Quantitative Economics 14 (2023), 651–687 1759-7331/20230651

A simple but powerful simulated certainty equivalent approximation method for dynamic stochastic problems

Yongyang Cai
Department of Agricultural, Environmental, and Development Economics, The Ohio State University

Kenneth L. Judd
Hoover Institution, Stanford University

We introduce a novel simulated certainty equivalent approximation (SCEQ) method for solving dynamic stochastic problems. Our examples show that SCEQ can quickly 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. With the SCEQ method, a desktop computer will suffice for large problems, but it can also use parallel tools efficiently. The SCEQ method is simple, stable, and can utilize any solver, making it suitable for solving complex economic problems that cannot be solved by other algorithms.

Keywords. Stochastic dynamic programming, competitive equilibrium, large-scale model, integrated assessment model, new Keynesian model, zero lower bound, occasionally binding constraint, nonstationary problem, parallelism.

JEL classification. C61, C63, C68, Q54, Q58.

1. Introduction

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

However, many real-world problems are neither stationary nor low-dimensional. Life-cycle models often have time-varying endowments and preferences. Representative

Yongyang Cai: cai.619@osu.edu
Kenneth L. Judd: kennethjudd@mac.com

We thank three anonymous referees for their helpful comments and suggestions. 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, Steinbuks, Judd, Jaegermeyr, and Hertel (2020), which has been separated into this methodological paper and another paper focusing on applications to large natural resource allocation problems.

[1] See, for example, Magill (1977), Blanchard and Kahn (1980), Kydland and Prescott (1982), Judd and Guu (1993), Jin and Judd (2002)

[2] See, for example, Haan and Marcet (1990), Judd (1992, 1998), Marcet and Lorenzoni (1998), Miranda and Fackler (2002), Judd, Maliar, and Maliar (2011, 2012)

 https://doi.org/10.3982/QE1835

agent models often include nonstationary changes in the population, technology, resource stock, and climate, such as DSICE (Cai, Judd, and Lontzek (2017), Cai and Lontzek (2019)). Computing the full set of state-contingent decisions and prices is often computationally challenging, particularly when limited to desktop computing hardware. Those solutions may produce more information than is needed to answer the questions an economist has for a model. In addition, it is challenging to compute global solutions, particularly when value functions and/or decision rules are not smooth, such as dynamic problems with inequality constraints. This paper proposes the Simulated Certainty Equivalent approximation (SCEQ) method, which uses standard computational tools to provide useful approximations of high-dimensional, nonstationary economic models.

Before discussing the details of SCEQ, we present a notation, which will help us explain the key ideas. Consider a stochastic dynamic economic model, defined by a set of equations, which may depend on time if, for example, population, productivity, or preferences depend on time in a manner that cannot be simplified by detrending. We want to solve the model for all time up to some terminal time T (which may be infinite). Let M_t denote the model which "begins" with time t specifications for tastes and technology and has the terminal time T.

Let P_t denote the M_t model constructed by setting all disturbances in M_t to zero. P_t is a perfect foresight model and for each state x, we define $P_t(x)$ to be the perfect foresight solution of P_t if the state at t is x. The $P_t(x)$ problems are relatively easy to solve since they are perfect foresight models. Basically, perfect foresight models begin from some state not caring about how it got to that state and then solves for all variables in current and future periods under the assumption that there will be no shocks. For example, Nordhaus' DICE (Nordhaus (2008)) model is a perfect foresight model.

The key feature of SCEQ is that it solves a large set of perfect foresight models $P_t(x)$. Suppose that the model M_0 has an initial state x_0. The first step in SCEQ is to solve the perfect foresight model $P_0(x_0)$, which produces a path for the state variables for $t = 0, 1, \ldots, T$. SCEQ then constructs a second path in the following way: simulate the model by following the solution of $P_0(x_0)$ until some period s, at which point we introduce shocks to the system for only period s, which change the state from x_s to $x_s + \epsilon_s$, and then solve the perfect foresight system $P_s(x_s + \epsilon_s)$. SCEQ repeats this for many times s and many shocks ϵ_s. The collection of perfect foresight solutions is then our approximate solution to the stochastic model.

SCEQ wants the collection of perfect foresight solutions to approximate the stochastic processes implied by the underlying stochastic model. Therefore, SCEQ chooses the ϵ_s shocks according to the specifications for uncertainty in the original stochastic model. The frequency of shocks should also match the original model's specification. Most of our examples have shocks at each time t, implying that we also have shocks at each time. Basically, we start with some state, solve the model under the assumption that there are no future shocks to pin down the current variables, but then shock the system and resolve the resulting perfect foresight problem at the next period's state.

This description of SCEQ is meant to describe the general intuition. A more complete description will follow in later sections. At this point, we will point out the key advantages of SCEQ.

First, it only needs methods to solve perfect foresight models, a relatively simple task. Several papers have described a variety of methods; see, for example, Fair and Taylor (1983), Hall (1985), Fisher, Holly, and Hallett (1986), Boucekkine (1995), Gilli and Pauletto (1997), Juillard, Laxton, McAdam, and Pioro (1998).

Second, SCEQ is fast, and able to produce many simulated paths at low cost. It can be naturally parallelized; that is, one can solve for many of the paths simultaneously. In fact, it scales linearly, which means that doubling the number of processors will double the number of paths in the same amount of time. The marginal cost of a simulation path is small. If there is a shock at time s, the solver has the solution for $P_s(x_s)$ because it equals the path produced by $P_{s-1}(x_{s-1})$ after one takes out the time $s-1$ variables. Therefore, the solver can use $P_s(x_s)$ as the initial guess for $P_s(x_s+\epsilon_s)$. This warm start will reduce the cost of solving. Furthermore, the solver may also use the Jacobian from solving $P_{s-1}(x_{s-1})$, creating a hot start. The use of warm starts and hot starts imply that the marginal cost of computing a path is much less than computing a path using more standard initial guesses.

