu_t}{\lambda_t}\phi R_t),\\
&\exp(\varepsilon_t^A) F_3(k_t, L_t, v_t) = p_t(1 + \phi(R_t - 1) + \frac{\mu_t}{\lambda_t}\phi R_t),\\
&w_t = N'(L_t),
\end{aligned}$$

where, $\lambda_t = u'(c_t - N(L_t))$, i_t is given by equation (4), and q_t is given by equation (5). Among the unknowns, μ_t is the Lagrangian multiplier placed on the collateral constraint. In the system of equations, the first is the budget constraint; the second the Euler equation for bond; the third the Euler equation for capital; the fourth the complementary slackness condition for the collateral constraint; the fifth and sixth are the optimality conditions for using labor and imported inputs, taking into account the interest payments on working capital loans and the shadow prices arising from the collateral constraint; the last is the optimality condition for labor supply. The equilibrium system can be casted in a recursive form with capital and bond as the endogenous states. Before describing the representation of the model using our framework, we discuss the state transformation and the associated consistency equations.

The motivation for state transformation, as described earlier, is that the feasible set of capital k_t and bond b_t may not be a rectangle, and the fact that the borrowing constraint depends on equilibrium capital price renders determining the feasible set ex-ante difficult. Instead, by noticing that the exact boundary condition of the feasible set is the consumption-labor bundle, $\tilde{c}_t \equiv c_t - N(L_t)$, being non-negative, we can transform the endogenous states from (k_t, b_t) to $(k_t, \tilde{c}_t)$. Unlike b_t, for which the period-$(t+1)$ bond is a policy variable and thus admits an explicit transition, the transition of $\tilde{c}_t$ is implicitly imposed by the condition that $\tilde{c}_{t+1}$ should be consistent with the choice of b_{t+1} via the future budget constraint. These conditions are precisely the *consistency equations* (equations (3) in the general framework) that should be included as part of the system of equations to be solved in each iteration. Therefore, with such transformation, the equation system described before should be modified to include the vector of $\tilde{c}_{t+1}$ for each realization of future shocks as unknowns, and correspondingly to replace the current budget constraint with the future budget constraints for each realization of shocks.

4.1.2 Computation

We use the functional forms and parameterization from the benchmark calibration in Mendoza (2010).[21] The first set of results concerns about the motivation for the state transformation discussed earlier. The left panel of Figure 2 projects the endogenous state $\tilde{c}$, over the natural state space (k, b).[22] As shown, for the rectangle state space defined over $(k, \tilde{c})$, the implied space of (k, b) is not rectangle—in fact, the value of b that is consistent with the same value of $\tilde{c}$ is increasing in k. Choosing a uniform lower bound for b too low may result in equilibrium non-existence for low values of k, since no positive consumption may exist at such collocation points. On the other hand, choosing a uniform lower bound for b too high may miss some combinations of (k, b)

[21] More detailed comparisons between the current results and those in Mendoza (2010), such as on business cycle moments, can be found on the toolbox website.

[22] Notice that the equilibrium is solved over the endogenous states $(k, \tilde{c})$. For each combination of $(k, \tilde{c})$ and realization of exogenous shocks, there is a value of b that is consistent with equilibrium conditions. We can therefore project policy functions defined over $(k, \tilde{c})$ to the space of (k, b) as long as b is monotone in $\tilde{c}$ given k and realizations of exogenous shocks, which is validated by the solution.

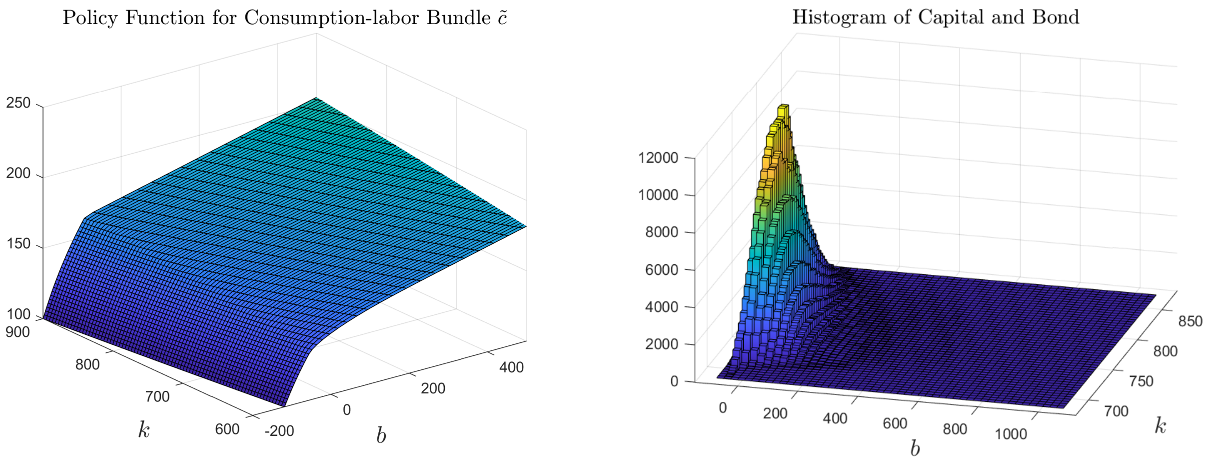


Figure 2: State Transformation and the Ergodic Distribution

Note: The figure is generated using the baseline parameters in Mendoza (2010). The model is solved with 80 grid points over the endogenous state k and 80 grid points over the endogenous state $\tilde{c}$. The policy function corresponds to the first of the 8 realizations of exogenous shocks, i.e., all of $\varepsilon^A, \varepsilon^R, \varepsilon^p$ take their lower values of the two realizations, respectively. The ergodic set is based on a simulated panel of 24 time series, each with 50,000 periods.

that appear in the model's ergodic set, as suggested by the right panel of Figure 2 that the economy spends its time in high levels of debt with high frequencies. The above discussions echo the challenge faced by researchers in choosing the state space of bonds without knowing the actual equilibrium properties, and highlight the usefulness of the state transformation strategy which is enabled by the consistency equation method in the current framework.

Figure 3 plots the policy functions for capital price and the multiplier associated with the collateral constraint. As shown, the collateral constraint tends to bind when the debt level is high (bond is more negative) and capital is low; capital price drops sharply in debt level when the collateral constraint binds, while it drops only modestly when the collateral constraint does not bind. These nonlinear regions appear in the model's ergodic set as shown in Figure 2, highlighting the importance of solving the current model with global methods.

4.1.3 Mapping into the General Setup

For the model in Mendoza (2010) described above, the correspondence with our general setup of the toolbox is $z = (\varepsilon^A, \varepsilon^R, \varepsilon^p)$, $s = (k, \tilde{c})$, and $x = (\mu, w, L, v, b', k')$.

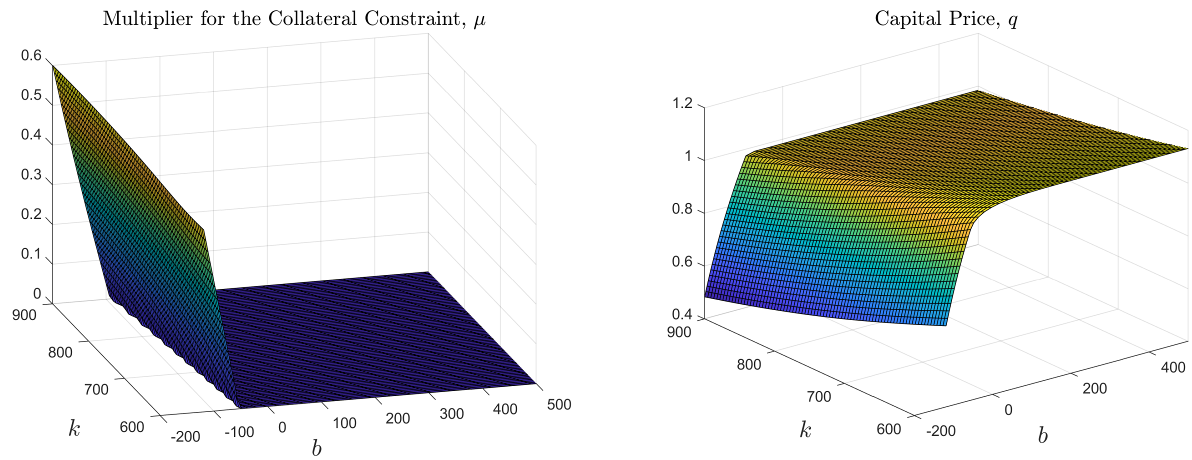


Figure 3: Policy Functions

Note: The figure is generated using the baseline parameters in Mendoza (2010). The model is solved with 80 grid points over the endogenous state k and 80 grid points over the endogenous state $\tilde{c}$. The policy functions are then generated by projecting the variables over the natural state space (k, b). The policy functions correspond to the first of the 8 realizations of exogenous shocks, i.e., all of $\varepsilon^A, \varepsilon^R, \varepsilon^p$ take their lower values of the two realizations, respectively.

