

NBER WORKING PAPER SERIES

A SIMPLE BUT POWERFUL SIMULATED CERTAINTY EQUIVALENT APPROXIMATION  
METHOD FOR DYNAMIC STOCHASTIC PROBLEMS

Yongyang Cai  
Kenneth L. Judd

Working Paper 28502  
<http://www.nber.org/papers/w28502>

NATIONAL BUREAU OF ECONOMIC RESEARCH  
1050 Massachusetts Avenue  
Cambridge, MA 02138  
February 2021

Cai and Judd appreciate the financial support from United States Department of Agriculture NIFA-AFRI grant 2015-67023-22905; Cai acknowledges the financial support from the National Science Foundation (SES-1463644 and SES-1739909) and USDA-NIFA-AFRI grant 2018-68002-27932. This paper originates from Cai et al. (2020), which has been separated into this methodological paper and another paper focusing on applications to large natural resource allocation problems. The views expressed herein are those of the authors and do not necessarily reflect the views of the National Bureau of Economic Research.

NBER working papers are circulated for discussion and comment purposes. They have not been peer-reviewed or been subject to the review by the NBER Board of Directors that accompanies official NBER publications.

© 2021 by Yongyang Cai and Kenneth L. Judd. All rights reserved. Short sections of text, not to exceed two paragraphs, may be quoted without explicit permission provided that full credit, including © notice, is given to the source.

A Simple but Powerful Simulated Certainty Equivalent Approximation Method for Dynamic Stochastic Problems

Yongyang Cai and Kenneth L. Judd

NBER Working Paper No. 28502

February 2021

JEL No. C61,C63,C68,E31,E52,Q54,Q58

**ABSTRACT**

We introduce a novel simulated certainty equivalent approximation (SCEQ) method for solving dynamic stochastic problems. Our examples show that this method only requires a desktop computer to solve high-dimensional finite- or infinite-horizon, stationary or nonstationary dynamic stochastic problems with hundreds of state variables, a wide state space, and occasionally binding constraints. The SCEQ method is simple, stable, and efficient, which makes it suitable for solving complex economic problems that cannot be solved by other algorithms.

Yongyang Cai  
Agricultural, Environmental and  
Development Economics  
The Ohio State University  
Columbus, OH 43210  
US  
cai.619@osu.edu

Kenneth L. Judd  
Hoover Institution  
Stanford University  
Stanford, CA 94305-6010  
and NBER  
kennethjudd@mac.com

A data appendix is available at  
<http://www.nber.org/data-appendix/w28502>

A Code is available at  
<https://drive.google.com/file/d/1SP-YyhKE6o1ZpHpdxo4HJ72lPYplWpej/view>

# 1 Introduction

Dynamic stochastic general equilibrium (DSGE) problems are often studied using stationary models, which are relatively easy to solve using local approximation methods.<sup>1</sup> Stationary problems have time-invariant decision rules, so in low-dimensional models, projection methods can be used to compute global solutions for value functions and/or decision rules.<sup>2</sup>

However, neither stationarity nor low dimensionality are reasonable simplifications for real-world problems. Life-cycle models often have time-varying endowments and preferences, and the population, technology, resource stock, and climate all change over time, making those problems nonstationary. Computing the full set of state-contingent decisions and prices is often too computationally costly, and may also produce more information than is needed for the question at hand. In addition, it is challenging to compute global solutions for non-smooth value functions and/or decision rules of DSGE problems with occasionally binding constraints. This paper proposes a simple but powerful Simulated Certainty Equivalent (SCEQ) method which only uses the standard tools available to economists to compute the properties of interest of many high-dimensional, nonstationary models, and/or problems with occasionally binding constraints.

The key reason for the advantages of SCEQ is its notion of “solution”. Most methods for solving dynamic economic models solve for value functions, decision rules, and/or price functions. Solving for unknown functions requires efficient ways to approximate functions, a difficult challenge for complex high-dimensional problems. However, the solutions are often used only to compute stochastic properties of the problem, such as means, variances, covariances, and trends, as well as optimal decisions at the initial periods for policy makers. Moreover, a low-degree approximation of value or policy functions, such as linear or log-linear approximation from perturbation methods, may create large approximation errors and lead to large errors in estimating the stochastic properties of the solution. Instead, SCEQ focused directly on computing a set of simulations of the solution without approximating functions, and use them to directly compute the statistical properties of the solution. The key

---

<sup>1</sup>See e.g., Magill (1977); Blanchard and Kahn (1980); Kydland and Prescott (1982); Judd and Guu (1993); Jin and Judd (2002).

<sup>2</sup>See e.g. Haan and Marcet (1990); Judd (1992, 1998); Marcet and Lorenzoni (1998); Miranda and Fackler (2002)

reason of SCEQ’s desirable properties is that it’s concept of solution is more limited than other methods but often sufficient to deliver the information a researcher wants to have.

SCEQ is simple, for several reasons. First, SCEQ does not use complicated computational methods like (adaptive) sparse grids, or require extensive resources like a supercomputer. Second, SCEQ does not need to use the “trial and guess” tricks that are often used in other methods. For example, projection methods often have to guess an approximation domain and an approximation method, then check if these can solve the problem, but this process may have to be repeated many times. Third, SCEQ only requires a numerical optimization solver, as it does not use a value or policy function approximation or numerical integration. Fourth, SCEQ is easy to program in common languages like GAMS (General Algebraic Modeling System)<sup>3</sup> and Matlab.<sup>4</sup> Fifth, it is easy and quick for SCEQ to check if its solution has the desired accuracy.

SCEQ is also powerful. First, SCEQ does not suffer from the so-called “curse of dimensionality”: its accuracy is independent of the state space’s dimensionality, and its computational time is at worst a cubic function of the state space’s dimensionality.<sup>5</sup> In fact, when we increase the number of countries in our real business cycle problems from  $N$  to  $2N$ , the total computational time of the SCEQ algorithm increases at only a nearly linear speed. Second, SCEQ efficiently solves high-dimensional dynamic stochastic problems with occasionally binding constraints: it only took hours on a standard computer to provide 1,000 simulation paths of solutions for 400-dimensional real business cycle problems with a lower bound on investments. Third, SCEQ efficiently solves nonstationary dynamic stochastic problems, even those with high dimensionality and occasionally binding constraints. Fourth, SCEQ can solve problems with a wide state space domain, even if high dimensionality, occasionally binding constraints, and nonstationarity are also present. Fifth, SCEQ’s efficiency (computa-

---

<sup>3</sup>There are many reliable and efficient professional solvers available using GAMS (McCarl et al., 2016), such as CONOPT (Drud, 1994) and SNOPT (Gill, Murray, and Saunders, 2005) for solving nonlinear programming problems. Moreover, these solvers are provided in the NEOS server (Czyzyk, Mesnier, and More, 1998) (<https://neos-server.org/neos/solvers/index.html>) and are free to use with GAMS.

<sup>4</sup>SCEQ can be implemented with Dynare (Adjemian et al., 2011), a Matlab toolbox for solving DSGE models, particularly stationary problems in macroeconomics.

<sup>5</sup>The “curse of dimensionality” refers to the case where computational cost grows exponentially with the dimensionality of state space.

tional time and accuracy) is independent of the number of exogenous state variables, because there are no numerical approximation or integration over them in SCEQ. Sixth, SCEQ can be very fast. In our applications, using a Mac Pro desktop computer, SCEQ only took minutes to provide all possible simulation paths of solutions for a 200-country real business cycle problem (Section 4.2.2), and only took hours to provide 1,000 simulation paths of solutions for larger 200-country real business cycle problems with 201 or 400 state variables (Sections 4.2.1 and 4.2.3). Seventh, SCEQ is stable and can solve both stochastic dynamic programming problems and stochastic dynamic competitive equilibrium problems. Lastly, SCEQ is highly accurate, providing solutions with an accuracy of 0.1-1% for all examples in this paper.

In this paper, we show that the SCEQ is highly accurate and achieves stable numerical solutions for dynamic stochastic problems. Our first numerical example is for illustration only. The second example is a multi-country real business cycle model with occasionally binding constraints. We solve cases with 10, 20, 50, 100, and 200 countries, each of which only takes minutes or hours on a Mac Pro desktop computer. The computational time is nearly linear to the number of countries, and increasing the dimensionality has little impact on the solution's accuracy. The third example is a nonstationary stochastic integrated assessment model of climate and the economy with six continuous endogenous state variables and one discrete state variable. Moreover, the model has occasionally binding constraints that make it challenging to solve with standard methods, but SCEQ took less than two hours to provide 1,000 simulation paths of solutions such as the optimal carbon taxes on a Mac Pro desktop computer. The last example applies SCEQ to solve a New Keynesian model with a zero lower bound, showing that SCEQ can solve stochastic competitive equilibrium problems. All the examples demonstrate that SCEQ can solve dynamic stochastic problems with a 0.1-1% degree of accuracy, which is within the acceptable range of accuracy for most dynamic stochastic economic problems.

The paper is organized as follows. Section 2 compares SCEQ with existing methods. Section 3 introduces the SCEQ method. Section 4 provides numerical examples. Section 5 discusses some properties of SCEQ. Section 6 concludes.

## 2 Comparison with Existing Methods

To solve stationary and smooth problems, perturbation or projection methods are commonly used, which each have their limitations. Perturbation methods can only

provide locally accurate solutions around the non-stochastic steady state, which are then treated as if they were globally valid. But it is often not sufficient for decision-making purposes because the non-stochastic steady state may be far away from the initial state. A practical dynamic optimization problem aims to provide initial-period decisions that are based on those periods' states, rather than states in the far future (although future dynamics can affect initial-period solutions; see Cai, Judd, and Steinbuks 2017 for further discussion of this issue), which imply a wide approximation domain for approximating value or policy functions. But a wide approximation domain requires a high-degree approximation of value or policy functions to obtain accurate solutions around the initial periods' states, implying that these problems will also be difficult for projection methods. Moreover, projection methods face challenges when solving high-dimensional problems or problems with strong nonlinearity.

Value function iteration (VFI) and time iteration are two common methods to solve dynamic stochastic problems. For nonstationary problems, time-varying approximation domains can be used to obtain accurate solutions while keeping a low-degree approximation for each period. For example, Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) apply the parallel VFI method to solve many large-scale dynamic stochastic integrated models of climate and the economy (DSICE). However, VFI and time iteration both face challenging issues such as high-dimensional state spaces, shape-preservation of value functions, appropriate approximation domains, and occasionally binding constraints. Brumm and Scheidegger (2017) introduce adaptive sparse grids in time iteration to overcome the kink problem, but their method requires complicated coding and still requires a large number of approximation points around the kinks. Moreover, if the problem is nonstationary, then the adaptive sparse grid method would be less efficient as it cannot reuse the last iteration's grids.<sup>6</sup>

Cai, Judd, and Steinbuks (2017) propose a stable and efficient nonlinear certainty equivalent approximation method (NLCEQ) for solving stationary dynamic stochastic problems. NLCEQ applies the concept of certainty equivalent approximation to transform an infinite-horizon stochastic problem into a finite-horizon deterministic problem, solves it to obtain the optimal solution of decisions for each approximation

---

<sup>6</sup>See Rust (1996), Judd (1998), Ljungqvist and Sargent (2000), Miranda and Fackler (2002), Bertsekas (2005, 2007), Aruoba, Fernandez-Villaverde, and Rubio-Ramirez (2006), Kollmann et al. (2011), Juillard and Villemot (2011), Guerrieri and Iacoviello (2015), Fernandez-Villaverde, Rubio-Ramirez, and Schorfheide (2016), Fernandez-Villaverde and Levintal (2018), Levintal (2018), and Cai (2019) for more discussion on the conventional methods.

node in the state space at the initial time, then fits the solutions with a globally non-linear approximation to the optimal policy function. Cai, Judd, and Steinbuks (2017) demonstrate that NLCEQ can be used to achieve accurate solutions to a variety of problems, including high-dimensional problems such as a real business cycle model with many countries, and problems with kinks such as a New Keynesian model with a zero lower bound. For high-dimensional problems, NLCEQ uses sparse grids approximation (see, e.g., Smolyak 1963; Malin, Krueger, and Kubler 2011; Judd et al. 2014), which can work well for smooth policy functions. For problems with kinks, NLCEQ can use high-degree approximations or adaptive sparse grids to reduce approximation errors caused by kinks.

Similar to NLCEQ, the SCEQ method uses the certainty equivalent approximation idea to transform infinite- or finite-horizon stochastic problems into deterministic finite-horizon problems in order to solve them, so it inherits the stability and efficiency properties of NLCEQ. However, SCEQ uses a period-by-period approach to construct one simulation path of solutions: first, it solves the transformed deterministic problem starting with a given initial state at the initial time  $s = 0$ , and uses the solution at  $s = 0$  to simulate the shocks, generating a simulated state at time  $s = 1$ . It then solves the new transformed deterministic problem starting with the simulated state at time  $s = 1$  to obtain the solution at  $s = 1$ , which is used together with new simulated shocks to generate a new simulated state at time  $s = 2$ . This process is iterated for each period until the time of interest to generate one simulation path. SCEQ obtains a set of simulation paths by repeating the period-by-period approach with different realization paths of the shocks, thus obtaining distributions of states and decisions for economic analysis.

Compared to the NLCEQ method, SCEQ has the following advantages:

(i) SCEQ can solve problems that are challenging for NLCEQ. If a problem has both a high-dimensional state space and kinks in the optimal policy function, then NLCEQ will have to choose a large number of approximation nodes in the state space to approximate this policy function with a sufficient level of accuracy. It will then take too much time to solve the large-scale optimization problems corresponding to the approximation nodes. In contrast, SCEQ is very suitable for solving such problems.