Third, there is no limitation on the stochastic processes. Large shocks to the system are as easily solved as small shocks. The desire to approximate the stochastic model's process also implies that the number of simulations should grow over time. We will often want the set of simulations to faithfully approximate the stochastic model conditional on the state x_t for some initial set of times $t = 1, \ldots, H$. That implies that we need a nontrivial sample of paths from each of the states x_t, $t = 1, \ldots, H$. Using k simulations from each of these states would imply at least k^H total simulated paths. This "curse of dimensionality" is partly offset by the use of hot starts but does imply H needs to be kept small. While conditional accuracy may be less for $t > H$, the paths for $t > H$ can still represent the unconditional distributions of variables.

Fourth, even if SCEQ does not produce desired information (such as an asset pricing kernel, one of the achievements in DSICE) the paths produced by SCEQ produces information that would help other methods. When solving stationary models, methods focus on computing time-invariant functions on a fixed portion of the state space. In nonstationary models, such as DSICE, the domain of the solution varies with the time t. If one uses projection methods to get a better solution, as done with DSICE, tractability requires guessing the range of states that will be part of the solution for each time t. This substantially adds to the necessary computational effort. The paths produced by SCEQ will approximate the domain of the solution for each time t. Furthermore, SCEQ produces information, which could be used to choose efficient methods of approximation in applying projection methods to the problem. We leave development of these observations to future papers.

Fifth, SCEQ has easily implemented accuracy checks. We will demonstrate the quality of SCEQ in several examples below, but a method should also include diagnostics, which indicate if the sample sizes are large enough and if SCEQ is delivering good approximations to the stochastic model, M_0, which is what we really care about.

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 multicountry 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 2 hours to provide 1000 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 an accuracy of two or three digits, 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

Alternative solution methods for dynamic economic models use different notions of what is a solution, reflecting different objectives. This section compares SCEQ to alternatives in terms of what different methods can deliver.

Local approximation methods are the cheapest to compute. The cheapest is the common log-linear approximations often used in macroeconomics, cheap because it needs only to compute the steady state (defined by a system of nonlinear equations), the local Jacobian (very cheap if one uses automatic differentiation), and solves the resulting Ricatti equation. Even high-dimensional models are easy to solve; for example, it takes only 90 seconds to do an eigenvalue–eigenvector analysis of a random 20,000 by 20,000 dense matrix on a 2019 Mac Pro with 24 cores. Linearizations are based on the Hartman–Grobman theorems for differential equations and maps.

Linearization methods have important limitations. First, the mathematical foundations require that the dynamic system is smooth in its variables. Second, the linear approximation is valid only for states close to the deterministic steady state where "close" can be very small. Perturbation methods can only provide locally accurate solutions around the nonstochastic steady state, which are then treated as if they were globally valid. But it is often not sufficient for decision-making purposes because the nonstochastic 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. Third, if the procedure uses only first- and second-order derivatives of the utility function then the approximation cannot accurately approximate movements in, for example, risk premia.

Higher-order expansions are easily computed, where each higher-order expansion is determined by a system of equations that can be computed directly (see Judd (1998), Jin and Judd (2002)). They will improve accuracy but its validity will be limited by the radius of convergence of the power series ("infinite-order") expansion. Higher-order expansions also require extended precision arithmetic, something which is readily available today.

Log-linear approximations are not affected by the stochastic elements of a model. They only produce a linear approximation for the underlying deterministic model and then simulate it with the stochastic processes. This is what we refer to as the "certainty equivalent" property of the log-linear approximation. Perturbation methods more generally can compute dependencies with respect to the statistics of a model's stochastic processes (see Judd (1998), Judd and Guu (1993)). The extra terms are defined by linear equations and may rely on bifurcation theorems. The mathematics behind all perturbation methods assume smoothness in a model's functions. Therefore, they cannot be directly applied to models with occasionally binding constraints.

At the other extreme are projection methods, which aim to approximate value functions, decision rules, and/or price functions over a large state space. Value function iterations (VFI), projection methods, and time iterations are often used 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 iterations 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 a 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.[3] These methods require efficient ways to approximate functions and compute conditional expectations, difficult challenges for complex high-dimensional problems. The solutions allow one to examine nonlinear properties of a solution over a global range of states. The high-quality approximations allow the solutions to approximate features such as movements in an asset pricing kernel. Advanced examples of this approach have often required high-power computing (see, e.g., Cai, Judd, and Lontzek (2017), Cai and Lontzek (2019)) but this is becoming less of a limitation with ongoing advances in computing power.

SCEQ lies between the full global nonlinear approximation methods and local approximations in terms of both cost and usefulness of results. The SCEQ method focuses directly on computing a set of simulations without approximating any nonlinear func-

[3]See Rust (1996), Judd (1998), Ljungqvist and Sargent (2018), Miranda and Fackler (2002), Bertsekas (2005, 2007), Aruoba, Fernandez-Villaverde, and Rubio-Ramirez (2006), Kollmann, Maliar, Malin, and Pichler (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.

tions. These simulations can be used directly to compute the statistical properties of the solution. It is a kind of a "certainty equivalent" method because the solution at any time t is based on a deterministic model for all times at and after time t. The shocks are drawn from the distributions of the stochastic model, implying that the results will reflect those stochastic properties. Simulations based on linear approximations can easily stray outside the domain where the linear approximation is valid, whereas each path produced by SCEQ is based on the model at the states it visits.

Paths generated by SCEQ have many advantages over simulations based on standard linear approximations. They certainly will be better at approximating the statistical properties. SCEQ shares some of the limitations inherent in certainty equivalent methods, which may argue for projection methods. However, solutions produced by projection methods are often used only to compute stochastic properties of the problem, such as means, variances, and covariances of the variables, which can be well approximated by SCEQ simulations. The choice among linearization methods, SCEQ, and projection methods can be based on what one wants to study and the set of states visited in equilibrium.

SCEQ is particularly attractive when trying to solve nonstationary models. There is no nonstochastic steady state in nonstationary models on which standard linearization methods are based. Projection methods may produce higher-quality approximations, but it is often necessary to guess an approximation domain and an approximation method, then check if these can solve the problem, and this process may have to be repeated many times.

SCEQ is related to another certainty equivalent method. 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 node in the state space at the initial time, then fits the solutions with a globally nonlinear 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 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, Maliar, Maliar, and Valero (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 ob-

tain 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)).[4]

(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,[5] 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.