4.2 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 methods developed in Fernández-Villaverde and Levintal (2018). Our toolbox's algorithm is a purely projection method. It uses wealth shares as endogenous state variables. As Barro et al. (2017) discuss in their paper, their solution method is not sufficiently accurate for large values of risk-aversion coefficients.[23] We show below that our method can tackle these cases effectively and uncover new economic insights in these cases. For the more risk-averse agent's risk-aversion coefficient of 50, our toolbox solves the model within a minute on a regular laptop (about 40 seconds on a MacBook Air 2.2 GHz Intel Core i7).

4.2.1 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., agents 1 are less risk-averse than agents 2. The other parameters between these

[23]See Table 2 in their paper.

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

$$U_{i,t} = \left\{ \frac{\rho + v}{1+\rho} C_{i,t}^{1-\theta} + \frac{1-v}{1+\rho} \left[\mathbb{E}_t \left(U_{i,t+1}^{1-\gamma_i} \right) \right]^{\frac{1-\theta}{1-\gamma_i}} \right\}^{1/(1-\theta)}. \tag{6}$$

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; P_t denotes its price. 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 (6) as follows:

$$u_{it}^{1-\theta} = \frac{\rho + v}{1+\rho} c_{i,t}^{1-\theta} + \frac{1-v}{1+\rho} (1 - c_{it})^{1-\theta} \left(\mathbb{E}_t \left[\left(R_{i,t+1} u_{it+1} \right)^{1-\gamma_i} \right] \right)^{\frac{1-\theta}{1-\gamma_i}}, \tag{7}$$

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

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

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

$$\left(\rho+v\right)c_{i,t}^{-\theta}=\left(1-v\right)\left(1-c_{it}\right)^{-\theta}\left[\mathbb{E}_{t}\left(R_{i,t+1}u_{it+1}\right)^{1-\gamma_{i}}\right]^{\frac{1-\theta}{1-\gamma_{i}}},\tag{8}$$

and

$$\mathbb{E}_{t}\left[\frac{\left(R_{t+1}^{e}-R_{t+1}^{f}\right)u_{it+1}}{\left(R_{i,t+1}u_{it+1}\right)^{\gamma_{i}}}\right]=0.\tag{9}$$

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

$$\begin{aligned}K_{it}&=x_{it}\left(1-c_{it}\right)\left(1+p_{t}\right)/p_{t}\omega_{it},\\ b_{it}&=\left(1-x_{it}\right)\left(1-c_{it}\right)\left(1+p_{t}\right)\omega_{it}.\end{aligned}$$

In equilibrium, prices are determined such that markets clear:

$$C_{1t}+C_{2t}=Y_{t},\tag{10}$$

$$K_{1t}+K_{2t}=1,\tag{11}$$

$$B_{1t}+B_{2t}=0.\tag{12}$$

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

$$\omega_{it}=K_{it-1}-v\left(K_{it-1}-\mu_{i}\right)+\frac{\left(1-v\right)R_{t}^{f}b_{it-1}}{\left(1+p_{t}\right)\left(1+g_{t}\right)}.\tag{13}$$

We see that given the market clearing conditions (11) and (12),

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

4.2.2 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 (8) is replaced by

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

Using this relationship for $i = 1, 2$, and the market clearing conditions (10), (11) and (12), we obtain

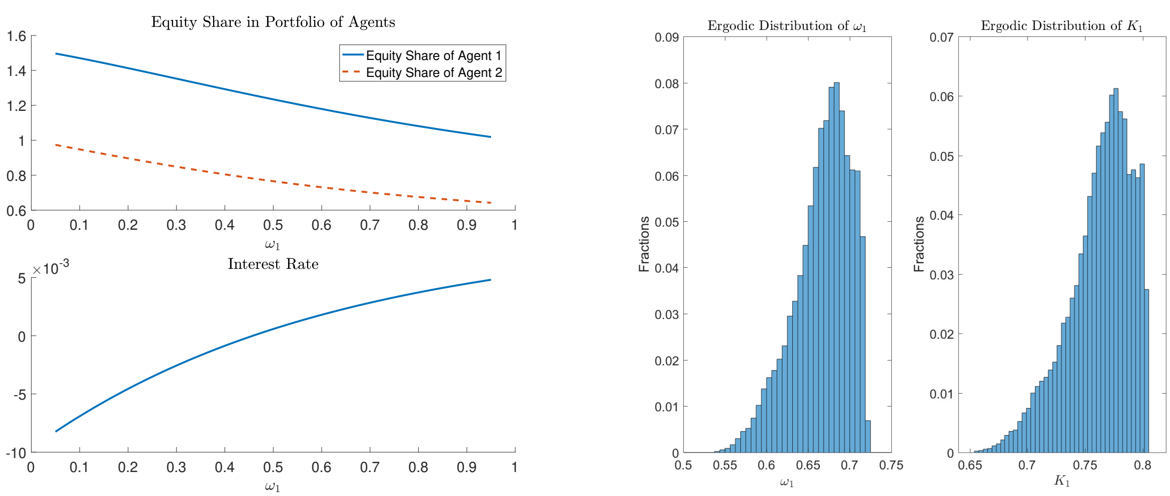
$$p_t = \frac{1 - v}{\rho + v}.$$

The utility function (7) is replaced by

$$\ln u_{it} = \frac{\rho + v}{1 + \rho} \ln c_{it} + \frac{1 - v}{1 + \rho} \ln (1 - c_{it}) + \frac{1 - v}{1 + \rho} \frac{1}{1 - \gamma_i} \ln \left[\mathbb{E}_t \left(R_{i,t+1} u_{it+1} \right)^{1 - \gamma_i} \right]. \tag{14}$$

To solve the model, we use the endogenous state variable ω_{1t} - wealth share of agents 1. This is a state variable with implicit law of motion given by (13), which correspond to equations (3) in the general framework. In each iteration and collocation point, the system of equations to be solved consists of five unknowns – $\{x_{1t}, x_{2t}, R_t^f, \omega_{it+1}(z_{t+1})\}$ (two values for ω_{t+1} for two future realizations) – and five equations (9) for $i = 1, 2$, the market clearing condition for bond (12) and the consistency equations (13).

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 4.



(a) Policy Functions

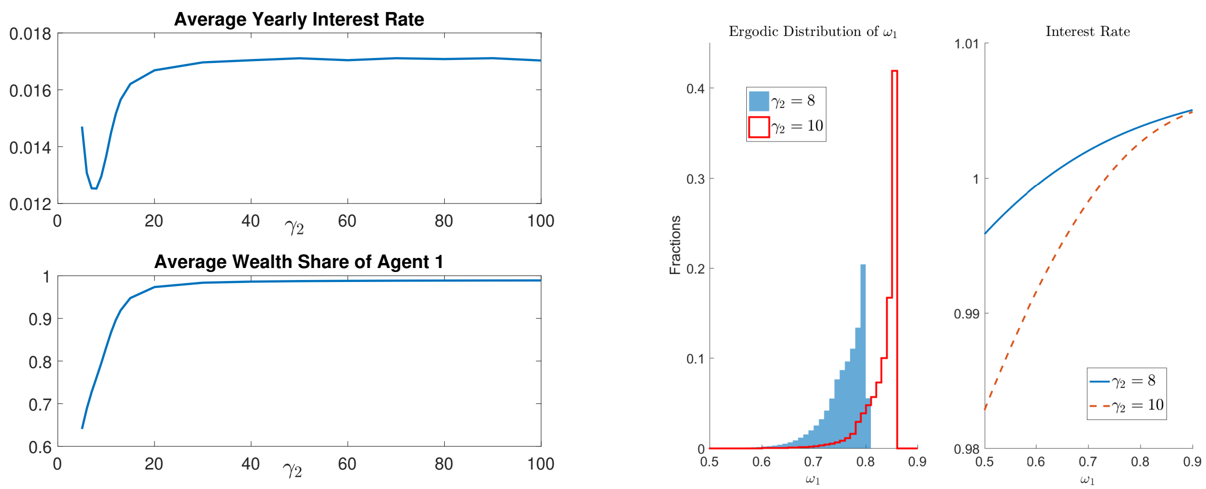
(b) Ergodic Distributions

Figure 4: 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 $\gamma_2 = 5.6$. The growth rate in normal times is 0.025. Rare disasters happen with probability 4%, and once a rare disaster happens, productivity drops by 32%. The model period is one quarter.

In Table 2 of Barro et al. (2017), the values for risk aversion parameters γ_1 and γ_2 are calibrated 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 $\bar{R}^f = 1.01$. Table 2 of their paper presents 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. For smaller values of γ_2 our results coincide with the results in Table 2 of Barro et al. (2017). But for larger values our results differ from theirs. In Figure 5(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 $\gamma_1 = 3.1$ since $\bar{R}^f$ is increasing in γ_2 when γ_2 is greater than 8.



(a) Interest Rate and Wealth Share of Agent 1

(b) Comparing Cases with Different γ_2

Figure 5: 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 as we vary γ_2. In Figure (b), we compare the policy functions of R^f and ergodic distributions when $\gamma_2 = 8$ and 10.

The non-monotonicity is driven by two opposing forces. First, as γ_2 becomes larger, agents 2 become more risk-averse, and demand more of the safe asset (bond). This higher demand 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 to equity is significantly higher than bond's return, the wealth share of agents 1, ω_1 tends to be 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 5 shows that when γ_2 is below 8, the first force dominates and $\bar{R}^f$ is decreasing in γ_2 as implicitly assumed in Barro et al. (2017).