(ii) NLCEQ needs to specify an approximation domain in the state space *a priori*, whereas SCEQ does not. NLCEQ requires this specification to choose approximation nodes, but we often do not know how wide a domain we should choose. If it is too

wide, then it requires too many approximation nodes to obtain a good approximation to the policy function, implying too many large-scale optimization problems. If it is too narrow, then next period’s optimal state will often be beyond the approximation domain, which limits the subsequent economic analysis. In contrast, if we want to construct a policy function optimization using simulated solutions from SCEQ, we can easily define an approximation domain with the minimum and maximum of the simulated solutions, or we can use a small set of “representative” points to cover the support of the simulated solutions and then use projection techniques (Maliar and Maliar, 2015).<sup>7</sup>

(iii) To obtain the same solution accuracy as SCEQ, NLCEQ suffers from the “curse of dimensionality” of the state space, both in endogenous and exogenous state variables.

(iv) NLCEQ only solves infinite-horizon stationary problems,<sup>8</sup> while SCEQ can solve both stationary and nonstationary problems in an infinite or a finite horizon.

(v) SCEQ can quickly check if its solution is accurate: it can just obtain one or several simulated paths at first and measure their accuracy. But NLCEQ has to wait until all of its optimization problems are solved, which incurs a larger computational cost.

In certain contexts, NLCEQ has its advantages. For example, NLCEQ could be faster than SCEQ for low-dimensional stationary problems with smooth policy functions, and NLCEQ can be applied to find optimal policy functions for deterministic problems.

VFI and time iteration have similar disadvantages when compared with SCEQ. They face even worse challenges than NLCEQ when solving problems with both a high-dimensional state space and kinks in the optimal policy function: they have to choose (time-varying) approximation domains in the state space *a priori*; they suffer

---

<sup>7</sup>SCEQ’s solution could have a wide domain, particularly if its initial state is far away from the non-stochastic steady state for a stationary problem. For example, in our real business cycle model, the initial levels of capital range from 0.1 to 10 across countries, while the steady state level is 1.

<sup>8</sup>NLCEQ can be adapted to solve nonstationary problems by iterating the process. That is, we apply NLCEQ for the first period with a given approximation domain and use its solution to generate an approximation domain for the second period. We then apply NLCEQ for the second period and use its solution to generate an approximation domain for the third period, and repeat this process for each subsequent period until the end of the required time period. This iterative method may incur a large computational cost, as NLCEQ has to solve a large number of large-scale optimization problems for every period.



the “curse of dimensionality” for both endogenous and exogenous state variables;<sup>9</sup> they are less flexible in checking whether they work or not, as they have to wait until the iteration converges for a stationary problem before checking their solution; and they require more complicated coding in low-level programming languages like Fortran or C, sparse grids, or parallelism for high-dimensional problems. In addition, VFI or time iteration are generally not stable.<sup>10</sup>

SCEQ has advantages over other proposed solution methods. Judd, Maliar, and Maliar (2011) suggest a generalized stochastic simulation algorithm (GSSA), and Judd, Maliar, and Maliar (2012) and Maliar and Maliar (2015) propose an  $\epsilon$ -distinguishable set (EDS) method to merge projection approaches and simulation. However, both methods only solve infinite-horizon stationary problems. Moreover, their methods are less flexible in checking whether they work or not, as they have to obtain a large number of simulated results before they can approximate policy functions. Grune, Semmler, and Stieler (2015) apply a nonlinear model predictive control method for solving dynamic problems, which only solves infinite-horizon stationary dynamic programming problems and focuses on deterministic problems. It does not provide a distribution of solutions or policy function approximation, nor accuracy measures for stochastic problems.

### 3 The SCEQ Method

Let  $\mathbf{x}_t$  be a vector of state variables (e.g., capital), and  $\mathbf{a}_t$  be a vector of decision variables (e.g., consumption) at each time  $t$ . The transition law of the state vector  $\mathbf{x}$  is

$$\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$$

where  $\epsilon_t$  is a serially uncorrelated random vector process<sup>11</sup> and  $\mathbf{g}_t$  is a vector of functions which could be time-varying: its  $i$ -th element,  $g_{t,i}$ , returns the  $i$ -th state variable at time  $t + 1$ :  $x_{t+1,i}$ .

---

<sup>9</sup>There are methods to alleviate or even avoid the “curse of dimensionality” for some problems, see e.g. Rust (1997, 2019), Judd et al. (2014), Brumm and Scheidegger (2017), Cai (2019), and Scheidegger and Bilonis (2019).

<sup>10</sup>Note that VFI can be stable if it uses a shape-preserving approximation method (Cai and Judd, 2013).

<sup>11</sup>If a dynamic model has serially correlated random variables, they are exogenous state variables, and we can use an uncorrelated vector  $\epsilon_t$  in their transition laws.

Without loss of generality, we assume the mean or median of  $\epsilon_t$  is 0. For notational simplicity we keep the same mathematical representation of a transition function even if some of its elements are redundant. For example, if  $g_{t,i}$  is deterministic, i.e.,  $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t)$ , since it can be rewritten as  $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t) + 0 \cdot \epsilon_t$ , we will still denote it as  $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$ . Similarly, if there are some unused elements of  $\epsilon_t$  or some redundant arguments in a function  $g_{t,j}$ , we can multiply them by zero in  $g_{t,j}$  and thus still use  $x_{t+1,j} = g_{t,j}(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$ .

We solve the following social planner's problem:

$$\begin{aligned} \max_{\mathbf{a}_t} \quad & \mathbb{E} \left\{ \sum_{t=0}^{T-1} \beta^t u_t(\mathbf{x}_t, \mathbf{a}_t) + \beta^T V_T(\mathbf{x}_T) \right\} \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t), \quad t = 0, 1, 2, \dots, T-1, \\ & \mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0, \quad t = 0, 1, 2, \dots, T-1, \end{aligned} \tag{1}$$

where  $u_t$  is a utility function which could be time-varying,  $\beta \in (0, 1)$  is the discount factor,  $\mathbb{E}$  is the expectation operator,  $T$  is the horizon ( $T = \infty$  if it is an infinite-horizon problem),  $V_T(\mathbf{x}_T)$  is a given terminal value function depending on the terminal state  $\mathbf{x}_T$  (it is zero everywhere for an infinite-horizon problem), and  $\mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0$  represents the feasibility constraints for actions  $\mathbf{a}_t$  (e.g., nonnegativity constraints  $\mathbf{a}_t \geq 0$ ).<sup>12</sup> We assume that the initial state  $\mathbf{x}_0$  is given, as it can usually be observed or estimated.

We assume that the social planner is interested in solutions for the first  $T^*$  periods. Macroeconomists are often interested in obtaining solutions around the non-stochastic steady state. However in reality, the initial state could be far away from the steady state, and a policymaker may be more interested in the solutions for the initial periods in the forward-looking model (1) than the far future states that could be around the steady state. For example, in environmental and climate change economics, we are often interested in solutions for the coming century rather than longer time periods. The following SCEQ algorithm obtains solutions for the first  $T^*$  periods.

---

<sup>12</sup>An equality constraint  $f(x, a) = 0$  can be written as a combination of  $f(x, a) \geq 0$  and  $-f(x, a) \geq 0$ .

---

**Algorithm 1** SCEQ for Stochastic Dynamic Programming Problems

---

**Step 1. Initialization step.** Given the initial state  $\mathbf{x}_0$  and a time of interest  $T^*$ , choose a time-varying number of periods  $\Delta_s$  and a time-varying “terminal” value function  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  for each time  $s$ . Simulate a sequence of  $\epsilon_t$  to get  $m$  paths, denoted  $\epsilon_t^i$  for path  $i$ , from  $t = 0$  to  $T^* - 1$ . Let  $\mathbf{x}_0^i = \mathbf{x}_0$  and iterate forward through steps 2 and 3 for  $s = 0, 1, 2, \dots, T^* - 1$ .

**Step 2. Optimization step.** Solve the following deterministic model starting at time  $s$  and simulated node  $\mathbf{x}_s^i$ :

$$\begin{aligned} \max_{\mathbf{a}_t} \quad & \sum_{t=s}^{s+\Delta_s-1} \beta^{t-s} u_t(\mathbf{x}_t, \mathbf{a}_t) + \beta^{\Delta_s} V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, 0), \quad t = s, s+1, \dots, s+\Delta_s-1, \\ & \mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0, \quad t = s, s+1, \dots, s+\Delta_s-1, \end{aligned} \tag{2}$$

where  $\mathbf{x}_s$  is given by  $\mathbf{x}_s^i$ , for each  $i = 1, \dots, m$ .

**Step 3. Simulation step.** Set  $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \epsilon_s^i)$ , where  $\mathbf{a}_s^i$  is the optimal decision at time  $s$  of the problem (2), for each  $i = 1, \dots, m$ .

---

Algorithm 1 obtains simulated pathways of optimal decisions and states. It contains three steps: (i) the initialization step, which chooses an appropriate  $\Delta_s$  and “terminal” value function  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  and simulates the shocks; (ii) the optimization step, which solves the finite-horizon deterministic optimization problems (2); (iii) the simulation step, which uses the optimal decision of (2) at time  $s$  to generate simulated states at time  $s+1$ .

Note that the inside loop across  $i$  can be switched with the outside loop across time, that is, for each  $i$ , we can obtain one simulation path by iteratively solving (2) and simulating  $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \epsilon_s^i)$  for  $s = 0, 1, 2, \dots, T^* - 1$ . In addition, with a fixed initial state, the solutions at  $s = 0$  are independent of simulation, so the optimization step just needs to solve the case with  $i = 1$  and assign its solutions to the other cases with  $i = 2, \dots, m$ . Algorithm 1 can also be applied to problems without a fixed initial state, as sometimes the initial state might be uncertain or hard to evaluate accurately. In this paper, we always assume that the initial state is fixed, without loss of generality.

### 3.1 Initialization

In the initialization step of Algorithm 1, for a finite  $T$ -horizon problem, we can set  $\Delta_s = T - s$  so that  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  is always the true terminal value function  $V_T(\mathbf{x}_T)$ . For an infinite horizon problem,  $\Delta_s$  and  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  are chosen such that the solution of (2) at its starting time  $s$  is almost identical to a solution with a larger  $\Delta_s$  for every state at  $s$ .  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  is chosen to be an approximation of the cumulative optimal welfare from time  $s + \Delta_s$  to time  $T - 1$ , i.e.,

$$\begin{aligned}
 V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) \approx \max_{\mathbf{a}_t} \quad & \mathbb{E} \left\{ \sum_{t=s+\Delta_s}^{T-1} \beta^t u_t(\mathbf{x}_t, \mathbf{a}_t) + \beta^T V_T(\mathbf{x}_T) \right\} \\
 \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t), \quad t = s + \Delta_s, \dots, T - 1, \\
 & \mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0, \quad t = s + \Delta_s, \dots, T - 1,
 \end{aligned} \tag{3}$$

One example of such an approximation is  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) = \sum_{t=s+\Delta_s}^{T-1} \beta^t u_t(\mathbf{x}_t, \mathbf{a}_t^*(\mathbf{x}_t)) + \beta^T V_T(\mathbf{x}_T)$ , where  $\mathbf{a}_t^*(\mathbf{x}_t)$  is a guess of the optimal policy function at  $t$ . For an infinite horizon stationary problem, we can use  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) = u(\mathbf{x}_{s+\Delta_s}, \mathbf{a}^*(\mathbf{x}_{s+\Delta_s})) / (1 - \beta)$ . The difference between  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  and the true optimal welfare since time  $s + \Delta_s$  (the “truncation error”) may impact the solution at time  $s$ . If  $\Delta_s$  is large, then the “terminal” value function  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  often has little impact on the solution of (2) at its starting time  $s$  so the truncation error is small,<sup>13</sup> but it also implies that (2) is a larger optimization problem and takes more computational time to solve.

Thus, for an infinite-horizon problem, in the initialization step of Algorithm 1 we first choose a large  $\Delta_s$  and a reasonable  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$ . For a few test nodes in the state space at time  $s$  (e.g., some reasonable extreme points), we let  $\mathbf{x}_s$  be given by a test node and then solve (2), to make sure that a much larger  $\Delta_s$  will not change the solutions. We use these solutions as the “true” solutions. We then choose a smaller  $\Delta_s$  and a different  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$ , re-solve (2) at the same test nodes, and compare these solutions with the “true” solutions to estimate the truncation error. For a stationary problem, we suggest choosing a constant  $\Delta_s$  and a time-invariant “terminal” value

---

<sup>13</sup>Most infinite-horizon dynamic economic models assume that the system asymptotically evolves towards its stationary state, so the discount factor  $\beta < 1$  makes the terms  $\beta^{t-s} u_t(\mathbf{x}_t, \mathbf{a}_t)$  small in magnitude for  $t \geq s + \Delta_s$  with a large  $\Delta_s$ , and a smaller  $\beta$  implies that we can choose a smaller  $\Delta_s$  in SCEQ. Such a truncation is often used in the literature, see e.g. Nordhaus (2008); Grune, Semmler, and Stielor (2015); Cai, Judd, and Steinbuks (2017); Cai, Judd, and Lontzek (2017); Cai and Lontzek (2019); Maliar et al. (2020).

function. In the end, we choose the best pair of  $\Delta_s$  and  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$  in terms of computational speed and size of the truncation error.