[4]SCEQ's solution could have a wide domain, particularly if its initial state is far away from the nonstochastic steady state for a stationary problem. For example, in our multicountry model, the initial levels of capital range from 0.1 to 10 across countries, while the steady-state level is 1.

[5]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.

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 iterations 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 the "curse of dimensionality" for both endogenous and exogenous state variables;[6] 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 iterations are generally not stable.[7]

SCEQ has advantages over other proposed simulation 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 ϵ-distinguishable set (EDS) method to merge projection approaches and simulation. However, both methods only solve infinite-horizon stationary problems. Moreover, it will be challenging for their methods to solve some problems that SCEQ can simply handle, for example, (1) a high-dimensional problem with kinks, in which the EDS method may have to use locally adjusted EDS grids and piecewise local basis functions, and then the associated global nonsmoothness may make it unstable or inefficient to find the global minimizer in their projection step to minimize residuals; (2) an application requiring a sufficiently accurate solution along some simulation time paths, which are outside its simulated ergodic set,[8] in which the EDS method has to include all the points along the time paths so it has a wide approximation domain, which is challenging for their projection methods. Furthermore, Maliar and Maliar (2015) point out that their solution methods may fail to converge, which are common limitations for all projection methods. In addition, 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. However, if one wants to solve stationary problems without kinks, then the GSSA, EDS, and projection methods could be more accurate or faster than SCEQ,[9] particularly when SCEQ uses the optimization solvers of Matlab,[10] not other programming

[6]There are methods to alleviate or even avoid the "curse of dimensionality" for some problems; see, for example, Rust (1997, 2019), Judd et al. (2014), Brumm and Scheidegger (2017), Cai (2019), and Scheidegger and Bilionis (2019).

[7]Note that VFI can be stable if it uses a shape-preserving approximation method (Cai and Judd (2013)).

[8]For example, our DSICE example in Section 4.3 starts with a real world initial state that is far away from its steady state, which will take thousands of years to be reached, while policymakers want to compute an optimal carbon tax path in this century.

[9]SCEQ could still be faster for the stationary problems without kinks, if there are many compute cores for parallelization, while the GSSA, EDS, and projection methods do not have efficient parallelization.

[10]The optimization step of SCEQ can be implemented with the deterministic simulation of Dynare (Adjemian et al. (2011)), a Matlab toolbox for solving DSGE models, particularly stationary problems in macroeconomics.

languages such as GAMS (General Algebraic Modeling System)[11] that can have much faster implementation in solving many large-scale optimization problems as shown in our examples. Moreover, the GSSA, EDS, and projection methods can solve stationary dynamic portfolio problems or dynamic stochastic problems with Epstein–Zin preferences (Epstein and Zin (1989)) but SCEQ cannot. In addition, theoretically it is always possible (while practically challenging) for the GSSA, EDS, and projection methods to improve their solution's accuracy with a higher degree approximation (and more simulation points for the GSSA and EDS methods), but SCEQ cannot. 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.

The examples in the following sections show the advantages of SCEQ in a variety of cases. We review our findings in the rest of this section. First, SCEQ does not suffer from the so-called "curse of dimensionality" in our examples: its accuracy is independent of the state space's dimensionality, and in our multicountry examples its computational time is a nearly linear and at worst a cubic function of the state space's dimensionality.[12] Second, SCEQ efficiently solves high-dimensional dynamic stochastic problems with occasionally binding constraints: it only took hours on a standard computer to provide 1000 simulation paths of solutions for 400-dimensional multicountry problems with a lower bound on investments. Third, SCEQ efficiently solves both stationary and 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 (computational 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 problem (Section 4.2.2), and only took hours to provide 1000 simulation paths of solutions for larger 200-country problems with 201 or 400 state variables (Sections 4.2.1 and 4.2.3). If there are many compute cores, SCEQ may take only minutes for the large 200-country problems with

[11]There are many reliable and efficient professional solvers available using GAMS (McCarl, Meeraus, van der Eijk, Bussieck, Dirkse, and Nelissen (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.

[12]The "curse of dimensionality" refers to the case where computational cost grows exponentially with the dimensionality of state space. There is also another potential "curse of dimensionality" in the control space. For example, if the control variables are discrete and we use enumeration to find the optimal solution, then it suffers from the "curse of dimensionality." But usually we can use an efficient optimization solver to avoid the "curse of dimensionality." For instance, if control variables are continuous and the objective and constraints are smooth, then Newton's method has the quadratic convergence rate in general in finding the optimal solution if its initial guess is in the neighborhood. For discrete control variables, an integer programming method may also avoid the "curse of dimensionality."

201 or 400 state variables, because it can be naturally parallelized with high efficiency. 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 two or three digits for all examples in this paper.

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

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

where $\boldsymbol{\epsilon}_t$ is a serially uncorrelated random vector process[13] and $\mathbf{g}_t$ is a vector of functions which could be time-varying: its ith element, $g_{t,i}$, returns the ith state variable at time $t+1$: $x_{t+1,i}$.

Without loss of generality, we assume the mean or median of $\boldsymbol{\epsilon}_t$ 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, that is, $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t)$, since it can be rewritten as $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t) + 0 \cdot \boldsymbol{\epsilon}_{t+1}$, we will still denote it as $x_{t+1,i} = g_{t,i}(\mathbf{x}_t, \mathbf{a}_t, \boldsymbol{\epsilon}_{t+1})$. Similarly, if there are some unused elements of $\boldsymbol{\epsilon}_{t+1}$ or some redundant arguments in a function $g_{t,j}$, we can multiply them by zero in $g_{t,j}$, and thus still use $x_{t+1,j} = g_{t,j}(\mathbf{x}_t, \mathbf{a}_t, \boldsymbol{\epsilon}_{t+1})$.

We solve the following social planner's problem:

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

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 $\mathbf{a}_t \geq 0$).[14] 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 nonstochastic 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

[13] If a dynamic model has serially correlated random variables, they are exogenous state variables, and we can use an uncorrelated vector $\boldsymbol{\epsilon}_t$ in their transition laws.

[14] An equality constraint $f(x, a) = 0$ can be written as a combination of $f(x, a) \geq 0$ and $-f(x, a) \geq 0$.

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.