However, when γ_2 is greater than 8, the second force dominates and $\bar{R}^f$ is increasing in γ_2. When γ_2 is greater than 20, $\bar{R}^f$is not responsive to γ_2 since the distribution of ω_1 becomes almost degenerated at its upper limit. See Figure 5(b) for a comparison of two cases: $\gamma_2 = 8$ versus $\gamma_2 = 10$.

4.2.3 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 $z = (g)$, $s = (\omega_1)$, and $x = \left(c_1, c_2, x_1, x_2, R^f, K_1, b_1, p\right)$.

4.3 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 a high equity risk premium, a low and relatively smooth interest rate, and countercyclical movements in the equity 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 very 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 new laws of motion and new pricing functions. These functions are again 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 *globally* 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. With this algorithm, our toolbox solves the baseline model in Guvenen (2009) in less 3 minutes on a regular laptop (about 2.9 minutes on a MacBook Air 2.2 GHz Intel Core i7) while it takes about 9 hours 15 minutes (on a MacProWorkstation with 8-core 3.2 GHz Xeon processors) using the original algorithm in Guvenen (2009) as reported in Appendix A in his paper.

4.3.1 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

$$U_{i,t} = \left\{ (1-\beta)\, c_{i,t}^{1-\rho^i} + \beta \left[\mathbb{E}_t \left(U_{i,t+1}^{1-\alpha} \right) \right]^{\frac{1-\rho^i}{1-\alpha}} \right\}^{1/\left(1-\rho^i\right)}. \tag{15}$$

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 the borrowing constraint is $b_{h,t+1} \geq -\underline{B}$. In the calibration $\underline{B}$ is set at six times of the average monthly wage rate. Non-stockholders are subject to 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:

$$P_t^s = \max_{\left\{I_{t+j}, L_{t+j}\right\}} \mathbb{E}_t \left[\sum_{j=1}^{\infty} \beta^j \frac{\Lambda_{h,t+j}}{\Lambda_{h,t}} D_{t+j} \right]. \tag{16}$$

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

$$K_{t+1} = (1-\delta) K_t + \Phi\left(\frac{I_t}{K_t}\right) K_t, \tag{17}$$

with the functional form for Φ from Jermann (1998). Each period, the firm sells one-period bonds at price P_t^f. The bond supply is constant and equals to a fraction χ 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 (17), $\{L_t, I_t\}$ are optimal for the representative firm; and (iii) all markets clear:

$$\mu b_{h,t+1} + (1-\mu) b_{n,t+1} = \chi \bar{K}, \tag{18}$$

$$\mu s_{t+1} = 1, \tag{19}$$

$$L_t = 1,$$

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

4.3.2 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 the total bond holding by the non-stockholders. The optimization problems of the households and the representative firm are globally 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. Complementary-slackness conditions:

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

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}:

$$q_t = \beta \mathbb{E}_t \left\{ \frac{\Lambda_{h,t+1}}{\Lambda_{h,t}} \left[\theta Z_t K_t^{\theta-1} - \frac{I_{t+1}}{K_{t+1}} + q_{t+1} \left(1 - \delta + \Phi \left(\frac{I_{t+1}}{K_{t+1}} \right) \right) \right] \right\}, \tag{20}$$

in which capital price q_t is the Lagrangian multiplier on the capital formation (17) and satisfies

$$q_t \Phi' \left(\frac{I_t}{K_t} \right) = 1. \tag{21}$$

The auxiliary variables can be determined by the utility function (15), market clearing conditions, (17) and the following two equations:

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

and

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

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, $c_{h,t+1}$,

$c_{n,t+1}$, $P^s_{t+1} + D_{t+1}$, I_{t+1}/K_{t+1}, $U_{h,t+1}$, and $U_{n,t+1}$ are functions of $\{K_{t+1}, B^n_{t+1}, Z_{t+1}\}$ and are solved from the previous iteration. Similar to Guvenen (2009), the initial guesses for these functions are obtained by solving a version of the model with no leverage ($\chi = 0$, $\underline{B} = 0$).[24]

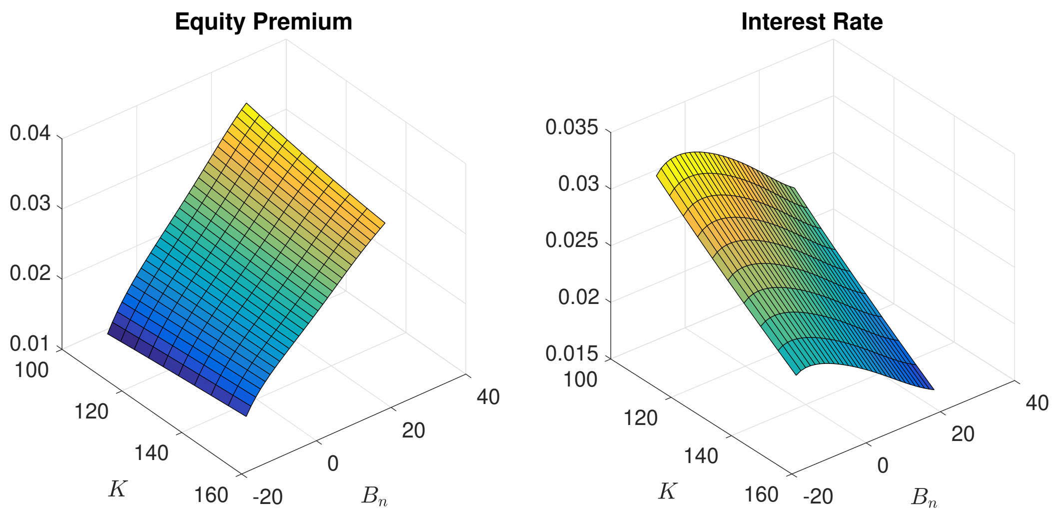


Figure 6: Asset Pricing Policy Functions in Guvenen (2009)

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 6, we plot the annual equity premium and interest rate as functions of $\{K, B^n\}$ by fixing $Z_t = 1$. The equity premium is increasing in B_n and interest rate is decreasing in B_n. This is because stockholders, having higher IES and tolerating more non-smooth consumption, insure non-stockholders against aggregate shocks. As B_n increases, stockholders become less wealthy, relative to non-stockholders, so they demand a higher premium to hold equity and to bear more volatility in consumption. At the same time, non-stockholders' aggregate precautionary saving increases and it pushes down interest rate. Figure 7 plots the ergodic distributions of capital and the financial wealth share of stockholders.

[24] It is relatively simple 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 guesses for the full model. The toolbox allows users to override parameters and to provide their own initial guess functions using the "WarmUp" option. Thus they do not need to write separate codes for different model versions. See the codes on the toolbox's website for detail. Furthermore, the functions provided can be defined on different grid points from the state variables, which offers users much flexibility. For example, a user can solve a model with coarse grids for speed first and then use its converged policy functions as the initial guess for the same model with finer grids.

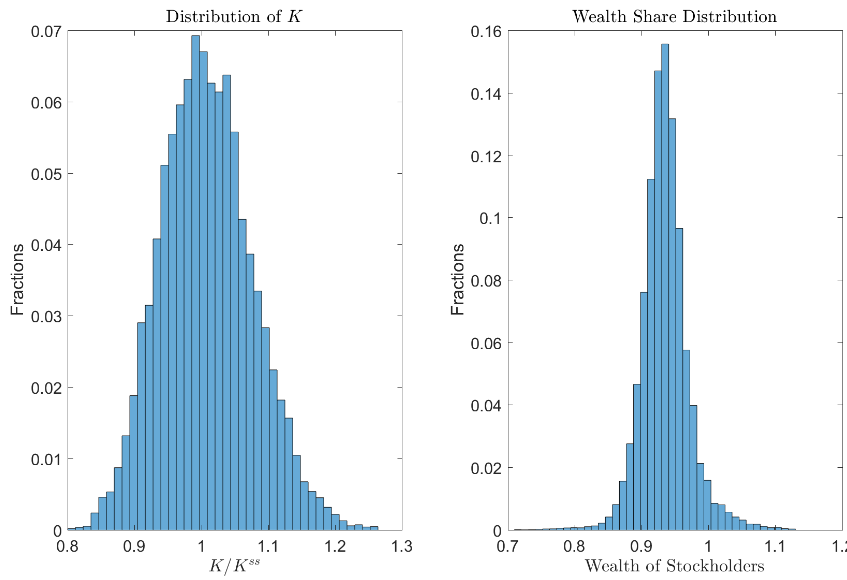


Figure 7: Ergodic Distributions of Capital and Wealth Share
Note: The Ergodic Distributions are generated by simulation. We use the same parameter values as in Table 1 of Guvenen (2009).

4.3.3 Mapping into the General Setup

For the model in Guvenen (2009) described above, the correspondence with our general setup of the toolbox is $z = (Z)$, $s = (K, B^n)$, and $x = (c_h, c_n, I, B^{n\prime}, \lambda_h, \lambda_n, P^s, P^f, q, U_h, U_n)$.