## 3.2 Optimization

The optimization step of Algorithm 1 applies the original certainty equivalent approximation idea of the NLCEQ method: for a given state at time  $s$ ,  $\mathbf{x}_s^i$ , we replace all future stochastic variables by their corresponding certainty equivalent approximation (e.g. expectations or medians) conditional on the current state  $\mathbf{x}_s^i$ , and convert the dynamic stochastic problem (1) into the deterministic finite-horizon dynamic problem (2).

Since  $\epsilon_t$  is a serially uncorrelated stochastic process, if all transition laws are continuous we can replace  $\epsilon_t$  in (2) by its (zero) mean or median for simplicity. Generally we can replace  $\epsilon_t$  by a deterministic function of its standard deviation, so SCEQ may provide a more accurate solution or even solve problems such as stochastic volatility (see, e.g., Caldara et al. (2012)).

For problems with a discrete Markov chain in transition laws, to obtain the corresponding deterministic model (1) we can use the same technique as described in Cai, Judd, and Steinbuks (2017) for NLCEQ with a discrete stochastic state. That is, given the Markov chain realization at time  $s$ , we can compute expectations of the Markov chain at all times after  $s$  conditional on the value at time  $s$ , then replace the stochastic process by the path of the conditional expectations in the optimization step of Algorithm 1.

We implement the optimal control method (see, e.g., Cai 2019) to solve (2) numerically: we view (2) as a large-scale nonlinear constrained optimization problem with  $\{\mathbf{a}_t^i : t \geq s\}$  and  $\{\mathbf{x}_t^i : t \geq s\}$  as its variables, and the transition equations and feasibility restrictions as its constraints. The problem can be directly solved with an appropriate nonlinear optimization solver such as CONOPT (Drud, 1994).

## 3.3 Simulation

In the simulation step of Algorithm 1 we use the optimal decision  $\mathbf{a}_s^i$  to generate the next-period state,  $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \epsilon_s^i)$ , given the realization of shocks,  $\epsilon_s^i$ . Once we reach the state  $\mathbf{x}_{s+1}^i$  at time  $s+1$ , we come back to implement the optimization step and then the simulation step. In other words, Algorithm 1 uses an adaptive

management strategy: decisions are made for the current period in the face of future uncertain shocks; once the next-period shock is observed, decisions for the next period are made by re-optimizing given the observed shock and new state. Observe that the serial correlation of random variables has been captured in their associated transition laws. By repeating this process iteratively  $T^*$  times, we compute one simulated path of optimal decisions,  $\{\mathbf{a}_s^i\}_{s=0}^{T^*-1}$ , and states,  $\{\mathbf{x}_s^i\}_{s=0}^{T^*-1}$ , which corresponds to the realized path of shocks,  $\{\epsilon_s^i\}_{s=0}^{T^*-1}$ . Repeating over  $i$ , we compute  $m$  simulated paths of optimal states and decisions, and then obtain their distributions.

### 3.4 Parallelism

Relative to other methods, SCEQ could take longer to run low-dimensional problems, as it requires solving  $m \times T^*$  optimization problems (2). But for high-dimensional problems, SCEQ can take much less time to run. Moreover, since simulation paths are independent of each other, Algorithm 1 can be naturally parallelized across simulation paths: e.g., one simulation path per compute core, so the wall clock time could be around the time spent to solve  $T^*$  optimization problems (2), which could be fast with modern hardware and optimization solvers. For example, it took seconds or at most several minutes in all of our examples, including the real business cycle problem with 200 countries (201 or 400 state variables) and  $\Delta_s = 50$ , to solve  $T^*$  optimization problems (2) on a standard computer. Therefore, we solve the large scale dynamic stochastic problems in minutes or hours, using parallelism on six compute cores of a Mac Pro desktop. Moreover, the running time can be reduced to seconds or minutes if we use many compute cores in a supercomputer.

### 3.5 SCEQ for Competitive Equilibrium

Like NLCEQ, the SCEQ method can also be adapted to solve competitive equilibrium problems with transition laws of states  $\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$ , where  $\mathbf{x}_t$  is the state vector,  $\mathbf{a}_t$  is the action vector, and  $\epsilon_t$  is a serially uncorrelated random vector process. We first use the certainty equivalent approximation idea to make the stochastic problem deterministic. A deterministic model's equilibrium solution should satisfy a set of equations and inequalities (including transition laws of states, feasibility constraints for actions, Euler equations, market clearing conditions, and other first-order

conditions):

$$\begin{cases} \mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, 0), & t = 0, 1, 2, \dots, T-1, \\ \mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0, & t = 0, 1, 2, \dots, T-1, \\ \mathbf{H}_t(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}, \mathbf{a}_{t+1}) = 0, & t = 0, 1, 2, \dots, T-1, \end{cases} \quad (4)$$

where  $\mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0$  represents the feasibility constraints for actions at period  $t$ , and  $\mathbf{H}_t(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}, \mathbf{a}_{t+1}) = 0$  represents the Euler equations, market clearing conditions, and other first-order conditions for the transformed deterministic model. If there are occasionally binding constraints, then the arguments of  $\mathbf{H}_t$  should also contain corresponding Lagrange multipliers, which we omit here without loss of generality. We can truncate infinite-horizon problems into finite-horizon problems with a terminal condition, such as  $(\mathbf{x}_T, \mathbf{a}_T) = (\mathbf{x}_{ss}, \mathbf{a}_{ss})$  where the pair  $(\mathbf{x}_{ss}, \mathbf{a}_{ss})$  is the steady or asymptotic state and its associated action. For a finite-horizon problem, we also need a terminal condition for  $(\mathbf{x}_T, \mathbf{a}_T)$ . Without loss of generality, we assume that the terminal condition is that the terminal policy function is given (i.e.,  $\mathbf{a}_{s+\Delta_s} = \mathbf{a}_{s+\Delta_s}^*(\mathbf{x}_{s+\Delta_s})$  with a given  $\mathbf{a}_{s+\Delta_s}^*$ ). Algorithm 2 summarizes the SCEQ method for solving stochastic competitive equilibrium problems.

---

**Algorithm 2** SCEQ for Stochastic Competitive Equilibrium Problems

---

**Step 1. Initialization step.** Given the initial state  $\mathbf{x}_0$  and a time of interest  $T^*$ , choose a time-varying number of periods  $\Delta_s$  and a time-varying “terminal” policy function  $\mathbf{a}_{s+\Delta_s}^*(\mathbf{x}_{s+\Delta_s})$  for each time  $s$ . Simulate a sequence of  $\epsilon_t$  to get  $m$  paths, denoted  $\epsilon_t^i$  for path  $i$ , from  $t = 0$  to  $T^* - 1$ . Let  $\mathbf{x}_0^i = \mathbf{x}_0$  and iterate forward through steps 2 and 3 for  $s = 0, 1, 2, \dots, T^* - 1$ .

**Step 2. Optimization step.** Solve the following deterministic model starting at time  $s$  and simulated node  $\mathbf{x}_s^i$ :

$$\begin{aligned} \max_{\mathbf{a}_t} \quad & 1 & (5) \\ \text{s.t.} \quad & \mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, 0), \quad t = s, s + 1, \dots, s + \Delta_s - 1, \\ & \mathbf{f}_t(\mathbf{x}_t, \mathbf{a}_t) \geq 0, \quad t = s, s + 1, \dots, s + \Delta_s - 1, \\ & \mathbf{H}_t(\mathbf{x}_t, \mathbf{a}_t, \mathbf{x}_{t+1}, \mathbf{a}_{t+1}) = 0, \quad t = s, s + 1, \dots, s + \Delta_s - 1, \\ & \mathbf{a}_{s+\Delta_s} = \mathbf{a}_{s+\Delta_s}^*(\mathbf{x}_{s+\Delta_s}), \end{aligned}$$

where  $\mathbf{x}_s$  is given by  $\mathbf{x}_s^i$ , for each  $i = 1, \dots, m$ .

**Step 3. Simulation step.** Set  $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \epsilon_s^i)$ , where  $\mathbf{a}_s^i$  is the optimal decision at time  $s$  of the problem (5), for each  $i = 1, \dots, m$ .

---

The objective of the maximization problem (5) is a constant, as we are finding a feasible solution for competitive equilibrium and a nonlinear constraint optimization solver (e.g., CONOPT in GAMS, or fmincon in Matlab) can solve such problems. If (5) do not have inequality conditions or they are not binding at the solution, then it can also be solved by an equation solver (e.g., fsolve in Matlab). Like Algorithm 1, Algorithm 2 can also be naturally parallelized.

### 3.6 Policy Function Approximation

Note that the SCEQ method does not need to approximate a value or policy function. After we obtain the simulated paths, we can use them to conduct economic analysis directly. For example, we can compute the expectation, distribution, and moments from the simulated solutions. Section 4.3 shows that we can use the SCEQ method to estimate the social cost of carbon in the first 100 years in the presence of economic risk. However, if necessary, we can employ a projection approach like the least-squares method to construct a policy function approximation from simulated solutions if



there are a large number of simulated results. This method is particularly efficient for stationary problems, because all simulated decisions  $\mathbf{a}_s^i$  follow the same policy function for all  $s$  and  $i$ . Moreover, the simulated results often locate inside a narrow domain, so a low-degree approximation often has enough approximation accuracy over the domain if the policy function is smooth. Furthermore, we can follow the method of Maliar and Maliar (2015) to construct a fixed grid covering the support of the simulated solution, then use a projection approach to obtain a policy function approximation.

### 3.7 Accuracy Measures

It is important to check the accuracy of solutions obtained with SCEQ, because we cannot *a priori* determine how the certainty equivalence assumption affects the solution's accuracy.

A standard accuracy measure method computes errors in equilibrium conditions (see e.g., Jin and Judd (2002) and Kollmann et al. (2011)), with the major errors for dynamic problems often being Euler or Bellman equation errors. For example, Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) apply value function iteration to obtain value/policy function approximations in each period. They randomly choose a set of points in the state space at period  $t$ , compute new solutions at the points from the Bellman equation using the value function approximation at  $t + 1$ , then compare those true values with the values of the period- $t$  value/policy function approximation at these points. This procedure produces Bellman equation errors for accuracy measures.

Cai, Judd, and Steinbuks (2017) apply NLCEQ to obtain policy function approximations. As in Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019), they randomly choose a set of points,  $\Theta$ , in the current-period state space, use the approximate policy function to compute the next-period states and decisions for each point in  $\Theta$ , then compute Euler equation errors at those points. The same procedure was employed in Jin and Judd (2002) and Kollmann et al. (2011) to investigate the solution accuracy of the approximate global decision rule along a simulated path for stationary problems. Those error estimation procedures require a global value or policy function approximation, which could be challenging for high-dimensional problems, particularly if there are kinks in the value or policy functions. The key

challenge is to adapt those error accuracy measures, developed to evaluate global solutions of dynamic problems, to the SCEQ method, which produces only simulated paths.

In SCEQ, we follow the idea of Jin and Judd (2002) and Kollmann et al. (2011) to measure the Euler equation errors at the simulated states without constructing a global function approximation. Instead, we develop an error formula that implicitly constructs a local policy function approximation for each simulated state we use in error checking.

For each simulated state at period  $t$ , we choose a set of quadrature nodes and then use the state transition laws to obtain a set of states at period  $t+1$ , denoted  $S_{t+1}$ . For each state in  $S_{t+1}$ , we use it as the starting state and  $t+1$  as the starting period, then solve the optimization problem (2) in Algorithm 1 for dynamic programming problems, or (5) in Algorithm 2 for competitive equilibrium problems. Thus, we obtain decisions in  $t+1$  for all states in  $S_{t+1}$ , then use the corresponding quadrature rules to estimate the Euler equation error at the time- $t$  simulated state. The quadrature formulas are called “interpolatory rules” because they implicitly use local information to construct a local approximation adequate for the purpose of integration. That is, we can compute the Euler equation errors at the simulated states without explicitly constructing a policy function approximation.<sup>14</sup>

Note that for an infinite horizon problem, if we choose  $\Delta_s = \hat{T} - s$  then the infinite horizon problem is viewed as a finite  $\hat{T}$ -horizon problem in SCEQ, so the truncation error is not reflected in the Euler errors. However, for an infinite-horizon stationary problem, if we choose a constant  $\Delta_s$  and a time-invariant “terminal” value function, then the truncation error is reflected in the Euler errors (see Appendix A.1 for more details). For some special cases like the example in Section 4.2.2, we can compute all possible paths within a time of interest  $T^*$ , and use them to compute Euler equation errors, with little computational cost.