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* Δ_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* $\boldsymbol{\epsilon}_t$ *to get* m *paths, denoted* $\boldsymbol{\epsilon}_t^i$ *for path* i*, from* $t = 1$ *to* T^**. Let* $\mathbf{x}_0^i = \mathbf{x}_0$ *and iterate forward through steps 2 and 3 for* $s = 0, 1, 2, \ldots, T^* - 1$.

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

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

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

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

Algorithm 1 obtains simulated pathways of optimal decisions and states. It contains three steps: (i) the initialization step, which chooses an appropriate Δ_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 $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \boldsymbol{\epsilon}_{s+1}^i)$ for $s = 0, 1, 2, \ldots, T^* - 1$. 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 $i = 2, \ldots, m$. 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, Δ_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 Δ_s for every state at s. Ideally, $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$ is chosen to be an approximation of the cumulative optimal welfare from time $s+\Delta_s$ to time $T-1$, that is,

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

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 Δ_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,[15] 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 Δ_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 Δ_s will not change the solutions. We use these solutions as the "true" solutions. We then choose a smaller Δ_s and a different $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s})$, resolve (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 Δ_s and a time-invariant "terminal" value function. In the end, we choose the best pair of Δ_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 $\boldsymbol{\epsilon}_t$ is a serially uncorrelated stochastic process, if all transition laws are continuous we can replace $\boldsymbol{\epsilon}_t$ in (2) by its (zero) mean or median for simplicity. Generally, we

[15]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 Δ_s, and a smaller β implies that we can choose a smaller Δ_s in SCEQ. Such a truncation is often used in the literature; see, for example, Nordhaus (2008), Grune, Semmler, and Stieler (2015), Cai, Judd, and Steinbuks (2017), Cai, Judd, and Lontzek (2017), Cai and Lontzek (2019), Maliar, Maliar, Taylor, and Tsener (2020).

can replace ϵ_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, Fernandez-Villaverde, Rubio-Ramirez, and Yao (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, $\mathbf{x}_{s+1}^i = \mathbf{g}_t(\mathbf{x}_s^i, \mathbf{a}_s^i, \boldsymbol{\epsilon}_{s+1}^i)$, given the realization of shocks, $\boldsymbol{\epsilon}_{s+1}^i$. 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 reoptimizing 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, $\{\mathbf{x}_s^i\}_{s=0}^{T^*}$, which corresponds to the realized path of shocks, $\{\boldsymbol{\epsilon}_s^i\}_{s=1}^{T^*}$. 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: For example, 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 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.

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 $\mathbf{x}_{t+1} = \mathbf{g}_t(\mathbf{x}_t, \mathbf{a}_t, \epsilon_{t+1})$, where $\mathbf{x}_t$ is the state vector, $\mathbf{a}_t$ is the action vector, and ϵ_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):

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

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

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* Δ_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* ϵ_t *to get* m *paths, denoted* ϵ^i_t *for path* i*, from* $t = 1$ *to* T^**. Let* $\mathbf{x}^i_0 = \mathbf{x}_0$ *and iterate forward through steps 2 and 3 for* $s = 0, 1, 2, \ldots, T^* - 1$.

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

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

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

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

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.

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 $\mathbf{a}^i_s$ 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 the 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, Θ, in the current-period state space, use the approximate policy function to compute the next-period states and decisions for each point in Θ, 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 Euler 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 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 errors at the simulated states without explicitly constructing a policy function approximation.[16]

Note that for an infinite horizon problem, if we choose $\Delta_s = \widehat{T} - s$ then the infinite horizon problem is viewed as a finite $\widehat{T}$-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 Δ_s and a time-invariant "terminal" value function, then the truncation error is reflected in the Euler errors (see Appendix A.1 in the Online Supplementary Material (Cai and Judd (2023) 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 errors, with little computational cost.

If there are many shocks, a large number of quadrature nodes are required to compute Euler errors, running the above accuracy measure method could be computationally expensive. Here, we also provide a weaker but no-cost indicator for checking accuracy: we use the Monte Carlo simulation idea to estimate the Euler 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 error at the initial state by simply taking the average across the simulated solutions

[16]For low-dimensional problems with smooth policy functions, we can also compute Euler 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 nonsmooth policy functions.

for the second period. Following the law of large numbers, the accuracy of the Euler error is proportional to $1/\sqrt{m}$. Our multicountry examples show that the Euler error at the initial state is close to the Euler errors across other periods.[17]

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 multicountry 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 two or three digits. 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.[18] It took minutes or at most hours to solving each example. To further illustrate the implementation of SCEQ, we also provide Matlab code for the first simple example with the fmincon optimization solver, and for the last competitive equilibrium example with the fsolve equation solver. However, since the fmincon optimization solver in Matlab is slow in solving large scale problems, it will be much more efficient to apply SCEQ in GAMS or other programming languages with efficient optimization solvers (e.g., AMPL (Fourer, Gay, and Kernighan (1990)), FORTRAN, C, etc.), while Matlab can be used for solving low-dimensional problems, which could still be challenging for other computational methods if their value/policy functions have kinks.

4.1 *An illustrative example*

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

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

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

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

[17] In some rare cases, the Euler error in the first period may not be enough. See Appendix A.6 for an example.

[18] 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.

where K_t is capital and c_t is consumption at time t, $\beta = 0.96$ is the discount factor, and $\alpha = 0.3$. 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 nonstochastic steady state of capital is $K_{ss} = 1$. We choose the full depreciation rate $\delta = 1$ and the utility function $u(c) = \ln(c)$ so the problem has an analytical solution of consumption policy function: $\mathcal{C}^*(K, \theta) = (1 - \alpha\beta)\theta A K^\alpha$. Appendix A.2 provides another illustrative example for the same optimal growth model but with a power utility function and 10% depreciation rate, in which case there is no analytical solution.

4.1.1 *Implementation of SCEQ* 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

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

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

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 $t \geq s + \Delta_s$.[19] We simulate a sequence of $\boldsymbol{\epsilon}_t$ to get $m = 1000$ paths for the periods of interest: $\{\boldsymbol{\epsilon}_t^i : 1 \leq i \leq m, 1 \leq t \leq T^*\}$.