4.4 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 can be robustly and efficiently solved by the toolbox, using the exact model and calibration 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 using the same system of equations. This seemingly minor detail is important in allowing the model to be written and solved in the same general framework as other models. We also illustrate the use of the adaptive grid method that automatically puts more points in regions of the state space with more nonlinearity because of binding borrowing constraint.

4.4.1 Model Description

In a small-open economy, the representative consumer derives utility from the consumption of tradable goods c_t^T and of non-tradable goods c_t^N according to

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

with the composite consumption

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

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 borrowing is subject to a borrowing constraint tied to the non-tradable good price as follows:

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

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

A sequential competitive equilibrium corresponds to stochastic sequences

$$\{b_{t+1}, c_t^T, c_t^N, c_t, p_t^N\}_{t=0}^{\infty}$$

such that $\{b_{t+1}, c_t^T, c_t^N\}$ solve 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 globally 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:

$$\begin{aligned} & p_t^N = \Big(\frac{1-\omega}{\omega}\Big)\Big(\frac{c_t^T}{c_t^N}\Big)^{\eta+1}, \\ & \lambda_t = \beta(1+r)\mathbb{E}_t \lambda_{t+1} + \mu_t, \\ & \mu_t\big[b_{t+1} + (\kappa^N p_t^N y_t^N + \kappa^T y_t^T)\big] = 0, \mu_t \geq 0, \\ & b_{t+1} + c_t^T + p_t^N c_t^N = b_t(1+r) + y_t^T + p_t^N y_t^N, \end{aligned}$$

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.

4.4.2 Computation

For computation, we use the exact parameters as in the benchmark calibration in Bianchi (2011). 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) into 16 states. The natural endogenous state variable of the economy is b_t.

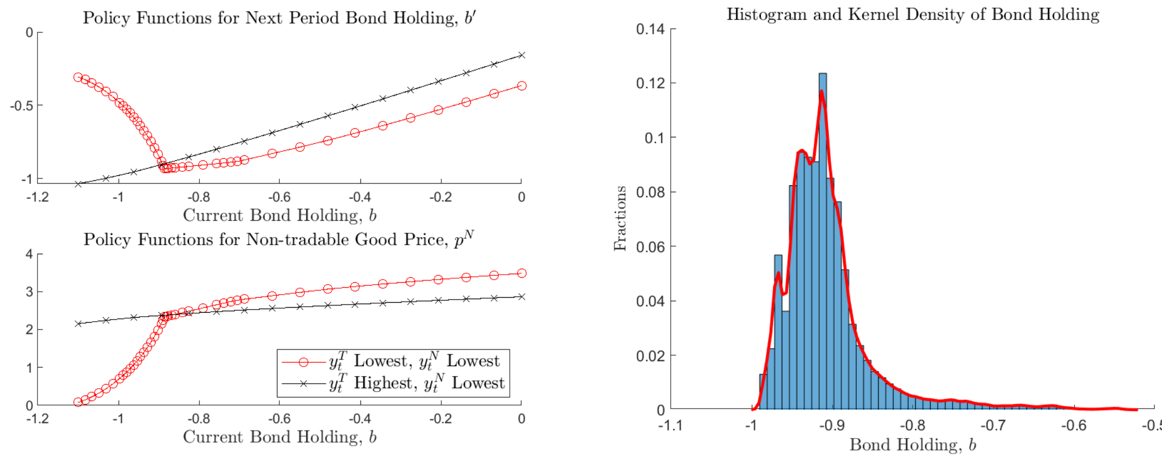
Like previous examples, a time step of policy iterations is to solve the equilibrium system defined above, for each collocation point of exogenous and endogenous states, taking the state transition function implicitly defined in $\lambda_{t+1}(y_{t+1}^N, y_{t+1}^T, b_{t+1})$ as given. After each time 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 the adaptive grid method provided by the toolbox, which automatically places more points in the region of the state space that features a high degree of non-linearity.[25] The equilibrium policy functions for p_t^N and b_{t+1}, and the ergodic distribution of b_t are presented in Figure 8.

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, showing that non-linear regions do exist in the equilibrium ergodic set 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 highly nonlinear. Importantly, the method accommodates the possibility that these highly nonlinear regions differ across exogenous states, as shown in the figure. This result illustrates the effectiveness of the adaptive grid method for this class of models, as these highly nonlinear regions of state

[25] As described in the user manual on the toolbox's website, users only need 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 it thus can accommodate models with a high-dimensional state space.



(a) Policy Functions

(b) The Ergodic Distribution of Bond

Figure 8: 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.

space cannot be determined ex-ante. Without the method, it would require very dense exogenous grids or painstaking manual configurations.

4.4.3 Mapping into the General Setup

For the model in Bianchi (2011) described above, the correspondence with our general setup of the toolbox is $z = (y^T, y^N)$, $s = (b)$, and $x = (b', c^T, c^N, c, \mu, \lambda, p^N)$.

4.5 Macroeconomic Implications of COVID-19 by Guerrieri et al (2020)

In this timely and important contribution, Guerrieri et al. (2020) analyze the effects of large supply shocks such as shutdowns, layoffs, and firm exits due to COVID-19. The authors show that in a two-sector model, these supply shocks can trigger changes in aggregate demand larger than the shocks themselves, leading to a decrease in the equilibrium interest rate. This is the case when the elasticity of substitution across sectors is not too large and the inter-temporal elasticity of substitution is sufficiently high.

Their model is deterministic and the supply shock is a one time, unexpected shock. They also assume maximally tight borrowing constraint. We extend their model to allow for stochastic, recurrent shocks and more relaxed borrowing constraint. This extension can be solved relatively easily using our toolbox. This extension shows that because these supply shocks are rare, the precautionary saving motive does not significantly change the results in the original model with a one time unexpected shock. The recurrent shocks lead to a very persistent reduction in the wealth of workers more directly affected by the shocks.

4.5.1 The Model

We follow closely the notation in Guerrieri et al. (2020). The total population is normalized to one, with a fraction ϕ of agents working in sector 1 and the remaining fraction $1-\phi$ of agents working in section 2. We assume that workers are perfectly specialized in their sector. Sector 1 is the contact-intensive sector that is directly affected by the supply shock.

The labor endowment of workers in sector 2 is constant and is set to $\bar{n}$, while the labor endowment of workers in sector 1 follows a two-point Markov process with state in $\{1,2\}$, where 1 corresponds to normal times and 2 corresponds to pandemics. During normal times, their labor endowment is $n_{1t} = \bar{n}$, while when a supply shock hits, their labor endowment drops to $n_{1t} = \delta\bar{n}$ with $\delta < 1$. In the COVID-19 example, sector 1 is contact-intensive, and a fraction δ of its production is shut down when the pandemic hits. On the other hand, sector 2 is unaffected. The transition matrix between these two states is

$$\begin{bmatrix} \pi_1 & 1-\pi_1 \\ 1-\pi_2 & \pi_2 \end{bmatrix},$$

in which $1-\pi_1$ is a small probability for the economy to enter the supply-driven crisis, and π_2 is the probability for the crisis to last for one more period.

The production technology is linear in both sectors: $Y_{jt} = N_{jt}$ for $j = 1,2$. Competitive firms in each sector j hire workers at wage W_{jt} and sell their products at price P_{jt}. Prices are flexible, and given the market structure we have $P_{jt} = W_{jt}$. The consumer's utility function is

$$\mathbb{E}_0\left[\sum_{t=0}^{\infty}\beta^t\frac{C_t^{1-\sigma}}{1-\sigma}\right], \tag{22}$$

in which $C_t = \left(\phi^\rho c_{1t}^{1-\rho} + (1-\phi)^\rho\, c_{2t}^{1-\rho}\right)^{\frac{1}{1-\rho}}$ features constant elasticity of substitution $1/\rho$ between the two goods and constant intertemporal elasticity of substitution $1/\sigma$.

As in Guerrieri et al. (2020), here we set good 2 to be the numeraire, i.e., $P_{2t} \equiv 1$.

Workers in sector j maximize (22) subject to

$$P_{1t}c_{1t}^j + c_{2t}^j + \frac{a_{t+1}^j}{1+r_t} \leq W_{jt}n_t^j + a_t^j, \tag{23}$$

where they allocate their labor income and bond holding from the previous period, a_t^j among consumption goods produced in the two sectors and bond holding into the next period. Interest rate r_t is determined competitively.

In addition, we assume that the workers are subject to the following borrowing constraint: $a_{t+1}^j \geq -\bar{A}$.

Denote sector j workers' Lagrangian multiplier for the budget constraint (23) by $\beta^t \lambda_t^j$ and the multiplier for the borrowing constraint by $\beta^t \mu_t^j$. The first-order conditions for the workers' optimal decision are:

$$\lambda_t^j = \left(C_t^j\right)^{\rho-\sigma} (1-\phi)^\rho \left(c_{2t}^j\right)^{-\rho},$$

$$P_{1t} = \left(\frac{c_{1t}^j/\phi}{c_{2t}^j/(1-\phi)}\right)^{-\rho}, \tag{24}$$

$$-\frac{\lambda_t^j}{1+r_t} + \mu_t^j + \beta \mathbb{E}_t \left(\lambda_{t+1}^j\right) = 0, \tag{25}$$

$$\mu_t^j \left(a_{t+1}^j + \bar{A}\right) = 0. \tag{26}$$

We also have the market clearing conditions for bond and consumption good 2:

$$\phi a_{t+1}^1 + (1-\phi) a_{t+1}^2 = 0,$$
$$\phi c_{2t}^1 + (1-\phi) c_{2t}^2 = (1-\phi)\bar{n}.$$

The market clearing condition for consumption good 1 is implied by Walras' law.