If there are many shocks, a large number of quadrature nodes are required to compute Euler equation errors, running the above accuracy measure method could be computationally expensive. Here we also provide a weaker but no-cost indicator

---

<sup>14</sup>For low-dimensional problems with smooth policy functions, we can also compute Euler equation errors by explicitly constructing policy function approximations as discussed in Section 3.6. This method’s computational cost is low, but it requires extra coding for approximating policy functions. More importantly, it would be challenging for problems with high dimensionality or non-smooth policy functions.

for checking accuracy: we use the Monte Carlo simulation idea to estimate the Euler equation error at the initial state only. Note that the initial state is given and fixed, so Algorithm 1 or 2 has already applied the Monte Carlo simulation to generate  $m$  simulated states in the second period and their corresponding decisions. That is, if we use the Monte Carlo quadrature rule, then we have already had the next-period states and decisions, so we can evaluate the Euler equation error at the initial state by simply taking the average across the simulated solutions for the second period. Following the Law of Large Numbers, the accuracy of the Euler equation error is proportional to  $1/\sqrt{m}$ . Our real business cycle examples show that the Euler error at the initial state is close to the Euler errors across other periods.<sup>15</sup>

## 4 Numerical Examples

We apply SCEQ to solve four dynamic stochastic problems. The first is a simple optimal growth problem for illustration purposes only. The second example shows that SCEQ can solve high-dimensional multi-country real business cycle problems with occasionally binding constraints. The third example solves nonstationary stochastic integrated assessment models with seven state variables (six of them are continuous) and occasionally binding constraints. The last example shows that SCEQ can solve stochastic competitive equilibrium problems like New Keynesian models with a zero lower bound. All the examples show that SCEQ can obtain an accuracy of 0.1-1%. For all examples, we use the GAMS programming language (McCarl et al., 2016) with the CONOPT optimization solver (Drud, 1994), and run the SCEQ code in parallel on a Mac Pro desktop computer with six cores.<sup>16</sup> It took minutes or at most hours to solving each example. To further illustrate the implementation of SCEQ, we also provide Matlab code with the fmincon optimization solver for the first simple example. The code files are available on <https://sites.google.com/site/yycai01/research>.

---

<sup>15</sup>In some rare cases, the Euler error in the first period may not be enough. See Appendix A.3 for an example.

<sup>16</sup>We run one GAMS procedure per core, while each procedure uses a different seed for generating different pseudo random number sequences used in SCEQ and its accuracy measures.

## 4.1 Application to Optimal Growth Model

To illustrate the SCEQ method, our first example is a simple optimal growth problem with stochastic discrete total factor productivity (TFP),  $\tilde{A}_t$ . We assume  $\tilde{A}_t = \theta_t A_t$ , where  $A_t$  is the deterministic trend, and  $\theta_t$  evolves according to the following stochastic process

$$\ln(\theta_{t+1}) = \rho \ln(\theta_t) + \sigma \varepsilon_{t+1} \quad (6)$$

where  $\varepsilon_t$  is a standard normal random variable,  $\rho = 0.95$ , and  $\sigma = 0.02$ . We solve the following optimal growth problem:

$$\begin{aligned} \max_{c_t} \quad & \mathbb{E} \left\{ \sum_{t=0}^{\infty} \beta^t u(c_t) \right\} \\ \text{s.t.} \quad & K_{t+1} = (1 - \delta)K_t + \theta_t A_t K_t^\alpha - c_t, \quad t = 0, 1, 2, \dots \end{aligned} \quad (7)$$

where  $K_t$  is capital and  $c_t$  is consumption at time  $t$ ,  $\beta = 0.96$  is the discount factor,  $\delta = 0.1$  is the depreciation rate,  $\alpha = 0.3$ , and  $u(c) = -c^{-1}$  is the utility function. The initial states are  $K_0 = 1$  and  $\theta_0 = 1$ . For simplicity, we assume  $A_t \equiv A := (1 - (1 - \delta)\beta)/(\alpha\beta)$ , so the non-stochastic steady state of capital is  $K_{ss} = 1$ .

Assume that we are interested in the solutions for the first 20 periods ( $T^* = 20$ ). Using the notation from Section 3,  $\mathbf{x} := (K, \theta)$  is the vector of state variables,  $\mathbf{a} := c$  is the decision variable, and the transition laws are

$$\begin{aligned} K_{t+1} &= g_t^K(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t) = (1 - \delta)K_t + \theta_t A K_t^\alpha - c_t \\ \theta_{t+1} &= g_t^\theta(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t) \end{aligned}$$

which can be written as  $\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$ , where  $\epsilon_t$  is standard normal and  $\mathbf{g}_t = (g_t^K, g_t^\theta)$  is a vector of two functions. Here we use  $\theta_{t+1} = g_t^\theta(\mathbf{x}_t, \mathbf{a}_t, \epsilon_t)$  to represent the transition law of  $\theta_t$ , (6).

### 4.1.1 Implementation of SCEQ

In the initialization step of SCEQ, we choose  $\Delta_s = 30$  and the “terminal” value function  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) = u(AK_{s+\Delta_s}^\alpha - \delta K_{s+\Delta_s})/(1 - \beta)$ . We assume that consumption after time  $s + \Delta_s$  is always  $AK_{s+\Delta_s}^\alpha - \delta K_{s+\Delta_s}$  so that capital after the “terminal” time

$s + \Delta_s$  is always the “terminal” capital  $K_{s+\Delta_s}$ , as long as  $\theta_t = 1.0$  for all  $t \geq s + \Delta_s$ .<sup>17</sup> In the optimization step of Algorithm 1 for solving problems at time  $s$ , the transition equation  $\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, 0)$  has  $\theta_t$  replaced by its median conditional on the realized value  $\theta_s^i$ , denoted  $(\theta_s^i)^{\rho^{t-s}}$ , for simulation path  $i$ . That is, the optimization step of Algorithm 1 solves the following problem:

$$\begin{aligned} \max_{c_t} \quad & \sum_{t=s}^{s+29} \beta^{t-s} u(c_t) + \beta^{30} u(AK_{s+30}^\alpha - \delta K_{s+30}) / (1 - \beta) \\ \text{s.t.} \quad & K_{t+1} = (1 - \delta)K_t + (\theta_s^i)^{\rho^{t-s}} AK_t^\alpha - c_t, \quad t = s, s + 1, \dots, s + 29, \end{aligned} \quad (8)$$

and the dynamic system starts with the simulated values  $(K_s^i, \theta_s^i)$  at time  $s$  for the  $i$ th simulation path. We solve the deterministic finite-horizon problem (8) using the CONOPT optimization solver to obtain the optimal consumption  $c_s^i$ .

The simulation step of Algorithm 1 uses  $(K_s^i, \theta_s^i, c_s^i)$  to simulate the state values at time  $s + 1$ :  $K_{s+1}^i = (1 - \delta)K_s^i + \theta_s^i A (K_s^i)^\alpha - c_s^i$  and  $\theta_{s+1}^i = (\theta_s^i)^\rho \exp(\sigma \epsilon_s^i)$ , where  $\epsilon_s^i$  is a realized value of  $\epsilon_s$  for the simulation path  $i$ . We generated 1,000 simulated paths for the periods of interest (the first 20 periods),<sup>18</sup> and the GAMS code took 4.2 minutes to run on a Mac Pro desktop computer.<sup>19</sup>

#### 4.1.2 Accuracy Measures

The normalized Euler error for this problem is

$$\left| 1 - \beta \mathbb{E} \left[ \frac{u'(c_{t+1})}{u'(c_t)} (1 - \delta + \theta_{t+1} A \alpha K_{t+1}^{\alpha-1}) \mid (K_t, \theta_t) \right] \right| \quad (9)$$

<sup>17</sup>We tried  $\Delta_s = 50$  and found that it has little impact on the solutions obtained with  $\Delta_s = 30$ .

<sup>18</sup>Since the problem is stationary, the solutions  $\{(K_s^i, \theta_s^i, c_s^i) : 0 \leq s < 20, 1 \leq i \leq 1000\}$  follow the same consumption policy function across time, so we can use them to construct our policy function approximation. Since the domain of simulated states is narrow, we can use a degree-3 complete Chebyshev polynomial of  $\ln(K)$  and  $\ln(\theta)$  to have a good approximation of the policy function, in which we use the least-squares fitting method to estimate the Chebyshev coefficients.

<sup>19</sup>If we do not implement parallelism on the six compute cores, then the runtime is 18 minutes, so the parallel efficiency is 71%. We also tried a parallel Matlab code, which was much slower than the GAMS code, taking 11 minutes to run on the same computer.

We can estimate it at the initial state by replacing the expectation with the average across 1,000 simulated solutions in the second period:

$$\left| 1 - \frac{\beta}{1000} \sum_{i=1}^{1000} \left[ \frac{u'(c_1^i)}{u'(c_0^i)} \left( 1 - \delta + \theta_1^i A \alpha (K_1^i)^{\alpha-1} \right) \mid (K_0 = 1, \theta_0 = 1) \right] \right| \quad (10)$$

The estimated normalized Euler error at the initial state is  $1.9 \times 10^{-4}$ . We also employ the accuracy measure method in Appendix A.1 and find that the  $\mathcal{L}^\infty$  Euler error over the 1,000 simulated paths is  $5.7 \times 10^{-4}$ , only slightly larger than the Euler error at the initial state.

### 4.1.3 Comparison with Value Function Iteration

We also solve this simple problem via value function iteration (VFI) to obtain the optimal policy function for consumption, using the 7-node Gauss-Hermite quadrature rule to compute the conditional expectation of the value function in the Bellman equation (Bellman, 1957):

$$V(\mathbf{x}_t) = \max_{c_t} u(c_t) + \beta \mathbf{E}_t \{V(\mathbf{x}_{t+1})\} \quad (11)$$

subject to the transition laws of the state variable vector  $\mathbf{x}_t = (K_t, \theta_t)$ .

To make sure that next period's  $\theta$  is inside the approximation domain of  $\theta$  for the value function for all Gauss-Hermite quadrature nodes, we have to choose it from the interval  $[0.223, 4.482]$ .<sup>20</sup> With this wide range for  $\theta$ , we have to choose a wide range for  $K$  ( $[0.04, 29.1]$ ) so that next period's capital will be inside the approximation domain of  $K$  for all possible values of  $\theta$ .<sup>21</sup> That is, the two-dimensional approximation domain for the state variable vector  $(K, \theta)$  is  $[0.04, 29.1] \times [0.223, 4.482]$ , much wider than the ranges of the simulated states ( $[0.76, 1.3] \times [0.81, 1.25]$ ).

This wide approximation domain requires a very high-degree approximation. For this specific example we can do a nonlinear change of variables so that the value function can be approximated with a lower degree approximation. Using  $\ln(K)$  and

<sup>20</sup>The range is proportional to  $\sigma/(1 - \rho)$ , so if  $\rho$  or  $\sigma$  is larger, then the range has to be wider. Moreover, if we use more Gauss-Hermite quadrature nodes, then the range will also be wider.

<sup>21</sup>If we use a narrower range for  $K$ , then next period's capital could be binding at its bounds, which creates kinks when approximating the value function, so the solution may be inaccurate (particularly for those states near the bounds). Moreover, the kinks might prevent the value function iteration process from converging.

$\ln(\theta)$  as the state variables, we approximate the value function as a degree-30 complete Chebyshev polynomial of  $\ln(K)$  and  $\ln(\theta)$  (see, e.g., Cai and Judd (2014)). We also approximate the corresponding consumption policy functions by a degree-30 complete Chebyshev polynomial of  $\ln(K)$  and  $\ln(\theta)$ , denoted  $\mathcal{C}^{\text{VFI}}(K, \theta)$ . The normalized  $\mathcal{L}^\infty$  Euler error of the solution  $\mathcal{C}^{\text{VFI}}(K, \theta)$ , defined as

$$\max_{K, \theta} \left| 1 - \beta \mathbb{E} \left[ \frac{u'(\mathcal{C}^{\text{VFI}}(K_+, \theta_+))}{u'(\mathcal{C}^{\text{VFI}}(K, \theta))} (1 - \delta + \theta_+ A \alpha K_+^{\alpha-1}) \mid (K, \theta) \right] \right|$$

is  $1.2 \times 10^{-4}$ , where  $\theta_+$  is next period's productivity shock conditional on the current period's  $\theta$ , and  $K_+$  is next period's capital:  $K_+ = (1 - \delta)K + \theta A K^\alpha - \mathcal{C}^{\text{VFI}}(K, \theta)$ . Thus, we see the VFI solution is accurate enough for checking the accuracy of our SCEQ solution.

For comparison with the VFI solution, we use the superscript ‘‘SCEQ’’ to denote the SCEQ solution. We then compare  $c_t^{i, \text{SCEQ}}$  and  $\mathcal{C}^{\text{VFI}}(K_t^{i, \text{SCEQ}}, A_t^{i, \text{SCEQ}})$  for all  $1 \leq i \leq 1000$  and  $t < 20$ . We find the  $\mathcal{L}^\infty$  relative error of the SCEQ solution, defined as

$$\max_{0 \leq t < 20, 1 \leq i \leq 1000} \frac{|c_t^{i, \text{SCEQ}} - \mathcal{C}^{\text{VFI}}(K_t^{i, \text{SCEQ}}, A_t^{i, \text{SCEQ}})|}{\mathcal{C}^{\text{VFI}}(K_t^{i, \text{SCEQ}}, A_t^{i, \text{SCEQ}})}$$

is 0.0016, and the  $\mathcal{L}^1$  relative error, defined as

$$\frac{1}{20000} \sum_{0 \leq t < 20, 1 \leq i \leq 1000} \frac{|c_t^{i, \text{SCEQ}} - \mathcal{C}^{\text{VFI}}(K_t^{i, \text{SCEQ}}, A_t^{i, \text{SCEQ}})|}{\mathcal{C}^{\text{VFI}}(K_t^{i, \text{SCEQ}}, A_t^{i, \text{SCEQ}})}$$

is  $8.0 \times 10^{-4}$ .<sup>22</sup> The relative errors are close to the Euler errors in Section 4.1.2, so SCEQ's own accuracy measures are also good for checking solution accuracy.

## 4.2 Application to Multi-Country Real Business Cycle Models

Den Haan, Judd, and Juillard (2011) introduce a multi-country real business cycle model. Here we apply SCEQ to solve its modified version. We assume that there are

---

<sup>22</sup>We view the VFI optimal consumption policy to be the ‘‘true’’ solution as there is no analytical solution for the optimal growth problem.