We start from the initial period $s = 0$, and set the initial state $(K_0^i, \theta_0^i) \equiv (K_0, \theta_0) = (1, 1)$ for each $i = 1, \ldots, m$. 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 θ_t replaced by its median conditional on the realized value θ_s^i (i.e., $(\theta_s^i)^{\rho^{t-s}}$ for any $t \geq s$) for simulation path i. That is, the optimization step of Algorithm 1 solves the following problem:

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

and the dynamic system starts with the state values (K_s^i, θ_s^i) at time s for the ith simulation path. We solve the deterministic finite-horizon problem (8) using the CONOPT optimization solver (or fmincon in Matlab) to obtain the optimal consumption c_s^i, for each $i = 1, \ldots, m$. The simulation step of Algorithm 1 uses the state values (K_s^i, θ_s^i) at time s and their associated solution c_s^i from the optimization step 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 $\theta_{s+1}^i = (\theta_s^i)^\rho \exp(\sigma\boldsymbol{\epsilon}_{s+1}^i)$ with the simulated value $\boldsymbol{\epsilon}_{s+1}^i$ in the initialization step.

[19] We tried $\Delta_s = 50$ and found that it has little impact on the solutions obtained with $\Delta_s = 30$.

We iterate the optimization step and the simulation step from $s = 0$ until $s = T^* - 1 = 19$. Thus, we generated $m = 1000$ simulated paths of solution $(K_s^i, \theta_s^i, c_s^i)$ for the periods of interest (the first 20 periods).[20] The GAMS code took 4.2 minutes to run on a Mac Pro desktop computer.[21]

4.1.2 *Accuracy measures* The normalized Euler error for this problem is

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

We can estimate it at the initial state by replacing the expectation with the average across 1000 simulated solutions in the second period:

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

The estimated normalized Euler error at the initial state is 5.3×10^{-8}. We also employ the accuracy measure method in Appendix A.1 and find that the $\mathcal{L}^\infty$ Euler error over the 1000 simulated paths is 3.1×10^{-7}, only slightly larger than the Euler error at the initial state. If we compare the SCEQ's simulated consumption, denoted $c_t^{i,\mathrm{SCEQ}}$, with the true analytical solution $\mathcal{C}^*(K, \theta) = (1 - \alpha\beta)\theta AK^\alpha$, then the $\mathcal{L}^\infty$ relative error of the SCEQ solution, defined as

$$\max_{0\le t<20,\, 1\le i\le 1000} \frac{\left|c_t^{i,\mathrm{SCEQ}} - \mathcal{C}^*\left(K_t^{i,\mathrm{SCEQ}}, \theta_t^{i,\mathrm{SCEQ}}\right)\right|}{\mathcal{C}^*\left(K_t^{i,\mathrm{SCEQ}}, \theta_t^{i,\mathrm{SCEQ}}\right)},$$

is 8.3×10^{-8}, where $(K_t^{i,\mathrm{SCEQ}}, \theta_t^{i,\mathrm{SCEQ}})$ are the SCEQ's simulated states. These small errors are due to that certainty equivalent approximation in future periods has no effect on the current-period solution for this specific example: with a given state (K_s, θ_s) at time s, no matter how we choose a deterministic path of θ_t for $t > s$, its associated deterministic model's optimal consumption at time s is always $\mathcal{C}^*(K_s, \theta_s)$ if the truncation error is small. In the other illustrative example in Appendix A.2, certainty equivalent approximation in future periods has effect on the current-period solution, and its SCEQ solution's error is around 0.1% using our estimated Euler error measures or the relative difference with a high-accuracy solution from the value function iteration. Both illustrative examples show that our estimated normalized Euler errors are close to the true relative errors, so we can use them for checking the accuracy of SCEQ's solution.

[20]Since the problem is stationary, the solutions $\{(K_s^i, \theta_s^i, c_s^i) : 0 \le s < 20, 1 \le i \le 1000\}$ 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.

[21]If we do not implement parallelism on the six compute cores, then the runtime is 22 minutes, so the parallel efficiency is 87%. We also tried a parallel Matlab code, and it took 11 minutes to run on the same computer, much slower than the GAMS code.

4.2 *Application to multicountry models*

In this application, we show that SCEQ can solve high-dimensional dynamic stochastic problems with occasionally binding constraints and up to 400 state variables and 600 decision variables in every period. Den Haan, Judd, and Juillard (2011) introduce a multicountry model. Here, we apply SCEQ to solve its modified version. We assume that there are N countries with a capital stock state vector $K_t = (K_{t,1}, \ldots, K_{t,N})$, and the production function for the jth country at time t is

$$Y_{t,j} = \zeta_{t,j} A_{t,j} (K_{t,j})^{\alpha} (\ell_{t,j})^{1-\alpha}, \tag{11}$$

where $\ell_{t,j}$ is labor supply, α 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

$$K_{t+1,j} = (1-\delta) K_{t,j} + I_{t,j}, \tag{12}$$

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

The utility function for the jth country is

$$u_j(c_{t,j}, \ell_{t,j}) = \frac{(c_{t,j})^{1-\frac{1}{\gamma_j}}}{1-\frac{1}{\gamma_j}} - B_{t,j} \frac{(\ell_{t,j})^{1+\frac{1}{\eta_j}}}{1+\frac{1}{\eta_j}}, \tag{13}$$

where $c_{t,j}$ is consumption, γ_j is the intertemporal elasticity of substitution, η_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 $c_t = (c_{t,1}, \ldots, c_{t,N})$ and $\ell_t = (\ell_{t,1}, \ldots, \ell_{t,N})$, where τ_j are country-specific weights. The social planner has the following aggregate world resource constraint:

$$\sum_{j=1}^{N} (c_{t,j} + I_{t,j} + \Gamma_{t,j}) = \sum_{j=1}^{N} Y_{t,j}, \tag{14}$$

where

$$\Gamma_{t,j} \equiv \frac{\phi}{2} K_{t,j} \left(\frac{I_{t,j}}{K_{t,j}} - \delta \right)^2 \tag{15}$$

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

$$I_{t,j} \geq I_{\min}, \quad \forall t, j. \tag{16}$$

That is, the social planner solves

$$\max_{c,\ell,I} \mathbb{E}\left(\sum_{t=0}^{\infty} \beta^t U(c_t, \ell_t)\right) \tag{17}$$

subject to the transition law (12) and feasibility constraints (14) and (16), for each t and j, where β is the discount factor.