We use a_t^1 as the endogenous state variable and look for a recursive equilibrium as a mapping from a_t^1 to the allocation and prices that satisfy the first-order conditions and market clearing conditions above.

Notice that by the pricing equation (24), $\frac{c_{1t}^1}{c_{1t}^2} = \frac{c_{2t}^1}{c_{2t}^2}$, which means that the consumption shares of workers in sector 1 are the same between these two consumption goods. Denote the consumption share of workers in sector 1 as $\tilde{c}_{1t}$; then

$$c_{1t}^1 = \tilde{c}_{1t} n_{1t}, \qquad c_{1t}^2 = (1-\tilde{c}_{1t})\phi n_{1t}/(1-\phi),$$
$$c_{2t}^1 = \tilde{c}_{1t}(1-\phi)\bar{n}/\phi, \qquad c_{2t}^2 = (1-\tilde{c}_{1t})\bar{n},$$

which lead to

$$C_t^1 = \frac{\tilde{c}_{1t}}{\phi} Y_t, \text{ and } C_t^2 = \frac{1 - \tilde{c}_{1t}}{1 - \phi} Y_t,$$

where $Y_t = \left[\phi n_{1t}^{1-\rho} + (1 - \phi) \bar{n}^{1-\rho} \right]^{\frac{1}{1-\rho}}$, and

$$\lambda_t^1 = \left(\frac{\tilde{c}_{1t}}{\phi} Y_t \right)^{-\sigma} \left(\frac{Y_t}{\bar{n}} \right)^{\rho}, \text{ and } \lambda_t^2 = \left(\frac{1 - \tilde{c}_{1t}}{1 - \phi} Y_t \right)^{-\sigma} \left(\frac{Y_t}{\bar{n}} \right)^{\rho}.$$

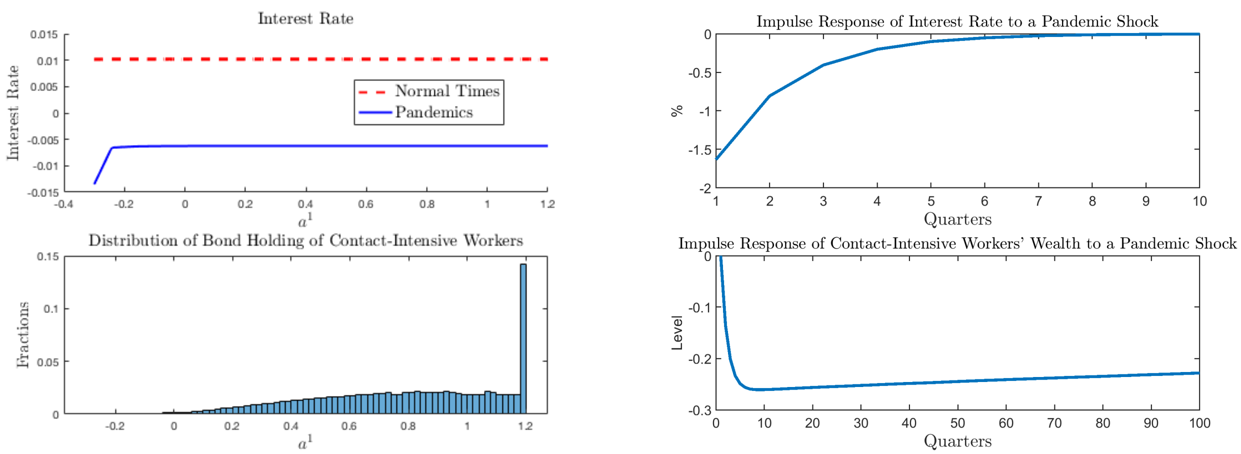
Overall, for each a_t^1 (and the exogenous state of the economy), the minimal equilibrium system can be represented by five unknowns, namely, $\left\{ \tilde{c}_{1t}, a_{t+1}^1, \mu_t^1, \mu_t^2, r_{t+1} \right\}$, and can be solved by a system of five equations, namely, the budget of workers in sector 1, equation (23), the FOC in equation (25), and the complementary-slackness condition in equation (26) for $j = 1, 2$.

4.5.2 Calibration and Results

We use quarters for model periods and standard parameters in the literature. For preferences, we use $\beta = 0.99$ as the quarterly discount factor. The inverse inter-temporal elasticity of substitution is set at $\sigma = 0.5$ (strictly less than 1, as required by the analytical results in Guerrieri et al. (2020) for supply shocks to trigger even larger aggregate demand responses). We vary the inverse intra-temporal elasticity of substitution ρ between 0.1 and 0.9.

For labor market parameters, we normalize $\bar{n}$ at 1. The share of the contact-intensive sector ϕ is set to 0.2. We assume that when the pandemic shocks hit, labor supply in the contact-intensive sector declines by 50% (roughly consistent with the increase in unemployment claims in the U.S. during the pandemics). We assume that the pandemics last for 2 quarters on average, equivalently, $\pi_2 = 0.5$. π_1 is chosen so that the economy stays in pandemics in approximately 0.5% of the times (consistent with the historical frequency reported in Jordà et al. (2020)). Borrowing limit $\bar{A}$ is set at 30% of the wage in normal times, a standard value in the literature.

For the benchmark results, we use $\rho = 0.75 > \sigma = 0.5$. The upper panel in Figure 9(a) shows the interest rate as a function of the endogenous state variable a_t^1 in normal times ($z = 1$) and during pandemics ($z = 2$). Interest rate is lower during pandemics, which reflects the result in Guerrieri et al. (2020) that the aggregate demand response outweighs the supply shock. In addition, the figure also shows that the effect is stronger when the net worth of workers in the contact-sensitive sector is low. The lower panel plots the ergodic distribution of bond holding of workers in sector 1. The possibility of pandemics leads to these workers' precautionary saving, sometimes up to the borrowing limit of workers in sector 2. However, the precautionary saving does not undo the results in Guerrieri et al. (2020). Because this is a dynamic extension of the model, we can study the dynamic responses of the economy to pandemic shocks. Figure 9(b) shows the impulse responses of interest rate and the wealth of sector 1 workers to a pandemic shock. While interest rate reverses relatively quickly to pre-pandemic level after the shock, workers in sector 1 suffers from a persistent, long-lasting wealth lost.



(a) Interest Rate Policy Function and the Ergodic Distribution

(b) Impulse Response Functions

Figure 9: Policy Function, Ergodic Distribution, and IRFs

Note: We use $\rho = 0.75 > \sigma = 0.5$ and other parameters described in the main text

To further investigate the robustness of the results in Guerrieri et al. (2020), Figure 10 plots the average interest rate before and after the pandemic shocks hit the economy as we vary ρ. The figure shows that when $\rho > \sigma$ (more precisely $1/\rho < 1/\sigma$), interest rate drops when the pandemic shock hits, while it rises when $\rho < \sigma$ ($1/\rho > 1/\sigma$). This is exactly the result emphasized in Guerrieri et al. (2020).

4.5.3 Mapping into the General Setup

For the extension of the model in Guerrieri et al. (2020) described above, the correspondence with our general setup of the toolbox is $z = (n_1)$, and $s = (a^1)$, and $x = \left(\tilde{c}_1, \mu^1, \mu^2, r\right)$.

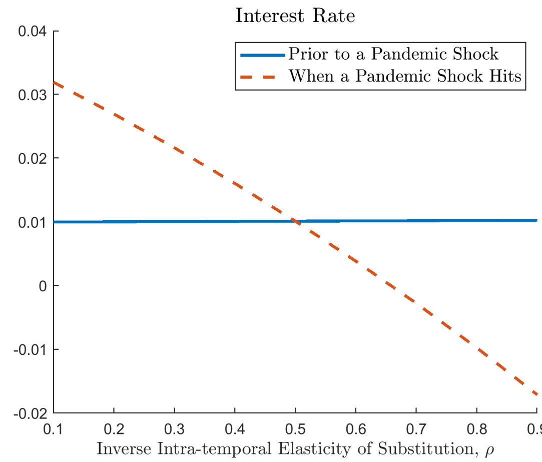


Figure 10: Interest Rate before and after Pandemics

Note: We use $\sigma = 0.5$ and other parameters described in the main text. The dashed curve corresponds to the interest rate when the shock switches from normal to pandemic, averaged in the model's ergodic set. The solid curve corresponds to the average interest rate prior to the period in which the pandemic shock hits.

5 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 number 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. In contrast, in fully heterogeneous agent incomplete markets 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. This point is emphasized in Cao (2020).