$N$  countries with a capital stock state vector  $K_t = (K_{t,1}, \dots, K_{t,N})$ , and the production function for the  $j$ th country at time  $t$  is

$$Y_{t,j} = \zeta_{t,j} A_{t,j} (K_{t,j})^\alpha (\ell_{t,j})^{1-\alpha}, \quad (12)$$

where  $\ell_{t,j}$  is labor supply,  $\alpha$  is the expenditure share of capital in production,  $A_{t,j}$  is the deterministic productivity trend, and  $\zeta_{t,j}$  is a country-specific productivity shock. The law of motion of capital is:

$$K_{t+1,j} = (1 - \delta)K_{t,j} + I_{t,j} \quad (13)$$

where  $I_{t,j}$  is investment and  $\delta$  is the depreciation rate of capital.

The utility function for the  $j$ th country is

$$u_j(c_{t,j}, \ell_{t,j}) = \frac{(c_{t,j})^{1-\frac{1}{\gamma_j}}}{1 - \frac{1}{\gamma_j}} - B_{t,j} \frac{(\ell_{t,j})^{1+\frac{1}{\eta_j}}}{1 + \frac{1}{\eta_j}} \quad (14)$$

where  $c_{t,j}$  is consumption,  $\gamma_j$  is the inter-temporal elasticity of substitution,  $\eta_j$  is the Frisch elasticity of labor supply, and  $B_{t,j}$  is the relative weight of consumption and leisure in the welfare function.

We solve the social planner's problem, where aggregate utility is defined as

$$U(c_t, \ell_t) = \sum_{j=1}^N \tau_j u_j(c_{t,j}, \ell_{t,j})$$

with  $c_t = (c_{t,1}, \dots, c_{t,N})$  and  $\ell_t = (\ell_{t,1}, \dots, \ell_{t,N})$ , where  $\tau_j$  are country-specific weights. The social planner has the following aggregate world resource constraint:

$$\sum_{j=1}^N (c_{t,j} + I_{t,j} + \Gamma_{t,j}) = \sum_{j=1}^N Y_{t,j}. \quad (15)$$

where

$$\Gamma_{t,j} \equiv \frac{\phi}{2} K_{t,j} \left( \frac{I_{t,j}}{K_{t,j}} - \delta \right)^2 \quad (16)$$

is an adjustment cost with  $\phi$  as the intensity of the friction. There is also a lower bound for investment:

$$I_{t,j} \geq I_{\min}, \quad \forall t, j \quad (17)$$



That is, the social planner solves

$$\max_{c, \ell, I} \mathbb{E} \left( \sum_{t=0}^{\infty} \beta^t U(c_t, \ell_t) \right) \quad (18)$$

subject to the transition law (13) and feasibility constraints (15) and (17), for each  $t$  and  $j$ , where  $\beta$  is the discount factor.

We set  $\beta = 0.99$ ,  $\alpha = 0.33$ ,  $\delta = 0.025$ , and  $\phi = 0.5$ . While SCEQ can also solve real business cycle problem with heterogeneous preferences, for convenience, we also let  $\gamma_j \equiv \gamma = 0.5$ ,  $\eta_j = \eta = 0.5$ ,  $\tau_j \equiv 1$ ,  $A_{t,j} \equiv A = (1 - (1 - \delta)\beta)/(\alpha\beta)$ , and  $B_{t,j} \equiv (1 - \alpha)A(A - \delta)^{-1/\gamma}$  so that the problem has a symmetric and stationary model specification and the non-stochastic steady state for each country is  $K_{ss} = 1$ , with associated decisions  $\ell_{ss} = 1$ ,  $c_{ss} = A - \delta$ , and  $I_{ss} = \delta$ .

The initial state for the  $j$ th country is set as

$$K_{0,j} = K_{\min} + (K_{\max} - K_{\min}) \frac{j - 1}{N - 1}$$

with  $K_{\min} = 0.1$  and  $K_{\max} = 10$  for  $j = 1, \dots, N$ . Note that we choose a wide range for the initial capital levels across countries, to more closely replicate real-world cross-country differences. In contrast, most other methods for high-dimensional problems solve around the steady state, assuming that all countries have similar levels of capital. We choose  $I_{\min} = 0.9I_{ss}$  so the inequality (17) will bind frequently.

Since our policy functions in this example have kinks, it will be challenging to apply other methods to solve our problem, as they will require a high-degree approximation or an adaptive sparse grids approximation. Here we apply SCEQ to solve three cases of our high-dimensional model with occasionally binding constraints and a wide-ranging state space.

#### 4.2.1 Case 1: Systematic Shock

Our first case assumes  $\zeta_{t,j} \equiv \zeta_t$  is independent of country  $j$  and instead is a systematic shock affecting all countries. We assume that  $\zeta_t$  is a Markov chain with three possible

values: 0.9, 1.0, and 1.1, and its transition probability matrix is

$$P = \begin{bmatrix} 0.8 & 0.2 & & \\ 0.2 & 0.6 & 0.2 & \\ & 0.2 & 0.8 & \end{bmatrix}. \quad (19)$$

For SCEQ, we choose  $\Delta_s = 50$  and the “terminal” value function  $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) = U(0.75AK_{s+\Delta_s}^\alpha, 1)/(1 - \beta)$ , where  $K_{s+\Delta_s}^\alpha = (K_{s+\Delta_s,1}^\alpha, \dots, K_{s+\Delta_s,N}^\alpha)$ . We assume that the labor supply after time  $s + \Delta_s$  is always 1, and that consumption after time  $s + \Delta_s$  is always 75% of the deterministic output at time  $s + \Delta_s$ , with  $\zeta_{s+\Delta_s} = 1.0$ . In the optimization step of Algorithm 1, we replace  $\zeta_t$  by its mean conditional on the realized values of  $\zeta_s$  for all  $t \geq s$ :

$$\mathbb{E}(\zeta_t \mid \zeta_s) = \varsigma \pi_{t,s},$$

where  $\varsigma = (0.9, 1.0, 1.1)$  is the vector of all possible values of  $\zeta_t$ , and  $\pi_{t,s}$  is a column vector representing the probability distribution conditional on the realized values of  $\zeta_s$ . If the realized value of  $\zeta_s$  is the  $k$ th element of  $\varsigma$ , we have  $\pi_{t,s} = P^{t-s}\pi_{s,s}$ , where  $\pi_{s,s}$  is a length-3 column vector with 1 for the  $k$ th element and 0 everywhere else. The optimization problem (2) thus becomes

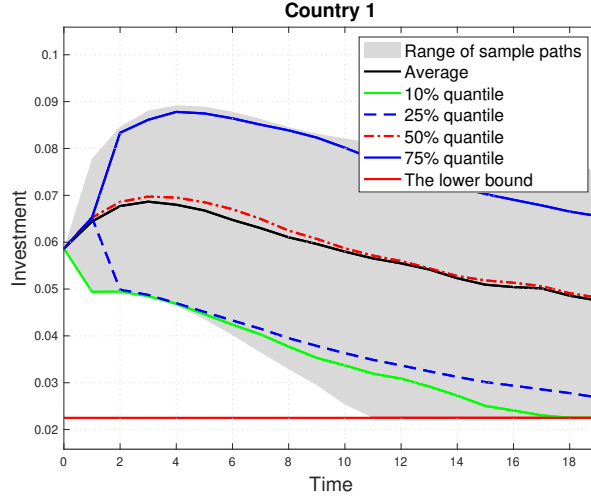
$$\begin{aligned} \max_{c,\ell,I} \quad & \sum_{t=s}^{s+49} \beta^t U(c_t, \ell_t) + \beta^{50} U(0.75AK_{s+50}^\alpha, 1)/(1 - \beta) \\ \text{s.t.} \quad & K_{t+1,j} = (1 - \delta)K_{t,j} + I_{t,j}, \\ & \sum_{j=1}^N (c_{t,j} + I_{t,j} + \Gamma_{t,j}) = \sum_{j=1}^N (\mathbb{E}(\zeta_t \mid \zeta_s^i) A(K_{t,j})^\alpha (\ell_{t,j})^{1-\alpha}), \\ & t = s, s + 1, \dots, s + 49, \end{aligned} \quad (20)$$

with the starting state  $K_s = K_s^i$  and  $\zeta_s = \zeta_s^i$  at time  $s$  for the  $i$ th simulation path.

We first solve the problem with  $N = 10$  countries and generate 1,000 simulated paths of the first 20 periods, which are assumed to be the periods of interest. It took 11 minutes on a Mac Pro desktop computer. Figure 1 displays the distributions of country 1’s optimal investments,  $I_{s,1}^i$ . From period 18 onwards, more than 10% of investments are binding at the lower bound  $I_{\min} = 0.09$ .<sup>23</sup> The  $\mathcal{L}^\infty$  Euler error on the

<sup>23</sup>The other countries have a greater percentage of binding investments because they have higher initial levels of capital. These countries have less incentive to invest as their capital levels are close to the non-stochastic steady state in the long run, especially if their initial capital levels are higher

Figure 1: Distribution of Investment for Country 1



simulated paths is 0.0069 and the  $\mathcal{L}^1$  Euler error is 0.0037, demonstrating that SCEQ works well for high-dimensional problems with kinks in the policy function. We can improve accuracy further by increasing  $\Delta_s$ , which reduces the truncation error. With  $\Delta_s = 100$ , the  $\mathcal{L}^\infty$  Euler error is 0.0024 and the  $\mathcal{L}^1$  Euler error is 0.0011, but running time increases to 19 minutes.

We also use SCEQ to solve problems with  $N = 20, 50, 100,$  and  $200$  countries, for  $\Delta_s = 50$ . Solutions are similar to the ones derived for  $N = 10$  countries. Table 1 reports the running times (in hours) of Algorithm 1 on a Mac Pro desktop computer, i.e., the time taken to generate 1,000 simulation paths of the first 20 periods and the Euler errors in  $\mathcal{L}^\infty$  and  $\mathcal{L}^1$  with  $\Delta_s = 50$ . The errors are almost identical to those for  $N = 10$ , demonstrating that SCEQ’s accuracy is independent of dimensionality, as SCEQ does not need to approximate value or policy functions. Even with  $N = 200$  countries, SCEQ just needs 16.1 hours to run on a Mac Pro desktop computer. Moreover, an increase from  $N$  countries to  $2N$  countries did not increase computational time exponentially, showing that the SCEQ’s computational time is nearly linear to the problem’s dimensionality.<sup>24</sup>

The last column of Table 1 also reports the Euler error at the initial state, which is computed with the average across the 1,000 simulated solutions for the second period.

---

than the steady state level. For example, the initial capital of country 10 is  $K_{\max} = 10$ , much larger than the steady state  $K_{ss} = 1$ , so its level of investment is always binding at  $I_{\min}$  in the initial periods, until its capital is close to the steady state level.

<sup>24</sup>As with  $N = 10$ , the solutions’ accuracy for larger  $N$  can also be improved by setting  $\Delta_s = 100$  to reduce truncation error, though it increases the computational time.

Table 1: Running Times and Errors for Case 1

$N$	Time (in hours) for SCEQ	Euler Error		
		$\mathcal{L}^\infty$	$\mathcal{L}^1$	Initial State
10	0.18	6.9(-3)	3.7(-3)	6.7(-3)
20	0.37	7.1(-3)	3.7(-3)	6.9(-3)
50	1.1	7.1(-3)	3.7(-3)	7.1(-3)
100	3.4	7.1(-3)	3.7(-3)	7.1(-3)
200	16.1	7.1(-3)	3.7(-3)	6.8(-3)

Note:  $a(-n)$  means  $a \times 10^{-n}$ .

The Euler errors at the initial state are close to those in  $\mathcal{L}^\infty$  across all simulated states, because the largest error usually happens at the most extreme states and the first period has the widest range of initial capital levels across countries.<sup>25</sup> Thus we can also use the Euler errors at the initial state to measure accuracy, as computing them has almost no additional cost.

#### 4.2.2 Case 2: Irreversible Risk

Sometimes a risk is irreversible, meaning that a shock leads to a permanent change to the system. Irreversible risks often have a significant impact on decisions, so it might seem like SCEQ would not be a suitable method, but here we show that SCEQ can solve these problems with a high level of accuracy and only take several minutes of computational time on a standard computer, much faster than Case 1 where the risk was reversible.<sup>26</sup>

Our second case assumes that  $\zeta_{t,j} \equiv \zeta_t$  is a systematic shock independent of country  $j$ , described as a Markov chain with two possible values:  $\varsigma_1 = 1.0$  and  $\varsigma_2 = 0.95$  and the following transition probability matrix:

$$P = \begin{bmatrix} 0.99 & 0 \\ 0.01 & 1 \end{bmatrix}. \quad (21)$$

The initial state is  $\zeta_0 = 1.0$ . This shock represents a global risk that results in a

<sup>25</sup>The Euler errors at the initial state could be slightly larger than those in  $\mathcal{L}^\infty$  across all simulated states, because they use different methods of estimation: Euler errors at the initial state have a standard error when using the average across the simulated solutions at the second period to estimate the expectation.