We set $\beta = 0.99$, $\alpha = 0.33$, $\delta = 0.025$, and $\phi = 0.5$. While SCEQ can also solve multi-country problems 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 jth country is set as

$$K_{0,j} = \exp\left(\ln(K_{\min}) + \big(\ln(K_{\max}) - \ln(K_{\min})\big)\frac{j-1}{N-1}\right)$$

with $K_{\min} = 0.1$ and $K_{\max} = 10$ for $j = 1, \ldots, N$. 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 $I_{\min} = 0.9 I_{ss}$ so the inequality (16) will bind frequently.

Since our policy functions in this application have kinks and a wide-ranging and high-dimensional state space (the dimensions of our cases in this application are from 21 to 400), it will be challenging for any existing methods to solve our problems, as they will require a high-degree approximation or an adaptive sparse grids approximation for value or policy functions. But it is simple for SCEQ. 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 ζ_t is a Markov chain with three possible values: 0.9, 1.0, and 1.1, and its transition probability matrix is

$$P = \begin{bmatrix} 0.8 & 0.2 & \\ 0.2 & 0.6 & 0.2 \\ & 0.2 & 0.8 \end{bmatrix}. \tag{18}$$

Its initial state is $\zeta_0 = 1$.

For SCEQ, we choose $\Delta_s = 50$ and the "terminal" value function $V_{s+\Delta_s}(\mathbf{x}_{s+\Delta_s}) = U(0.75AK^{\alpha}_{s+\Delta_s}, 1)/(1-\beta)$, where $K^{\alpha}_{s+\Delta_s} = (K^{\alpha}_{s+\Delta_s,1}, \ldots, K^{\alpha}_{s+\Delta_s,N})$, assuming 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 ζ_t by its mean conditional on the realized values of ζ_s for all $t \geq s$:

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

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

$$\begin{aligned}
\max_{c,\ell,I} \quad & \sum_{t=s}^{s+49} \beta^{t-s} U(c_t, \ell_t) + \beta^{50} U\big(0.75 A K_{s+50}^{\alpha}, 1\big)/(1-\beta) \\
\text{s.t.} \quad & K_{t+1,j} = (1-\delta)K_{t,j} + I_{t,j}, \\
& \sum_{j=1}^{N} (c_{t,j} + I_{t,j} + \Gamma_{t,j}) = \sum_{j=1}^{N} \big(\mathbb{E}\big(\zeta_t \mid \zeta_s^i\big) A (K_{t,j})^{\alpha} (\ell_{t,j})^{1-\alpha}\big), \\
& t = s, s+1, \ldots, s+49,
\end{aligned} \tag{19}$$

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

We first solve the problem with $N = 10$ countries and generate 1000 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 with six cores. 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 $I_{\min} = 0.09$.[22] The $\mathcal{L}^{\infty}$ Euler error on the simulated

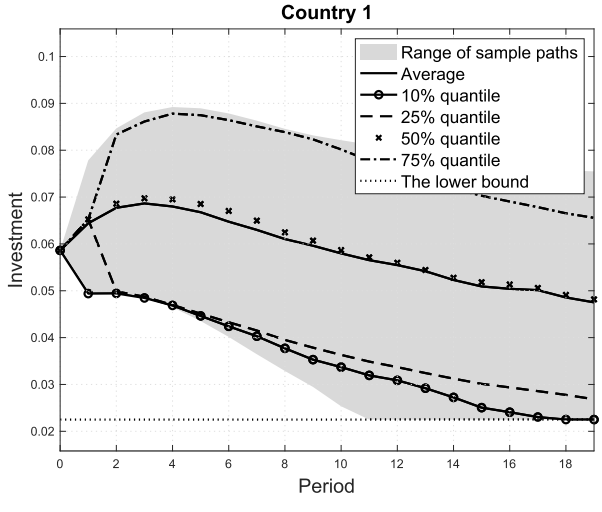

FIGURE 1. Distribution of investment for country 1.

[22]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

TABLE 1. Running times and errors for Case 1.

N	Number of Cores	Time (in Hours) for SCEQ	Euler Error		
			$\mathcal{L}^\infty$	$\mathcal{L}^1$	Initial State
10	6	0.18	6.9(−3)	3.7(−3)	6.7(−3)
20	6	0.37	7.1(−3)	3.7(−3)	6.9(−3)
50	6	1.1	7.1(−3)	3.7(−3)	7.1(−3)
100	6	3.4	7.1(−3)	3.7(−3)	7.1(−3)
200	6	16.1	7.1(−3)	3.7(−3)	6.8(−3)

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

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 Δ_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, that is, the time taken to generate 1000 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.[23]

The last column of Table 1 also reports the Euler error at the initial state, which is computed with the average across the 1000 simulated solutions for the second period. 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.[24] 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

nonstochastic steady state in the long run, especially if their initial capital levels are higher 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.

[23]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.

[24]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.

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.[25]

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:

$$P = \begin{bmatrix} 0.99 & 0 \\ 0.01 & 1 \end{bmatrix}. \tag{20}$$

The initial state is $\zeta_0 = 1.0$. This shock represents a global risk that results in a 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 ζ_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 $0 \le t < T^*$, the second path assumes that the shock happens in the second period, that is, $\zeta_0 = 1.0$ and $\zeta_1 = \cdots = \zeta_{T^*-1} = 0.95$, and so on, with the last path having $\zeta_0 = \cdots = \zeta_{T^*-2} = 1.0$ 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), that is, solve problem (19) with

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

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

Table 2 shows that with $T^* = 20$ and $\Delta_s = 50$, SCEQ took only seconds for $N \le 50$, 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 in Case 1.

TABLE 2. Running times and errors for Case 2.

N	Number of Cores	Time (in Minutes) for SCEQ	Euler Error $\mathcal{L}^\infty$
10	1	0.1	6.9(−3)
20	1	0.2	6.9(−3)
50	1	0.8	6.9(−3)
100	1	3	6.9(−3)
200	1	17	6.9(−3)

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

[25]Appendix A.6 shows that SCEQ can also solve problems with an irreversible risk and endogenous probabilities.

TABLE 3. Running times and errors for Case 3.