However, Cao (2020) shows that incomplete markets models with finite agent types are useful special cases of the fully heterogeneous agent incomplete markets model in 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 on the toolbox's website. The dynamics of the aggregate variables in the two models are similar. This comparison suggests that, in general, the solution of the finite-agent models can be useful in understanding the properties of the fully heterogeneous 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 by observing that given a conjectured law of motion for the aggregate capital, the households' Euler equation, together with the complementary-slackness condition, is necessary and sufficient for optimality:

$$u'(c_t) = \mathbb{E}\left[u'(c_{t+1})(1-\delta+r_{t+1})\right] + \lambda_t,$$

where λ_t is the Lagrangian multiplier on households' borrowing constraint and r_{t+1} is the rental rate of capital at $t+1$. The toolbox can also be used to simulate the implied dynamics of the cross-sectional wealth distribution and aggregate capital. Then, with an additional fixed-point iteration on their laws of motion, which can be coded simply in MATLAB, the toolbox solution can be used to solve for a recursive equilibrium. This idea can be used to solve Krusell and Smith's baseline model in less than 100 lines of our toolbox code and 100 lines of MATLAB code. We also provide these codes, as well as the implementation for Krusell and Smith's model with heterogeneous discount factors, on the toolbox's website.

Similarly, we can use the toolbox to compute stationary recursive equilibrium in heterogeneous agent models without aggregate shocks such as Huggett (1993) and Aiyagari (1994), and transitional path equilibrium in Huggett (1997). The codes for these models are also available on the toolbox's website.

6 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 toolbox uses a policy function iteration methods and hence can be used to solve for stochastic transition paths as in Storesletten et al. (2019). Lastly, the toolbox generates Monte-Carlo simulations from model solutions. These simulations can be used for model estimation with the Generalized Method of Moments or Bayesian Estimation Methods. This is the natural next step which we leave for future developments of the toolbox.

References

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

Barro, R. J. (2006). 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.

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

Blanchard, O. J. and C. M. Kahn (1980). The solution of linear difference models under rational expectations. *Econometrica 48*(5), 1305–1311.

Bocola, L. (2016). The pass-through of sovereign risk. *Journal of Political Economy 124*(4), 879–926.

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.

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). Uncovering the effects of the zero lower bound with an endogenous financial wedge. Technical report. SSRN 3531341.

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

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

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.

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

Elenev, V., T. Landvoigt, and S. Van Nieuwerburgh (2021). A macroeconomic model with financially constrained producers and intermediaries. *Econometrica 89*(3), 1361–1418.

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.

Greenwood, J., Z. Hercowitz, and G. W. Huffman (1988). Investment, Capacity Utilization, and the Real Business Cycle. *The American Economic Review 78*(3), 402–417.

Guerrieri, L. and M. Iacoviello (2015). Occbin: A toolkit for solving dynamic models with occasionally binding constraints easily. *Journal of Monetary Economics 70*, 22 – 38.

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

Guerrieri, V., G. Lorenzoni, L. Straub, and I. Werning (2020). Macroeconomic implications of covid-19: Can negative supply shocks cause demand shortages? Technical report, National Bureau of Economic Research.

Gust, C., E. Herbst, D. López-Salido, and M. E. Smith (2017). 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.

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.

Heinkenschloss, M., M. Ulbrich, and S. Ulbrich (1999). Superlinear and quadratic convergence of affine-scaling interior-point Newton methods for problems with simple bounds without strict complementarity assumption. *Mathematical Programming 86*(3), 615–635. Publisher: Springer.

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

Huggett, M. (1997). The one-sector growth model with idiosyncratic shocks: Steady states and dynamics. *Journal of Monetary Economics 39*(3), 385 – 403.

Jermann, U. J. (1998). Asset pricing in production economies. *Journal of Monetary Economics 41*(2), 257–275.

Jordà, O., S. R. Singh, and A. M. Taylor (2020). Longer-run economic consequences of pandemics. Working Paper 26934, National Bureau of Economic Research.

Judd, K. L. (1992). Projection methods for solving aggregate growth models. *Journal of Economic Theory 58*(2), 410 – 452.

Judd, K. L., L. Maliar, and S. Maliar (2011). Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models. *Quantitative Economics 2*(2), 173–210.

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.

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

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

Mendoza, E. G. and S. Villalvazo (2020). Fipit: A simple, fast global method for solving models with two endogenous states and occasionally binding constraints. *Review of Economic Dynamics 37*, 81 – 102.

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.*

Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control 33*(3), 649–665.

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

Sims, C. A. (2002). Solving linear rational expectations models. *Computational Economics 20*(1), 1–20.

Smets, F. and R. Wouters (2007, June). Shocks and frictions in us business cycles: A bayesian dsge approach. *American Economic Review 97*(3), 586–606.

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

Uhlig, H. (1999). A toolkit for analysing nonlinear dynamic stochastic models easily. In R. Marimon and A. Scott (Eds.), *Computational Methods for the Study of Dynamic Economies*, pp. 30–61. Oxford University Press.

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.

Online Appendix

A The Design of the GDSGE Toolbox

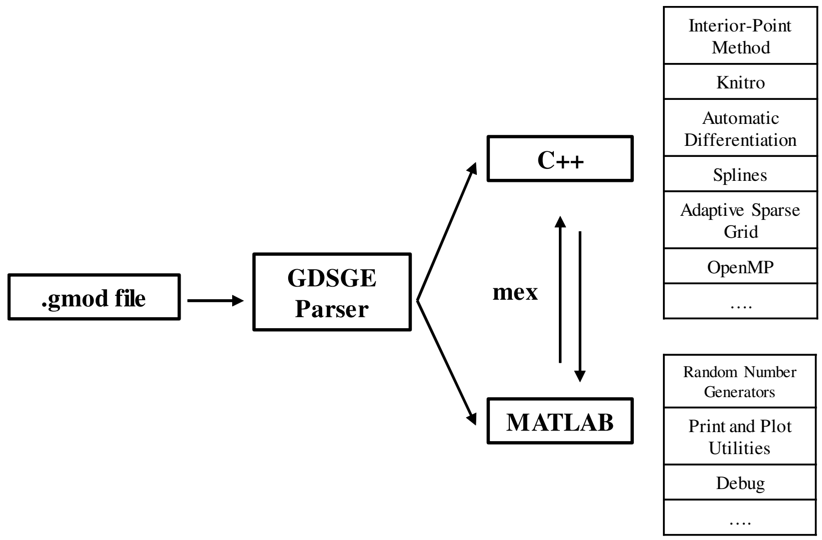


Figure 11: Toolbox Design and Implementations

In this section, we described in detail how the toolbox is designed and implemented. The design of the toolbox is depicted in Figure 11. 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 A.1. The gmod files can be uploaded to the toolbox's website, and the toolbox compiles the files into MATLAB script files and C++ dynamic libraries that 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 approximating policy functions, are described in Subsection A.2

The MATLAB script files and C++ dynamic libraries should run locally on users' computers. After they finish running, they return the policy and state transition functions from converged time iterations and the Monte Carlo simulation samples.

A.1 User Inputs: the gmod Files

The toolbox asks users to provide gmod files that contain the equilibrium system (1) of their models. In this subsection, we provide the description for a minimal gmod file such as that for the leading example in Section 2, and refer readers to the toolbox's website for a detailed user manual. A minimal 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.[26]

shock_num. The number of discretized points for *var_shock*. For multidimensional *var_shock*, this should be the size of the Cartesian products of the discretized sets for 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.[27]

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 $x'(z')$. In the policy iteration at step n, $x'(z')$ can be evaluated by interpolating the policy function solved in the last iteration, i.e., $\mathcal{P}^{(n-1)}(z', s')$.[28] Even though the general

[26]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.

[27]For function approximations using fixed-grid discretization such as splines, the grids will be used by default; for the adaptive sparse grid method, the two end points of the grids will be used as the range of the state variables.

[28]As emphasized in Section 3.2, our framework allows for state variables with implicit state transitions, denoted by $\bar{\bar{s}}$ following the notation in the section. This is made possible by including the future values of these state variables, $\bar{\bar{s}}'$, as unknowns and supplying additional *consistency equations* that $\bar{\bar{s}}'$ need to satisfy in equilibrium. Therefore, it only requires a single-step interpolation to evaluate $\mathcal{P}^{(n-1)}(z', (\bar{s}', \bar{\bar{s}}'))$ and obtain future policy variables x' that are necessary in evaluating the current equation system. An alternative approach to deal with implicit state transitions, adopted by Elenev et al. (2021) for example, is to first use the transition function $\mathcal{T}^{(n-1)}(z, z', s)$ solved in the last iteration to forecast $\bar{\bar{s}}' = \mathcal{T}^{(n-1)}_{\bar{\bar{s}}'}(z, z', s)$, and then evaluate $\mathcal{P}^{(n-1)}(z', (\bar{s}', \bar{\bar{s}}'))$ to get future policy variables x'. This alternative approach can be implemented in our toolbox by declaring future values of these state variables as *var_interp*, and interpolate $\mathcal{T}^{(n-1)}$ and $\mathcal{P}^{(n-1)}$ sequentially as described above. This alternative approach, however, does not solve some of our examples. The reason is outlined in the introduction and should be clearer now—this alternative approach directly uses the last-period transition function $\mathcal{T}^{(n-1)}$ to evaluate $\bar{\bar{s}}'$ as a function of current state variables only, and does not allow the current policy variables to affect $\bar{\bar{s}}'$. In the context of the Heaton and Lucas (1996) example, this approach basically does not allow the current choice of bonds and shares to affect the future wealth share, which is slightly at odds with equilibrium properties. However, using dampening,

formulation allows any policy variable in x to appear as a future state, in practice not all of them are necessary. Here, we only include those variables that 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*. 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', or future policies, x', in the notation for the general framework. The model block can use the following built-in functions.