<sup>26</sup>Appendix A.3 shows that SCEQ can also solve problems with an irreversible risk and endogenous probabilities.

permanent 5% damage to economic output, with a 1% probability of occurrence in each period (the shock can only occur once). This problem is nonstationary as  $\zeta_t$  will converge to 0.95. For this specific example, there are only  $T^*$  different paths within the time interval of interest  $T^*$ : the first path has  $\zeta_t = 1.0$  for all  $0 \leq t < T^*$ , the second path assumes that the shock happens in the second period, i.e.,  $\zeta_0 = 1.0$  and  $\zeta_1 = \dots = \zeta_{T^*-1} = 0.95$ , and so on, with the last path having  $\zeta_0 = \dots = \zeta_{T^*-2} = 1.0$  and  $\zeta_{T^*-1} = 0.95$ . Thus, we can solve all different paths using SCEQ. Since the stochastic problem becomes deterministic once the shock happens, we just need to solve  $2T^* - 1$  optimization problems characterized by (2), i.e., solve problem (20) with

$$\mathbb{E}(\zeta_t | \zeta_s^i) = \begin{cases} P_{11}^{t-s} \varsigma_1 + (1 - P_{11}^{t-s}) \varsigma_2, & \text{if } \zeta_s^i = \varsigma_1 \\ \varsigma_2, & \text{otherwise} \end{cases}$$

where  $P_{11} = 0.99$  is element (1, 1) of the transition probability matrix  $P$ .

With  $T^* = 20$  and  $\Delta_s = 50$ , SCEQ took only seconds for  $N \leq 50$ , 3 minutes for  $N = 100$ , and 17 minutes for  $N = 200$  on a Mac Pro desktop computer without using parallelism. The solutions show that investments bind frequently, and the Euler error in  $\mathcal{L}^\infty$  is around 0.0068 for each  $N$ , close to those from Case 1.

### 4.2.3 Case 3: Country-Specific Shocks

The last case assumes that every country has a country-specific shock  $\zeta_{t,j}$ , which is also correlated with a systematic shock affecting all countries ( $\varepsilon_{t+1}$ ). We assume  $\zeta_{t,j}$  is a continuous exogenous state variable, following the stochastic process

$$\ln(\zeta_{t+1,j}) = \rho \ln(\zeta_{t,j}) + \sigma_1 \epsilon_{t+1,j} + \sigma_2 \varepsilon_{t+1}$$

for each  $j = 1, \dots, N$ , where  $\epsilon_{t+1,j}, \varepsilon_{t+1} \sim i.i.d. \mathcal{N}(0, 1)$  (i.e.,  $\epsilon_{t+1,j}$  and  $\varepsilon_{t+1}$  are independent and identical standard normal distributions across time and countries),  $\rho = 0.95$ , and  $\sigma_1 = \sigma_2 = 0.01$ . Thus, an  $N$ -country real business cycle model has  $2N$  state variables, of which  $N$  are exogenous, while Cases 1 and 2 have  $N$  endogenous state variables and only one exogenous discrete state variable.

In the optimization step of Algorithm 1, we still solve (20) but replace  $\mathbb{E}(\zeta_t | \zeta_s^i)$  with  $(\zeta_{s,j}^i)^{\rho^{t-s}}$ , the median of  $\zeta_{t,j}$  at time  $t$  conditional on the realized value  $\zeta_{s,j}^i$  at time  $s$ , with the starting state  $K_s = K_s^i$  and  $\zeta_{s,j} = \zeta_{s,j}^i$  at time  $s$  for the  $i$ th simulation

Table 2: Running Times and Errors for Case 3

$N$	Time (in hours) for SCEQ	Euler Error at the Initial State
10	0.18	$6.7(-3)$
20	0.33	$6.8(-3)$
50	0.97	$7.1(-3)$
100	3.3	$7.3(-3)$
200	16.4	$7.2(-3)$

Note:  $a(-n)$  means  $a \times 10^{-n}$ .

path. We generate 1,000 simulated paths of the first 20 periods. The solutions show that investments bind frequently an the lower bound  $I_{\min}$  for each country.

Table 2 reports the running times (in hours) and Euler errors at the initial state, for the number of countries  $N = 10, 20, 50, 100,$  and  $200$  with  $\Delta_s = 50$ . We see that SCEQ solves the problems within minutes or hours, close to the computational times in Case 1, although their dimensions are nearly double those in Case 1. Moreover, the Euler errors at the initial state are close to those in Case 1. This example shows that SCEQ’s efficiency is independent of the number of exogenous state variables, because exogenous state variables are replaced by their certainty equivalent approximation (e.g. mean or median) in SCEQ so have almost no impact on SCEQ’s computational time or accuracy.

### 4.3 Application to DSICE

Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) solve a dynamic stochastic integrated model of climate and economy (DSICE) that has economic and climate risks. DSICE is a DSGE extension of DICE (Nordhaus, 2008, 2017), which has exogenous paths for the population, TFP, land emissions, abatement cost, carbon intensity, and exogenous radiative forcing. Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) employ Epstein-Zin preferences (Epstein and Zin, 1989), long-run economic risk (Bansal and Yaron, 2004), and climate tipping risks (Lenton et al., 2008) with endogenous tipping probabilities, uncertain duration, and uncertain damage. Here we apply SCEQ to solve a simpler version of DSICE, in which we follow the same deterministic economic and climate systems, but we assume a simple economic risk and do not use Epstein-Zin preferences.<sup>27</sup> All exogenous paths and parameter

<sup>27</sup>Appendix A.3 shows that SCEQ can solve another simpler version of DSICE with climate tipping

values follow Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019), except those specified below.

### 4.3.1 Model Overview

We briefly describe the deterministic version of DSICE. Let  $\mathbf{M}_t = (M_{\text{AT},t}, M_{\text{UO},t}, M_{\text{LO},t})^\top$  be the carbon concentrations in the atmosphere, and upper and lower levels of the ocean respectively. These concentrations evolve over time according to:

$$\mathbf{M}_{t+1} = \Phi_{\text{M}}\mathbf{M}_t + (E_t, 0, 0)^\top, \quad (22)$$

where  $\Phi_{\text{M}}$  is a linear transition matrix, and  $E_t = E_{\text{Ind},t} + E_{\text{Land},t}$  is the annual total carbon emissions, where  $E_{\text{Ind},t}$  is industrial emissions and  $E_{\text{Land},t}$  is exogenous land emissions. Let  $\mathbf{T}_t = (T_{\text{AT},t}, T_{\text{OC},t})^\top$  be temperature anomalies of the atmosphere and ocean, following the law of motion

$$\mathbf{T}_{t+1} = \Phi_{\text{T}}\mathbf{T}_t + (\xi_1 F_t, 0)^\top, \quad (23)$$

where  $\Phi_{\text{T}}$  is a linear transition matrix and  $\xi_1$  is a parameter.  $F_t = \eta \log_2(M_{\text{AT},t}/M_{\text{AT}}^*) + F_{\text{EX},t}$  is global radiative forcing, where  $\eta$  is a parameter,  $M_{\text{AT}}^*$  is the pre-industrial atmospheric carbon concentration, and  $F_{\text{EX},t}$  is the exogenous radiative forcing.

The economic system has a state variable, capital ( $K_t$ ), which is used to define gross economic output  $Y_t = A_t K_t^\alpha L_t^{1-\alpha}$ , where  $\alpha$  is a parameter,  $A_t$  is exogenous TFP, and  $L_t$  is the exogenous global population size at time  $t$ . Output is reduced by the temperature anomaly according to the damage factor

$$\Omega(T_{\text{AT},t}) = \frac{1}{1 + \pi_1 T_{\text{AT},t} + \pi_2 (T_{\text{AT},t})^2},$$

where  $\pi_1$  and  $\pi_2$  are parameters. Economic production has industrial emissions  $E_{\text{Ind},t} = \sigma_t(1 - \mu_t)Y_t$ , which is proportional to gross output but can be reduced by mitigation, measured by the emission control rate  $\mu_t \in [0, 1]$ , and  $\sigma_t$  is exogenous carbon intensity. The mitigation cost is  $\Psi_t = \theta_{1,t}\mu_t^{\theta_2}Y_t$ , where  $\theta_{1,t}$  is the exogenous

---

risks and endogenous tipping probabilities. It took only two minutes on a single compute core and gave an acceptable solution accuracy.

abatement cost and  $\theta_2$  is a parameter. Thus the transition law of capital is

$$K_{t+1} = (1 - \delta)K_t + \Omega(T_{AT,t})Y_t - C_t - \Psi_t, \quad (24)$$

where  $\delta$  is the depreciation rate and  $C_t$  is consumption. Note that the emission control rate  $\mu_t$  may be binding at its upper bound, so this problem has an occasionally binding constraint.

The deterministic model solves the following social planner's problem

$$\max_{C_t, \mu_t} \sum_{t=0}^{\infty} \beta^t u(C_t/L_t)L_t \quad (25)$$

subject to the transition laws (22)-(24) of six state variables:  $\mathbf{M}_t$ ,  $\mathbf{T}_t$ , and  $K_t$ , where  $C_t$  and  $\mu_t$  are decision variables,  $\beta$  is the discount factor, and  $u$  is a power utility function  $u(c) = c^{1-\gamma}/(1-\gamma)$  with  $\gamma = 1.45$  denoting the elasticity of marginal utility of per capita consumption  $c$  as in DICE-2016 (Nordhaus, 2017). The initial states are observed and given.

Now we add a simple economic risk. We assume that TFP,  $\tilde{A}_t$ , is stochastic:  $\tilde{A}_t = \zeta_t A_t$ , where  $A_t$  is the deterministic trend and  $\zeta_t$  is a Markov chain representing a productivity shock. For simplicity, we assume that  $\zeta_t$  follows the same distribution and transition probabilities as in Section 4.2, that is,  $\zeta_t$  has three possible values: 0.9, 1.0, and 1.1, and its transition probability matrix is given by (19). Gross economic output is  $Y_t = \tilde{A}_t K_t^\alpha L_t^{1-\alpha}$ , and we solve the social planner's problem

$$\max_{C_t, \mu_t} \mathbb{E} \left\{ \sum_{t=0}^{\infty} \beta^t u(C_t/L_t)L_t \right\} \quad (26)$$

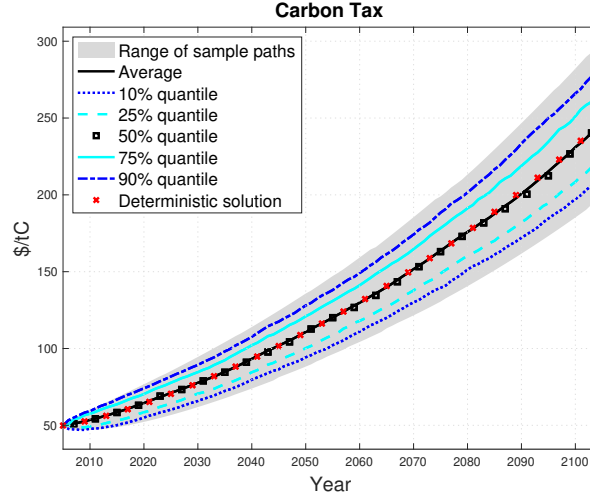
subject to the transition laws (22)-(24) and the Markov chain of  $\zeta_t$ .

### 4.3.2 Implementation of SCEQ

Because DSICE is a nonstationary stochastic problem with six endogenous continuous state variables, one exogenous discrete state variable, and occasionally binding constraints, it is challenging to solve it using standard methods like VFI unless we can choose the appropriate approximation methods and time-varying approximation domains as in Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019). Here we



Figure 2: Carbon Tax for DSICE with Economic Risk



can apply SCEQ to overcome these challenges.<sup>28</sup> As in Section 4.1, in the optimization step of Algorithm 1 we replace  $\zeta_t$  by its mean conditional on the realized value of  $\zeta_s$  for all  $t \geq s$ .

Figure 2 displays distributions of the optimal carbon taxes in the first  $T^* = 100$  years from 1,000 simulation paths obtained by SCEQ, which took 1.8 hours to run on a Mac Pro desktop computer. The economic risk has little impact on the initial carbon tax, and the average or median path is almost identical to the deterministic solution. These results are different from those in the stochastic growth benchmark example of Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019), but Figure 2 still shows that the optimal carbon tax is stochastic and has a wide range in 2100: \$185 to \$282 per ton of carbon, although the range in Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) is much wider. The differences from the results of Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019) arise because here we do not use long-run economic risk or Epstein-Zin preferences.

<sup>28</sup>We follow DICE by truncating the infinite-horizon problem to a 600-year problem with the terminal value function being zero everywhere, as reasonable terminal conditions at the 600th year have little impact on the solution in the first 100 years — the time of interest — due to the small compound discount factor and a small magnitude of utility in the long run (as consumption will be large and the elasticity of marginal utility  $\gamma > 1$ ). In SCEQ, we let  $\Delta_s = 600 - s$  and the terminal value function is  $V_{600}(\mathbf{x}_{600}) \equiv 0$ .