N	Number of Cores	Time (in Hours) for SCEQ	Euler Error at the Initial State
10	6	0.18	6.7(−3)
20	6	0.33	6.8(−3)
50	6	0.97	7.1(−3)
100	6	3.3	7.3(−3)
200	6	16.4	7.2(−3)

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

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 (ε_{t+1}). We assume $\zeta_{t,j}$ is a continuous exogenous state variable, following the stochastic process:

$$\ln(\zeta_{t+1,j}) = \rho \ln(\zeta_{t,j}) + \sigma_1 \epsilon_{t+1,j} + \sigma_2 \varepsilon_{t+1} \tag{21}$$

for each $j = 1, \ldots, N$, where $\epsilon_{t+1,j}$, $\varepsilon_{t+1} \sim$ i.i.d. $\mathcal{N}(0, 1)$ (i.e., $\epsilon_{t+1,j}$ and ε_{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 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.[26]

In the optimization step of Algorithm 1, we still solve (19) but replace $\mathbb{E}(\zeta_t \mid \zeta_s^i)$ 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 ith simulation path. We generate 1000 simulated paths of the first 20 periods. The solutions show that investments bind frequently an the lower bound $I_{\min}$ for each country.

Table 3 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.

[26]This example is a more complicated version of the multicountry model in Judd, Maliar, and Maliar (2012) and Maliar and Maliar (2015), as our example has kinks, a much wider-ranging state space, labor, and adjustment cost, while their model does not. Our example is also challenging for the method of Judd, Maliar, and Maliar (2012) and Maliar and Maliar (2015) to solve. But it is simple to apply SCEQ to solve their model; see Appendix A.3.

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.[27] All exogenous paths and parameter 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 $\mathbf{M}_t = (M_{\mathrm{AT},t}, M_{\mathrm{UO},t}, M_{\mathrm{LO},t})^\top$ be the carbon concentrations in the atmosphere, and upper and lower levels of the ocean, respectively. These concentrations evolve over time according to

$$\mathbf{M}_{t+1} = \mathbf{\Phi}_{\mathrm{M}}\mathbf{M}_t + (E_t, 0, 0)^\top, \tag{22}$$

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

$$\mathbf{T}_{t+1} = \mathbf{\Phi}_{\mathrm{T}}\mathbf{T}_t + (\xi_1 F_t, 0)^\top, \tag{23}$$

where $\mathbf{\Phi}_{\mathrm{T}}$ is a linear transition matrix and ξ_1 is a parameter. $F_t = \eta \log_2(M_{\mathrm{AT},t}/M^*_{\mathrm{AT}}) + F_{\mathrm{EX},t}$ is global radiative forcing, where η is a parameter, M^*_{AT} is the preindustrial atmospheric carbon concentration, and $F_{\mathrm{EX},t}$ 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 α 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

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

where π_1 and π_2 are parameters. Economic production has industrial emissions $E_{\mathrm{Ind},t} = \sigma_t(1-\mu_t)Y_t$, which is proportional to gross output but can be reduced by mitigation, measured by the emission control rate $\mu_t \in [0, 1]$, and σ_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 abatement cost and θ_2 is a parameter. Thus, the transition law of capital is

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

[27] Appendix A.6 shows that SCEQ can solve another simpler version of DSICE with climate tipping risks and endogenous tipping probabilities. It took only 2 minutes on a single compute core and gave an acceptable solution accuracy.

where δ is the depreciation rate and C_t is consumption. Note that the emission control rate μ_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:

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

subject to the transition laws (22)–(24) of six state variables: $\mathbf{M}_t$, $\mathbf{T}_t$, and K_t, where C_t and μ_t are decision variables, β 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, $\widetilde{A}_t$, is stochastic: $\widetilde{A}_t = \zeta_t A_t$, where A_t is the deterministic trend and ζ_t is a Markov chain representing a productivity shock. For simplicity, we assume that ζ_t follows the same distribution and transition probabilities as in Section 4.2, that is, ζ_t has three possible values: 0.9, 1.0, and 1.1, and its transition probability matrix is given by (18). Gross economic output is $Y_t = \widetilde{A}_t K_t^{\alpha} L_t^{1-\alpha}$, and we solve the social planner's problem

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

subject to the transition laws (22)–(24) and the Markov chain of ζ_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 can apply SCEQ to overcome these challenges.[28] As in Section 4.1, in the optimization step of Algorithm 1 we replace ζ_t by its mean conditional on the realized value of ζ_s for all $t \geq s$.

Figure 2 displays distributions of the optimal carbon taxes in the first $T^* = 100$ years from 1000 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) due to the differences of utility function and economic risk, but Figure 2 still shows that the optimal carbon tax is stochastic and has

[28]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 be $V_{600}(\mathbf{x}_{600}) \equiv 0$.

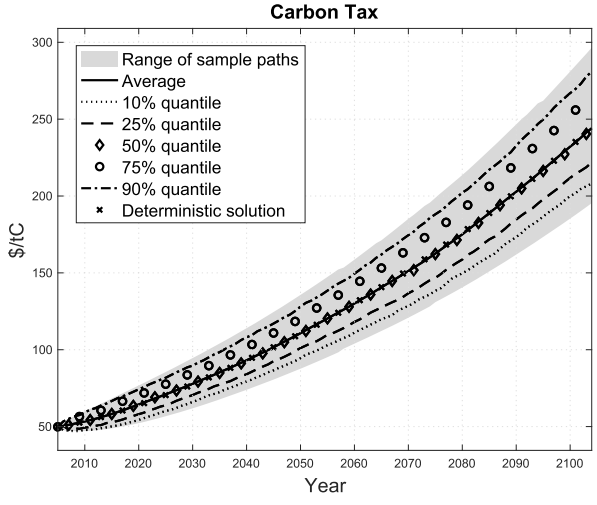

FIGURE 2. Carbon tax for DSICE with economic risk.

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.

4.3.3 *Accuracy measures* Table 4 lists the Euler errors, specified in Appendix A.5, at the simulated states using the accuracy measure method in Appendix A.1. It shows that the $\mathcal{L}^\infty$ Euler error is 0.0011 and the $\mathcal{L}^1$ Euler error is 1.1×10^{-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. Table 4 shows that 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.[29] In addition, the relative $\mathcal{L}^\infty$ error is 0.0028 and

TABLE 4. Errors for DSICE with economic risk.