GDSGE_EXPECT. Calculate the conditional expectation of the model objects 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 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 minimizes numerical errors within a time step. We also implement the conventional and fast,

the method can still solve Heaton and Lucas's model with reasonable accuracy. We provide its GDSGE implementation on the toolbox's website.

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 on the toolbox's website 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) for example) to understand the transmission mechanisms, or to estimate the models.

A.2 Implementations

Once a gmod file is processed by the toolbox, it 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 actual computations are implemented in the native C++ code to achieve maximum performance and are contained in the dynamic library, while the MATLAB script file provides a convenient interface to print, debug, and specify options. To reach maximum computation efficiency, the actual calculations including equation solver, interpolation, automatic differentiation, and parallel computation, are implemented in C++. Below we discuss each of them in detail.[29]

Equation Solver The time iteration step requires solving the systems of equations (2) at collocation points 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 projection-based interior-point method to respect the box constraints (Powell, 1970; Coleman and Li, 1996; Bellavia et al, 2012).[30]

We also provide interfaces to commercial optimization software SNOPT and Knitro for users with licenses.[31] The equation solvers are supplemented with randomization

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

[30]It is important that the equation solver can respect box constraints for unknowns, as the toolbox needs to deal with inequality conditions such as complementary slackness conditions where the unknowns are imposed to have a sign restriction.

[31]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

over initial guesses at each collocation point.

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 adapt the automatic differentiation method implemented by Adept (Hogan, 2014). This library utilizes the expression template feature of C++, and hence much of the differentiation is accomplished at compile time, bringing the computational cost on par with evaluating analytical gradients.

Interpolation The time iteration step (2) involves function approximations because $(z', s'(z'))$ might fall outside the set of collocation points, $\mathcal{C}^{(n)}$. The default option is multidimensional linear interpolation or splines. We also implement a multidimensional 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 policy function iteration, the systems of equations (2) 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, thus all problems share a same block of memory for function approximation parameters, minimizing the overhead for data communications. When we evaluate 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.

B Example Toolbox Codes

In this appendix, we provide the gmod files for the models discussed in Section 4. These codes can also be downloaded from the toolbox's website, together with the gmod codes for many other models.

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.

B.1 Mendoza (2010)

```
% Parameters
parameters gamma sigma omega beta alpha eta delta a phi kappa tau;
parameters flow_scale;

beta    = 0.92;     % Discount factor
sigma   = 2;        % CRRA
omega   = 1.846;    % Labor elasticity
gamma   = 0.306;    % Capital share
alpha   = 0.592;    % Labor share
eta     = 0.102;    % Intermediate share
delta   = 0.088;    % Depreciation rate
a       = 2.75;     % Capital adjustment cost
phi     = 0.26;     % Working capital exernal finance share
kappa   = 0.2;      % Tightness of collateral constraint
tau     = 0.17;     % Consumption tax

flow_scale = 1e4;   % Budget normalization factor

% Toolbox options
PrintFreq = 10;
SaveFreq = inf;
SIMU_RESOLVE=0; SIMU_INTERP=1; % Use interp for fast simulation

% Shocks
var_shock A R p;
shock_num = 8;
% Markov process of shocks (from Mendoza AER (2010):
shock_trans =  [0.51363285, 0.16145114, 0.07773228, 0.02443373, 0.16145114, 0.03201937, 0.02443373, 0.00484575;...
                0.03201937, 0.64306463, 0.00484575, 0.09732026, 0.16145114, 0.03201937, 0.02443373, 0.00484575;...
                0.07773228, 0.02443373, 0.51363285, 0.16145114, 0.02443373, 0.00484575, 0.16145114, 0.03201937;...
                0.00484575, 0.09732026, 0.03201937, 0.64306463, 0.02443373, 0.00484575, 0.16145114, 0.03201937;...
                0.03201937, 0.16145114, 0.00484575, 0.02443373, 0.64306463, 0.03201937, 0.09732026, 0.00484575;...
                0.03201937, 0.16145114, 0.00484575, 0.02443373, 0.16145114, 0.51363285, 0.02443373, 0.07773228;...
                0.00484575, 0.02443373, 0.03201937, 0.16145114, 0.09732026, 0.00484575, 0.64306463, 0.03201937;...
                0.00484575, 0.02443373, 0.03201937, 0.16145114, 0.02443373, 0.07773228, 0.16145114, 0.51363285];
AMean = 7 % Scale to match average output with SCU preference in Mendoza (2010)
A = AMean*[0.986595988257; 1.013404011625; 0.986595988257; 1.013404011625; 0.986595988257; 1.013404011625; 0.986595988257;
      1.013404011625];
R = [1.064449652804; 1.064449652804; 1.064449652804; 1.064449652804; 1.106970405389; 1.106970405389; 1.106970405389; 1.106970405389];
p = [0.993455002946; 0.993455002946; 1.062225686321; 1.062225686321; 0.993455002946; 0.993455002946; 1.062225686321; 1.062225686321];

% State
var_state cTilde k;

cTilde_min = 100;
cTilde_max = 300;
cTilde_shift = 10;
cTilde_pts = 30;
cTilde = linspace(cTilde_min,cTilde_max,cTilde_pts);

k_min = 600;
k_max = 900;
k_shift = 10;
k_pts = 30;
k = linspace(k_min,k_max,k_pts);

% Initial period
var_policy_init L v;
inbound_init L 0 1000;
inbound_init v 0 1000;

var_aux_init flow q d;

model_init;
  % Some trivial equations
  w = L^(omega-1)*(1+tau);
  Y = A* k^gamma * L^alpha * v^eta;
  F_1 = gamma*Y/k;
  F_2 = alpha*Y/L;
  F_3 = eta*Y/v;

  NL = L^omega/omega;
```

```
  flow = Y - p*v - phi*(R-1)*(w*L+p*v) - NL*(1+tau);

  lambda = cTilde^(-sigma);
  q = 1;
  d = F_1-delta;

  equations;
    F_2 - w*(1+phi*(R-1));
    F_3 - p*(1+phi*(R-1));
  end;
end;

% Interpolation
var_interp flow_interp q_plus_d_interp;
% Initial
initial flow_interp flow/flow_scale;
initial q_plus_d_interp q + d;
% Transition
flow_interp = flow/flow_scale;
q_plus_d_interp = q + d;

% Policy
ADAPTIVE_FACTOR = 1.5;
var_policy L v mu nbNext kNext cTildeNext[8];
inbound L 0 1000 adaptive(ADAPTIVE_FACTOR);
inbound v 0 1000 adaptive(ADAPTIVE_FACTOR);
inbound mu 0 1;
inbound nbNext 0 2000 adaptive(ADAPTIVE_FACTOR);
inbound kNext 0 1000 adaptive(ADAPTIVE_FACTOR);
inbound cTildeNext 0.0 500 adaptive(ADAPTIVE_FACTOR);

% Extra variables returned
var_aux lambda flow q d c Y bNext b inv nx gdp b_gdp nx_gdp lev wkcptl;
var_output c Y cTildeNext q mu inv kNext bNext b v nx gdp b_gdp nx_gdp lev wkcptl;

model;
  % Output and marginal product
  Y = A* k^gamma * L^alpha * v^eta;
  F_1 = gamma*Y/k;
  F_2 = alpha*Y/L;
  F_3 = eta*Y/v;
  w = L^(omega-1)*(1+tau);

  % Some calculations
  lambda = cTilde^(-sigma)/(1+tau);
  inv = kNext - k*(1-delta) + a/2*(kNext-k)^2/k;
  q = 1+a*(kNext-k)/k;
  d = F_1 - delta + a/2*(kNext-k)^2/(k^2);
  qb = 1/R;

  % Transform back bNext
  bNext = (nbNext + phi*R*(w*L+p*v) - kappa*q*kNext)/qb;

  % Interpolation
  [flow_future',q_plus_d_future'] = GNDSGE_INTERP_VEC'(cTildeNext',kNext);
  flow_future' = flow_future'*flow_scale;

  % some named equations
  lambda_future' = cTildeNext'^(-sigma)/(1+tau);
  euler_bond_residual = -1 + mu + beta*GNDSGE_EXPECT{lambda_future'}/(qb*lambda);
  euler_capital_residual = -1 + kappa*mu + beta*GNDSGE_EXPECT{lambda_future'*q_plus_d_future'}/(q*lambda);
  % consistency equation
  cTilde_consis' = flow_future' + bNext - cTildeNext'*(1+tau);

  % Extra variables returned
  NL = L^omega/omega;
  flow = Y - p*v - phi*(R-1)*(w*L+p*v) - qb*bNext - inv - NL*(1+tau);
  c = cTilde + NL;
  b = cTilde*(1+tau) - flow;
  gdp = Y-p*v;
  b_gdp = b / gdp;
  nx = qb*bNext - b + phi*(R-1)*(w*L+p*v);
  nx_gdp = nx / gdp;
  lev = (qb*bNext-phi*R*(w*L+p*v)) / (q*kNext);
```

```
  wkcptl = phi*(w*L+p*v);

  equations;
    euler_bond_residual;
    nbNext*mu;
    euler_capital_residual;
    F_2 - w*(1+phi*(R-1)+mu*phi*R);
    F_3 - p*(1+phi*(R-1)+mu*phi*R);
    cTilde_consis';
  end;
end;

simulate;
  num_periods = 50000;
  num_samples = 24;
  initial cTilde 200;
  initial k 800;
  initial shock ceil(shock_num/2);

  var_simu c Y q inv b mu bNext v nx gdp b_gdp nx_gdp lev wkcptl;
  cTilde' = cTildeNext';
  k' = kNext;
end;
```