### 4.3.3 Accuracy Measures

We compute the Euler errors, specified in Appendix A.2, at the simulated states using the accuracy measure method in Appendix A.1. We find that the  $\mathcal{L}^\infty$  Euler error is 0.0011 and the  $\mathcal{L}^1$  Euler error is  $1.1 \times 10^{-4}$ . We also compare the SCEQ solution with the solution obtained from VFI for the first 100 years, as in Cai, Judd, and Lontzek (2017) and Cai and Lontzek (2019). We use time-varying approximation domains and degree-6 complete Chebyshev polynomials for approximating value functions in VFI. As in Section 4.1, we use the policy functions from VFI to compute the decisions at the realized SCEQ simulated states, then compare them with the corresponding SCEQ simulated decisions. For the optimal carbon taxes in the first 100 years, the relative  $\mathcal{L}^\infty$  error is 0.023 and the relative  $\mathcal{L}^1$  error is 0.0031, if we treat the VFI solution as the “true” solution.<sup>29</sup> In addition, the relative  $\mathcal{L}^\infty$  error is 0.0028 and 0.013 for  $C_t$  and  $\mu_t$  respectively, and the relative  $\mathcal{L}^1$  error is 0.0024 and 0.0017 for  $C_t$  and  $\mu_t$  respectively.<sup>30</sup>

## 4.4 Application to a New Keynesian Model with Zero Lower Bound

We have shown that with Algorithm 1, SCEQ can solve stationary or nonstationary stochastic dynamic programming problems with high dimensionality and occasionally binding constraints. Here we show that using Algorithm 2, SCEQ can also solve stochastic competitive equilibrium problems with occasionally binding constraints. Our example uses the New Keynesian model with a zero lower bound (ZLB), from Guerrieri and Iacoviello (2015) and Cai, Judd, and Steinbuks (2017).<sup>31</sup>

---

<sup>29</sup>The VFI solutions only have an accuracy of 0.1-1% because we use degree-6 complete Chebyshev polynomial approximations, so the relative errors are biased. Obtaining an additional digit of accuracy from VFI will require a much higher degree approximation, making it too time consuming to run it on a desktop computer. For instance, we use a degree-30 complete Chebyshev polynomial approximation to obtain the  $\mathcal{L}^\infty$  Euler error  $1.2 \times 10^{-4}$  for the VFI solution of the example in Section 4.1, while it only requires a degree-10 complete Chebyshev polynomial approximation to obtain the  $\mathcal{L}^\infty$  Euler error  $6.4 \times 10^{-4}$ , or a degree-20 complete Chebyshev polynomial approximation to obtain the  $\mathcal{L}^\infty$  Euler error  $3.9 \times 10^{-4}$ .

<sup>30</sup>Note that the carbon tax is  $1000\theta_{1,t}\theta_2\mu_t^{\theta_2-1}/\sigma_t$ , so the relative error of  $\mu_t$  is amplified when computing the relative error of carbon taxes.

<sup>31</sup>New Keynesian DSGE models have been studied frequently in the literature, see e.g., Woodford (2003); Negro et al. (2007); Smets and Wouters (2007); Gali (2008); Fernandez-Villaverde et al. (2015); Maliar and Maliar (2015).

#### 4.4.1 Model Overview

Since we use the exact same New Keynesian model as in Cai, Judd, and Steinbuks (2017), we only give a brief description here. The model consists of a representative household, a government, a final-good firm, and intermediate firms. The government consumes a fraction  $s_g$  of the final good and issues bonds every period with a nominal interest rate  $r_t$  which has a zero lower bound. The final-good firm purchases intermediate goods from intermediate firms to produce the final good  $y_t$  and sell it at the price  $p_t$ . The intermediate firms are assumed to have Calvo-type prices for the intermediate goods: in each period a fraction  $1 - \theta$  of the firms have optimal prices and the remaining fraction keep the same price as the previous period. At each period  $t$ , the representative household consumes the remaining fraction  $1 - s_g$  of the final good, buys newly issued bonds, sells the expired bonds, earns wages from labor supply, receives a lump-sum transfer from the government, and receives profits from all firms. The representative household wants to maximize the present value of expected utilities subject to a budget constraint. The discount factor  $\beta_t$  is the stochastic process

$$\ln(\beta_{t+1}) = (1 - \rho) \ln(\beta^*) + \rho \ln(\beta_t) + \sigma \epsilon_{t+1} \quad (27)$$

where  $\epsilon_t \sim i.i.d. \mathcal{N}(0, 1)$ , and  $\beta^*$  is the non-stochastic steady state discount factor.

The New Keynesian model has one endogenous state variable ( $v_t$ ) that represents price dispersion:

$$v_{t+1} = (1 - \theta)q_t^{-\alpha} + \theta\pi_t^\alpha v_t \quad (28)$$

where  $\pi_t \equiv p_t/p_{t-1}$  is the gross inflation rate, and

$$q_t = \left( \frac{1 - \theta\pi_t^{\alpha-1}}{1 - \theta} \right)^{\frac{1}{1-\alpha}} \quad (29)$$

where  $\alpha$  is a parameter in the production function of the final-good firm. Appendix A.5 derives the following equilibrium equations:

$$1 = \frac{1}{\chi_{t,1}} \left( y_t^{1+\eta} v_{t+1} + \theta \mathbb{E}_t \{ \beta_{t+1} \pi_{t+1}^\alpha \chi_{t+1,1} \} \right) \quad (30)$$

$$1 = \frac{1}{\chi_{t,2}} \left( \frac{1}{1-s_g} + \theta \mathbb{E}_t \{ \beta_{t+1} \pi_{t+1}^{\alpha-1} \chi_{t+1,2} \} \right) \quad (31)$$

$$q_t = \frac{\alpha \chi_{t,1}}{(\alpha-1) \chi_{t,2}} \quad (32)$$

$$1 = \mathbb{E}_t \left\{ \beta_{t+1} \frac{1+r_t}{\pi_{t+1}} \frac{y_t}{y_{t+1}} \right\} \quad (33)$$

$$z_t = (1+r^*) \left( \frac{\pi_t}{\pi^*} \right)^{\phi_\pi} \left( \frac{y_t}{y^*} \right)^{\phi_y} - 1 \quad (34)$$

$$r_t = \max(z_t, 0) \quad (35)$$

where  $\chi_{t,1}$  and  $\chi_{t,2}$  are defined in Appendix A.5, and  $\pi^*$ ,  $r^*$ , and  $y^*$  are the steady-state gross level of inflation, nominal interest rate, and output, respectively.

We apply Algorithm 2 to solve this stochastic competitive equilibrium problem. The state vector is  $\mathbf{x}_t = (\beta_t, v_t)$ , with the transition laws (27) and (28). The variables  $\chi_{t,1}$ ,  $\chi_{t,2}$ , and  $y_t$  are viewed as action variables  $\mathbf{a}_t$ , while the other variables  $q_t$ ,  $\pi_t$ ,  $z_t$  and  $r_t$  can be substituted by expressions of  $\chi_{t,1}$ ,  $\chi_{t,2}$ , and  $y_t$ , according to (29), (32), (34), and (35). Except the transition laws, the equilibrium conditions are (29)-(35). The initial states are  $\beta_0 = \beta^*$  and  $v_0 = v^*$ , where  $v^*$  is the steady state price dispersion. We choose  $\Delta_s = 200$  and let the ‘‘terminal’’ decision rule  $\mathbf{a}_{s+\Delta_s}^*(\mathbf{x}_{s+\Delta_s})$  be given as  $\chi_{s+\Delta_s,1} \equiv \chi_1^*$ ,  $\chi_{s+\Delta_s,2} \equiv \chi_2^*$ , and  $y_{s+\Delta_s} = y^*$ , where  $\chi_1^*$ ,  $\chi_2^*$ , and  $y^*$  are the steady state values of  $\chi_{t,1}$ ,  $\chi_{t,2}$ , and  $y_t$  respectively. In the optimization step of Algorithm 2, we replace the expectations in (30), (31), and (33) by their median conditional on the realized values of  $\beta_s$ . That is, (30), (31), and (33) are defined as

$$1 = \frac{1}{\chi_{t,1}} \left( y_t^{1+\eta} v_{t+1} + \theta \tilde{\beta}_{t+1} \pi_{t+1}^\alpha \chi_{t+1,1} \right) \quad (36)$$

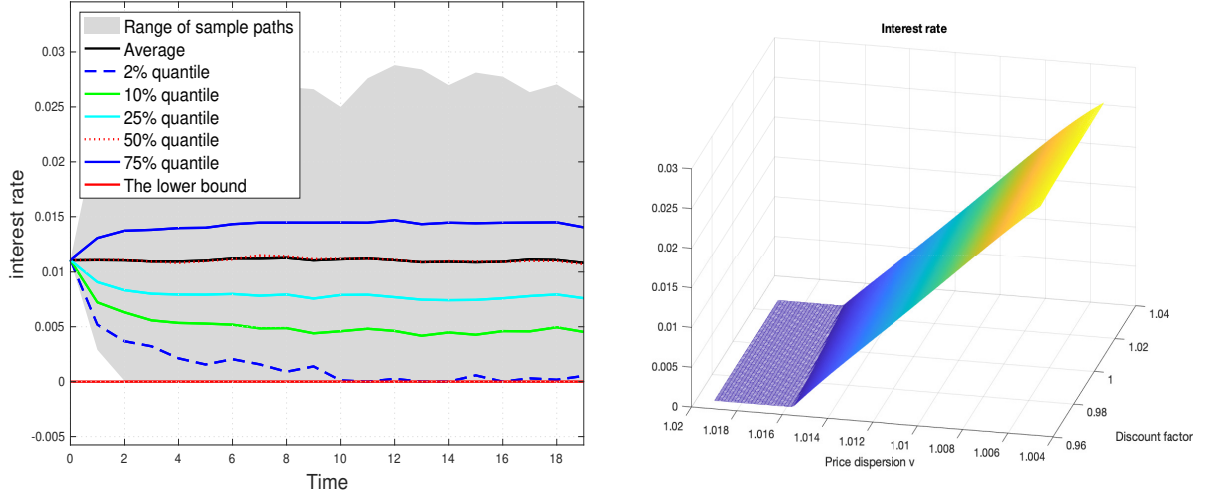
$$1 = \frac{1}{\chi_{t,2}} \left( \frac{1}{1-s_g} + \theta \tilde{\beta}_{t+1} \pi_{t+1}^{\alpha-1} \chi_{t+1,2} \right) \quad (37)$$

$$1 = \tilde{\beta}_{t+1} \frac{1 + \max(z_t, 0)}{\pi_{t+1}} \frac{y_t}{y_{t+1}} \quad (38)$$

where

$$\tilde{\beta}_{t+1} = \exp \left( (1-\rho) \ln(\beta^*) + \rho \ln(\tilde{\beta}_t) \right)$$

Figure 3: Simulated Interest Rates and the Policy Function for Interest Rate



for  $t = s, s + 1, \dots, s + \Delta_s - 1$ , and  $\tilde{\beta}_s = \beta_s^i$  is the realized value of  $\beta_s$  in the  $i$ th simulation path.

We assume that the time of interest,  $T^*$ , is 20 periods. We generated 1,000 simulation paths using Algorithm 2, which only took 4 minutes on a Mac Pro desktop computer. Using the simulated solutions, we also construct a policy function approximation for  $z_t$  using a degree-4 complete Chebyshev polynomial  $Z(\beta, v)$ , and then determine the interest rate as  $r = \max(Z(\beta, v), 0)$ . Figure 3 displays the distribution of interest rates (the left panel) and the policy function approximation for the interest rate (the right panel), showing that some interest rates are binding at zero. The Euler error over the simulated solutions is 0.0032 in  $\mathcal{L}^\infty$ . This example shows that SCEQ can solve a stochastic competitive equilibrium problem with occasionally binding constraints.

## 5 Discussion

If utility  $u_t$  is a common power function with a relative risk aversion parameter  $\gamma$ , then  $\gamma$  plays two roles: risk aversion and the inverse of the intertemporal elasticity of substitution (IES) for time-separable power utilities. The role of the risk aversion parameter disappears in the certainty equivalent approximation model (2), but the inverse of IES,  $1/\gamma$ , still affects the solution to the deterministic model (2). For this reason, the SCEQ method cannot work for dynamic portfolio problems, where the

degree of risk aversion is important for risky portfolio choices, dynamic stochastic problems with Epstein–Zin preferences in which the risk aversion and the IES are separated (e.g., Cai, Judd, and Lontzek (2017); Cai and Lontzek (2019)), or static problems, where  $\gamma$  only represents risk aversion. However, for many dynamic stochastic problems with time-separable power utilities, the role of the inverse of IES often dominates the role of risk aversion in affecting the solutions, so the SCEQ method can solve them accurately.

When we solve a dynamic stochastic problem, the first step is to choose the computational method. Unfortunately we cannot know which method can work or is more efficient *a priori*. It will be frustrating to spend an enormous amount of time and resources to try one method but finally discover it does not work well. For example, before checking Euler errors, NLCEQ has to first solve all optimization problems corresponding to the approximation nodes, VFI or time iteration has to wait until it converges before obtaining optimal value or policy functions, and GSSA and the EDS method have to run a large number of simulated paths to approximate policy functions.

With SCEQ, we can apply the accuracy measure method to do an early and fast check with only a small amount of computational time and resources. For example, we can just assume the time of interest to be  $T^* = 2$  and obtain  $m$  simulated paths with only two periods using Algorithm 1 or 2, then compute the Euler equation error at the initial state. If it is not small (with a large  $\Delta_s$ ), then we switch to other solution methods. That is, a small Euler equation error at the initial state is a necessary condition for choosing SCEQ. Alternatively, we can just solve only one or several simulated paths (i.e., choosing a small  $m$ ), and compute the Euler equations errors along the small number of simulated paths. If they are not small (with a large  $\Delta_s$ ), then we switch to other solution methods. These two methods can also be applied to check if the SCEQ code has bugs or  $\Delta_s$  is large enough: we suggest using the same code but changing the variances of the random variables  $\epsilon_t$  to be nearly zero.<sup>32</sup> Since SCEQ solves deterministic models very accurately, if the computed Euler equation errors for the nearly-zero-variance case are not nearly zero, it implies that there are bugs in the code or that  $\Delta_s$  is not large enough (i.e., the truncation error is large).

---

<sup>32</sup>If the problem is stationary and the initial state is equal or close to the non-stochastic steady state, then we suggest changing the initial state for debugging. Otherwise, with the nearly zero variances and some terminal conditions, the simulated states could be always close to the non-stochastic steady state so then the Euler equation errors would be small.

Thus, we can know almost *a priori* if SCEQ works for a specific problem.

For some special cases, such as problems with an irreversible risk (Section 4.2.2), the number of all different possible paths within a timeframe of interest  $T^*$  may be smaller than  $m$ , the number of (different) simulated paths. In such cases, we can compute the solutions along all different paths in the time of interest and use them to generate  $m$  simulation paths with little computational cost. For example, in Section 4.2.2, the number of all different possible paths in the first 20 periods is just 19, so we solve them all and use these 19 paths to simulate 10,000 paths with little additional cost. Thus, for that example, SCEQ took only minutes on a Mac Pro desktop computer to solve a DSGE with 201 state variables and occasionally binding constraints, much faster than NLCEQ or other methods like VFI, even if they use sparse grid methods. But for general problems, the number of all different possible paths within a timeframe of interest is often much larger than  $m$ , which is often chosen to be 1,000 or 10,000. For example, if there is a binary stochastic state variable and none of its transition probabilities are zero, then there are  $2^{20} \approx 10^6$  different paths in the first 20 periods.

Some rare cases may require high-accuracy solutions that SCEQ cannot provide, but we can still use SCEQ to provide (time-varying) approximation domains and initial guesses for other methods that may have higher accuracy (and much higher computational costs). For example, in the simulation step of SCEQ, we can always choose the worst scenario or the best scenario to obtain a lower or upper bound of states for constructing (time-varying) approximation domains, which can then be used for other methods.

## 6 Conclusions

This paper introduces a novel computational method, SCEQ, for solving dynamic stochastic problems. We have shown that SCEQ can be much more efficient and stable than other common computational methods such as NLCEQ and VFI, while retaining a solution accuracy of 0.1-1%. We have also shown that SCEQ can solve problems that other existing methods cannot, such as problems with both high dimensionality and occasionally binding constraints.

SCEQ is simple but powerful. It avoids complicated computational techniques for approximation and integration to make it as simple as perturbation methods, but it

still provides globally valid solutions while perturbation methods provide only locally valid solutions. Using a standard computer and an efficient optimization solver, SCEQ is stable and can provide accurate solutions for many dynamic stochastic problems with high dimensionality, occasionally binding constraints, nonstationarity, and/or a wide range of state space, without using an extensive amount of resources. Moreover, SCEQ does not suffer from the curse of dimensionality. SCEQ provides distributions of solutions which can be used to compute stochastic properties, such as means, variances, covariances, and trends, so it can have extensive use in economic analysis. For example, structural estimation such as the GMM method can apply SCEQ to estimate structural parameters.

## References

- Adjemian, Stéphane, Houtan Bastani, Michel Juillard, Frédéric Karamé, Junior Maih, Ferhat Mihoubi, George Perendia, Johannes Pfeifer, Marco Ratto, and Sébastien Villemot. 2011. “Dynare: Reference Manual Version 4.” Dynare Working Papers 1, CEPREMAP. URL <https://www.dynare.org>.
- Aruoba, S. Boragan, Jesus Fernandez-Villaverde, and Juan F. Rubio-Ramirez. 2006. “Comparing solution methods for dynamic equilibrium economies.” *Journal of Economic Dynamics and Control* 30 (12):2477–2508.
- Bansal, Ravi and Amir Yaron. 2004. “Risks for the Long Run: A Potential Resolution of Asset Pricing Puzzles.” *The Journal of Finance* 59 (4):1481–1509.
- Bellman, Richard. 1957. *Dynamic Programming*. Princeton University Press.
- Bertsekas, D. 2005. *Dynamic Programming and Optimal Control*, vol. I. Athena Scientific, Nashua.
- . 2007. *Dynamic Programming and Optimal Control*, vol. II. Nashua: Athena Scientific.
- Blanchard, Olivier Jean and Charles M. Kahn. 1980. “The Solution of Linear Difference Models under Rational Expectations.” *Econometrica* 48 (5):1305–1311.
- Brumm, Johannes and Simon Scheidegger. 2017. “Using Adaptive Sparse Grids to Solve High-Dimensional Dynamic Models.” *Econometrica* 85 (5):1575–1612.



- Cai, Yongyang. 2019. “Computational methods in environmental and resource economics.” *Annual Review of Resource Economics* 11:59–82.
- Cai, Yongyang and Kenneth L. Judd. 2013. “Shape-preserving Dynamic Programming.” *Mathematical Methods of Operations Research* 77 (3):407–421.
- . 2014. “Advances in Numerical Dynamic Programming and New Applications.” In *Handbook of Computational Economics*, vol. 3, edited by Kenneth L. Judd and Karl Schmedders. Amsterdam: Elsevier, 479–516.
- Cai, Yongyang, Kenneth L. Judd, and Thomas S. Lontzek. 2017. “The social cost of carbon with economic and climate risks.” Hoover economics working paper 18113. URL <https://www.hoover.org/research/social-cost-carbon-economic-and-climate-risk>.
- Cai, Yongyang, Kenneth L. Judd, and Jevgenijs Steinbuks. 2017. “A Nonlinear Certainty Equivalent Approximation Method for Dynamic Stochastic Problems.” *Quantitative Economics* 8 (1):117–147.
- Cai, Yongyang and Thomas S. Lontzek. 2019. “The social cost of carbon with economic and climate risks.” *Journal of Political Economy* 127 (6):2684–2734.
- Cai, Yongyang, Jevgenijs Steinbuks, Kenneth L. Judd, Jonas Jaegermeyr, and Thomas W. Hertel. 2020. “Modeling Uncertainty in Large Natural Resource Allocation Problems.” World Bank Policy Research Working Paper 9159.
- Caldara, Dario, Jesus Fernandez-Villaverde, Juan F. Rubio-Ramirez, and Wen Yao. 2012. “Computing DSGE models with recursive preferences and stochastic volatility.” *Review of Economic Dynamics* 15 (2):188–206.
- Czyzyk, J., M. P. Mesnier, and J. J. More. 1998. “The NEOS server.” *IEEE Computational Science & Engineering* 5:68–75.
- Den Haan, Wouter J., Kenneth L. Judd, and Michel Juillard. 2011. “Computational suite of models with heterogeneous agents II: Multi-country real business cycle models.” *Journal of Economic Dynamics and Control* 35 (2):175–177.
- Drud, A. S. 1994. “CONOPT – A Large Scale GRG Code.” *ORSA Journal on Computing* 6:207–216.

- Epstein, Larry G. and Stanley E. Zin. 1989. “Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework.” *Econometrica* 57 (4):937–969.
- Fernandez-Villaverde, Jesus, Grey Gordon, Pablo Guerron-Quintana, and Juan F. Rubio-Ramirez. 2015. “Nonlinear adventures at the zero lower bound.” *Journal of Economic Dynamics and Control* 57:182–204.
- Fernandez-Villaverde, Jesus and Oren Levintal. 2018. “Solution Methods for Models with Rare Disasters.” *Quantitative Economics* 9:903–944.
- Fernandez-Villaverde, Jesus, Juan Rubio-Ramirez, and Frank Schorfheide. 2016. “Solution and Estimation Methods for DSGE Models.” In *Handbook of Macroeconomics*, vol. 2, edited by John B. Taylor and Harald Uhlig, chap. 9. Elsevier, 527–724.
- Gali, Jordi. 2008. *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*. Princeton, N.J: Princeton University Press.
- Gill, Philip E., Walter Murray, and Michael A. Saunders. 2005. “SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization.” *SIAM Review* 47 (1):99–131.
- Grune, Lars, Willi Semmler, and Marleen Stieler. 2015. “Using nonlinear model predictive control for dynamic decision problems in economics.” *Journal of Economic Dynamics and Control* 60:112–133.
- Guerrieri, Luca and Matteo Iacoviello. 2015. “OccBin: A toolkit for solving dynamic models with occasionally binding constraints easily.” *Journal of Monetary Economics* 70:22–38.
- Haan, Wouter J. den and Albert Marcet. 1990. “Solving the Stochastic Growth Model by Parameterizing Expectations.” *Journal of Business & Economic Statistics* 8 (1):31–34.
- Jin, H.-H. and K.L. Judd. 2002. “Perturbation methods for general dynamic stochastic models.” Working paper, Stanford University.

- Judd, Kenneth L. 1992. “Projection methods for solving aggregate growth models.” *Journal of Economic Theory* 58 (2):410–452.
- . 1998. *Numerical Methods in Economics*. The MIT press.
- Judd, Kenneth L. and Sy-Ming Guu. 1993. “Perturbation Solution Methods for Economic Growth Models.” In *Economic and Financial Modeling with Mathematica*, edited by Hal R. Varian. New York, NY: Springer, 80–103.
- Judd, Kenneth L., Lilia Maliar, and Serguei Maliar. 2011. “Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models.” *Quantitative Economics* 2 (2):173–210.
- . 2012. “Merging Simulation and Projection Approaches to Solve High-Dimensional Problems.” Tech. Rep. w18501, National Bureau of Economic Research. URL <https://www.nber.org/papers/w18501>.
- Judd, Kenneth L., Lilia Maliar, Serguei Maliar, and Rafael Valero. 2014. “Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain.” *Journal of Economic Dynamics and Control* 44:92–123.
- Juillard, Michel and Sebastien Villemot. 2011. “Multi-country real business cycle models: Accuracy tests and test bench.” *Journal of Economic Dynamics and Control* 35 (2):178–185.
- Kollmann, Robert, Serguei Maliar, Benjamin A. Malin, and Paul Pichler. 2011. “Comparison of solutions to the multi-country Real Business Cycle model.” *Journal of Economic Dynamics and Control* 35 (2):186–202.
- Kydland, Finn E. and Edward C. Prescott. 1982. “Time to Build and Aggregate Fluctuations.” *Econometrica* 50 (6):1345–1370.
- Lenton, T. M., H. Held, E. Kriegler, J. W. Hall, W. Lucht, S. Rahmstorf, and H. J. Schellnhuber. 2008. “Tipping elements in the Earth’s climate system.” *Proceedings of the National Academy of Sciences* 105 (6):1786–1793.
- Levintal, Oren. 2018. “Taylor Projection: A New Solution Method for Dynamic General Equilibrium Models.” *International Economic Review* 59:1345–1373.

- Ljungqvist, L. and T.J. Sargent. 2000. *Recursive Macroeconomic Theory*. Cambridge: MIT Press.
- Magill, Michael J. P. 1977. “A local analysis of N-sector capital accumulation under uncertainty.” *Journal of Economic Theory* 15 (1):211–219.
- Maliar, Lilia and Serguei Maliar. 2015. “Merging simulation and projection approaches to solve high-dimensional problems with an application to a new Keynesian model.” *Quantitative Economics* 6 (1):1–47.
- Maliar, Lilia, Serguei Maliar, John B. Taylor, and Inna Tsener. 2020. “A tractable framework for analyzing a class of nonstationary Markov models.” *Quantitative Economics* 11 (4):1289–1323.
- Malin, Benjamin A., Dirk Krueger, and Felix Kubler. 2011. “Solving the multi-country real business cycle model using a Smolyak-collocation method.” *Journal of Economic Dynamics and Control* 35 (2):229–239.
- Marcet, Albert and Guido Lorenzoni. 1998. “Parameterized Expectations Approach: Some practical issues.” In *Computational Methods for Study of Dynamic Economies*, edited by R. Marimon and A. Scott. New York: Oxford University Press, 143–171.
- McCarl, B., A. Meeraus, P. van der Eijk, M. Bussieck, S. Dirkse, and F. Nelissen. 2016. *McCarl Expanded GAMS User Guide Version 24.6*. URL <https://www.gams.com/mccarlGuide>. Accessed on December 19, 2020.
- Miranda, Mario J. and Paul L. Fackler. 2002. *Applied Computational Economics and Finance*. The MIT Press.
- Negro, Marco Del, Frank Schorfheide, Frank Smets, and Rafael Wouters. 2007. “On the Fit of New Keynesian Models.” *Journal of Business & Economic Statistics* 25 (2):123–143.
- Nordhaus, William D. 2008. *A Question of Balance: Weighing the Options on Global Warming Policies*. Yale University Press.
- . 2017. “Revisiting the social cost of carbon.” *Proceedings of the National Academy of Sciences of the United States of America* 114 (7):1518–1523.

- Rust, John. 1996. “Numerical Dynamic Programming in Economics.” In *Handbook of Computational Economics*, vol. 1, edited by H. M. Amman, D. A. Kendrick, and J. Rust. Elsevier, 619–729.
- . 1997. “Using Randomization to Break the Curse of Dimensionality.” *Econometrica* 65 (3):487–516.
- . 2019. “Has Dynamic Programming Improved Decision Making?” *Annual Review of Economics* 11 (1):833–858.
- Scheidegger, Simon and Ilias Billionis. 2019. “Machine learning for high-dimensional dynamic stochastic economies.” *Journal of Computational Science* 33:68–82.
- Smets, Frank and Rafael Wouters. 2007. “Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach.” *American Economic Review* 97 (3):586–606.
- Smolyak, S.A. 1963. “Quadrature and interpolation formulas for tensor products of certain classes of functions.” *Soviet Mathematics Doklady* 4:240–243.
- Woodford, Michael. 2003. *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton, N.J. ; Woodstock, Oxfordshire England: Princeton University Press.