B.2 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;
```

B.3 Guvenen (2009)

```
% Parameters
parameters beta alpha rhoh rhon theta delta mu xsi chi a1 a2 Kss Bbar bn_shr_lb bn_shr_ub varianceScale;

beta = 0.9966;      % discount factor
alpha = 6;          % risk aversion
rhoh = 1/.3;        % inv IES for stockholders
rhon = 1/.1;        % inv IES for non-stockholders
theta = .3;         % capital share
delta = .0066;      % depreciation rate
mu = .2;            % participation rate
xsi = .4;           % adjustment cost coefficient
```

```
chi = .005;         % leverage ratio
a1 = (((delta^(1/xsi))*xsi)/(xsi-1));
a2 = (delta/(1-xsi));
Kss = ((1/beta-1+delta)/theta)^(1/(theta-1));
Bbar = -0.6*(1-theta)*Kss^theta; %borrowing constraint
varianceScale = 1e4;

TolEq = 1e-4;
INTERP_ORDER = 4; EXTRAP_ORDER = 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^theta);
  W = (1-theta)*Z*(K^theta);
  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-rhoh)))^(1/(1-rhoh));
  vn = ((1-beta)*(c_n^(1-rhon)))^(1/(1-rhon));
  vhpow = vh^(1-alpha);
  vnpow = vn^(1-alpha);
  Pf = 0;
  Ps = 0;
  Div = Y - W - (1-Pf)*chi*Kss;  % investment is zero

  Eulerstock = (vh^(rhoh-alpha))*(c_h^-rhoh)*(Ps + Div);
  Eulerbondh = (vh^(rhoh-alpha))*(c_h^-rhoh);
  Eulerbondn = (vn^(rhon-alpha))*(c_n^-rhon);

  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^(rhoh-alpha))*(c_h^-rhoh)*(theta*Z*(K^(theta-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 std_ExcessR
    SharpeRatio;

model;
  Y = Z*(K^theta);              % output
  W = (1-theta)*Z*(K^theta);    % 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] =
      GDSGE_INTERP_VEC(shock,Knext,bn_shr_next);
  EPD_square_future = EPD_square_future*varianceScale;

  vh =  ((1-beta)*(c_h^(1-rhoh)) + beta*(Evh_future^((1-rhoh)/(1-alpha))))^(1/(1-rhoh));
  vn =  ((1-beta)*(c_n^(1-rhon)) + beta*(Evn_future^((1-rhon)/(1-alpha))))^(1/(1-rhon));

  Eulerstock = (vh^(rhoh-alpha))*(c_h^-rhoh)*(Ps + Div);
  Eulerbondh = (vh^(rhoh-alpha))*(c_h^-rhoh);
  Eulerbondn = (vn^(rhon-alpha))*(c_n^-rhon);

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

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

  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 = (GDSGE_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^((alpha-rhoh)/(1-alpha))))/((c_h^(-rhoh))*Ps);
  foc_bondh = 1 - (beta*EEulerbondh_future*(Evh_future^((alpha-rhoh)/(1-alpha))) + lambdah)/((c_h^(-rhoh))*Pf);
  foc_bondn = 1 - (beta*EEulerbondn_future*(Evn_future^((alpha-rhon)/(1-alpha))) + lambdan)/((c_n^-rhon)*Pf);
  foc_f = 1 - (beta*EEulerf_future*(Evh_future^((alpha-rhoh)/(1-alpha))))/((c_h^(-rhoh))*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 = 10000;
  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 R_ep SharpeRatio std_ExcessR;

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

B.4 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;
```

B.5 Guerrieri et al. (2020)

```
% Parameters
```

```
parameters beta rho sigma phi nbar delta Abar;
beta = 0.99;    % discount factor
rho = 0.75;     % 1/rho intratemporal elasticity
sigma = 0.5;    % 1/sigma intertemporal elasticity
phi = 0.2;      % share of sector 1
nbar = 1;       % normal labor endowment
delta = 0.5;    % fraction of labor endowment during crisis
Abar = 0.3;     % borrowing limit
TolEq = 1e-8;   % Solve with high adccuracy

% Shocks
var_shock n1;
shock_num = 2;
pi2 = 0.5;      % the pandemic lasts for 2 quarters
freq = 0.005;   % frequency of pandemic: 0.5 percent of the time.
pi1 = 1 - (freq/(1-freq))*(1-pi2);
shock_trans = [pi1,1-pi1;
               1-pi2,pi2];
n1 = [nbar,delta*nbar];

% Endogenous States
var_state a1;
Ngrid = 301;
a1_lb = -Abar;
a1_ub = (1-phi)*Abar/phi;
a1 = linspace(a1_lb,a1_ub,Ngrid);

% Last period
var_policy_init c1_shr;
inbound_init c1_shr 0 1;
var_aux_init P1 log_lambda1 log_lambda2;

model_init;
   c1_1 = c1_shr*(phi*n1)/phi;
   c2_1 = (1-c1_shr)*(phi*n1)/(1-phi);
   c1_2 = c1_shr*((1-phi)*nbar)/phi;
   c2_2 = (1-c1_shr)*((1-phi)*nbar)/(1-phi);

   Y = (phi*n1^(1-rho) + (1-phi)*nbar^(1-rho))^(1/(1-rho));
   lambda1 = (c1_shr/phi*Y)^(-sigma)*(Y/nbar)^rho;
   lambda2 = ((1-c1_shr)/(1-phi)*Y)^(-sigma)*(Y/nbar)^rho;
   log_lambda1 = log(lambda1);
   log_lambda2 = log(lambda2);

   % price of good 1
   P1 = ((c1_1/phi)/(c1_2/(1-phi)))^(-rho);
   % wage of sector 1
   W1 = P1;
   budget1_resid = P1*c1_1 + c1_2 - W1*n1 - a1;

  equations;
     budget1_resid;
  end;
end;

var_interp log_lambda1_interp log_lambda2_interp;
initial log_lambda1_interp log_lambda1;
initial log_lambda2_interp log_lambda2;
% Updates
log_lambda1_interp = log_lambda1;
log_lambda2_interp = log_lambda2;

% Endogenous variables, bounds, and initial values
var_policy c1_shr a1n mu1 mu2 r;
inbound c1_shr 0 1;
inbound a1n -Abar (1-phi)*Abar/phi;
inbound mu1 0 1;
inbound mu2 0 1;
inbound r -0.5 0.5;

% Other equilibrium variables
var_aux a2 P1 log_lambda1 log_lambda2;

model;
```

```
   a2 = -a1*phi/(1-phi);
   c1_1 = c1_shr*(phi*n1)/phi;
   c2_1 = (1-c1_shr)*(phi*n1)/(1-phi);
   c1_2 = c1_shr*((1-phi)*nbar)/phi;
   c2_2 = (1-c1_shr)*((1-phi)*nbar)/(1-phi);

   Y = (phi*n1^(1-rho) + (1-phi)*nbar^(1-rho))^(1/(1-rho));
   lambda1 = (c1_shr/phi*Y)^(-sigma)*(Y/nbar)^rho;
   lambda2 = ((1-c1_shr)/(1-phi)*Y)^(-sigma)*(Y/nbar)^rho;
   log_lambda1 = log(lambda1);
   log_lambda2 = log(lambda2);

   % price of good 1
   P1 = ((c1_1/phi)/(c1_2/(1-phi)))^(-rho);
   % wage of sector 1
   W1 = P1;

   log_lambda1Future' = log_lambda1_interp'(a1n);
   log_lambda2Future' = log_lambda2_interp'(a1n);
   lambda1Future' = exp(log_lambda1Future');
   lambda2Future' = exp(log_lambda2Future');

   budget1_resid = P1*c1_1 + c1_2 + a1n/(1+r) - W1*n1 - a1;
   euler_residual = 1 - beta*(1+r) * GDSGE_EXPECT{lambda1Future'}/lambda1 - mu1;
   euler_residua2 = 1 - beta*(1+r) * GDSGE_EXPECT{lambda2Future'}/lambda2 - mu2;

   a2n = -a1n*phi/(1-phi);
   slackness1 = mu1*(a1n + Abar);
   slackness2 = mu2*(a2n + Abar);
  equations;
     budget1_resid;
     euler_residual;
     euler_residua2;
     slackness1;
     slackness2;
  end;
end;

simulate;
  num_periods = 10000;
  num_samples = 20;
  initial a1 0;
  initial shock 1;

  var_simu a2 P1 r c1_shr;

  a1' = a1n;
end;
```

C User Manual

The user manual, online compiler, gmod files and other examples can be found on the toolbox's website: http://www.gdsge.com.