		Relative Difference With the VFI Solution					
Euler Error		Tax		C_t		μ_t	
$\mathcal{L}^\infty$	$\mathcal{L}^1$	$\mathcal{L}^\infty$	$\mathcal{L}^1$	$\mathcal{L}^\infty$	$\mathcal{L}^1$	$\mathcal{L}^\infty$	$\mathcal{L}^1$
1.1(−3)	1.1(−4)	2.3(−2)	3.1(−3)	2.8(−3)	2.4(−3)	1.3(−2)	1.7(−3)

[29]The errors of the VFI solutions are about 0.1–1% because we use degree-6 complete Chebyshev polynomial approximations, so our relative errors using the VFI solution as the "true" solution are a bit 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 1.2×10^{-4} for the VFI solution of the example in Appendix A.2, while it only requires a degree-10 complete Chebyshev polynomial approxima-

0.013 for C_t and μ_t, respectively, and the relative $\mathcal{L}^1$ error is 0.0024 and 0.0017 for C_t and μ_t, respectively.[30]

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).[31]

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 β_t is the stochastic process

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

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

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

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

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

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

tion to obtain the $\mathcal{L}^\infty$ Euler error 6.4×10^{-4}, or a degree-20 complete Chebyshev polynomial approximation to obtain the $\mathcal{L}^\infty$ Euler error 3.9×10^{-4}.

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

[31]New Keynesian DSGE models have been studied frequently in the literature; see, for example, Woodford (2003), Negro, Schorfheide, Smets, and Wouters (2007), Smets and Wouters (2007), Gali (2008), Fernandez-Villaverde, Gordon, Guerron-Quintana, and Rubio-Ramirez (2015), Maliar and Maliar (2015).

where α is a parameter in the production function of the final-good firm. Appendix A.8 derives the following equilibrium equations:

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

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

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

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

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

$$r_t = \max(z_t, 0), \tag{35}$$

where $\chi_{t,1}$ and $\chi_{t,2}$ are defined in Appendix A.8, and π^*, 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.[32] 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, π_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 $y_{s+\Delta_s} \equiv y^*$, where χ_1^*, χ_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 β_s. That is, (30), (31), and (33) are defined as

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

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

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

where

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

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

[32] In our GAMS code, we use dnlp to deal with the nonlinear programming problem with discontinuous derivatives caused by (35). In our Matlab code, we use fsolve to solve the system of equations, where fsolve uses a finite difference method to smooth out the kink in (35).

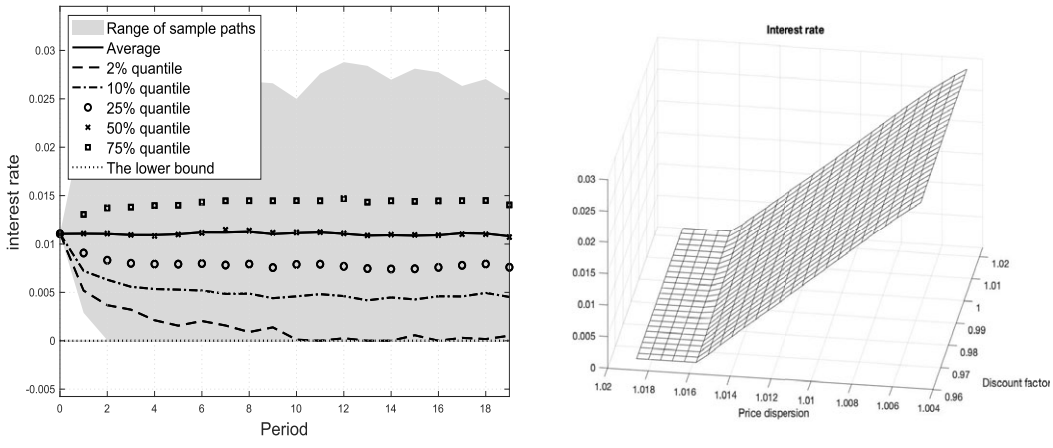

FIGURE 3. Simulated interest rates and the policy function for interest rate.

We assume that the time of interest, T^*, is 20 periods. We generated 1000 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 γ, then γ 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 γ 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 the time iteration have to wait until it converges before obtaining optimal value or policy functions, and the GSSA and 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 error at the initial state. If it is not small (with a large Δ_s), then we switch to other solution methods. That is, a small Euler 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 errors along the small number of simulated paths. If they are not small (with a large Δ_s), then we switch to other solution methods. These two methods can also be applied to check if the SCEQ code has bugs or Δ_s is large enough: we suggest using the same code but changing the variances of the random variables ϵ_t to be nearly zero.[33] Since SCEQ solves deterministic models very accurately, if the computed Euler errors for the nearly-zero-variance case are not nearly zero, it implies that there are bugs in the code or that Δ_s is not large enough (i.e., the truncation error is large). 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 1000 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. Moreover, we may use the SCEQ solution to test which approximation method

[33] If the problem is stationary and the initial state is equal or close to the nonstochastic 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 the Euler errors would always be small.

in other methods would be suitable: For example, if we use complete Chebyshev polynomials, then we may test how the high degree would be required for approximation.

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 with acceptable accuracy. 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 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, Fréderic 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, https://www.dynare.org. [658]

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. [655]

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. [676]

Bertsekas, Dimitri P. (2005), *Dynamic Programming and Optimal Control*, Vol. I. Athena Scientific, Nashua. [655]

Bertsekas, Dimitri P. (2007), *Dynamic Programming and Optimal Control*, Vol. II. Athena Scientific, Nashua. [655]

Blanchard, Olivier Jean and Charles M. Kahn (1980), "The solution of linear difference models under rational expectations." *Econometrica*, 48 (5), 1305–1311. [651]

Boucekkine, Raouf (1995), "An alternative methodology for solving nonlinear forward-looking models." *Journal of Economic Dynamics and Control*, 19 (4), 711–734. [653]

Brumm, Johannes and Simon Scheidegger (2017), "Using adaptive sparse grids to solve high-dimensional dynamic models." *Econometrica*, 85 (5), 1575–1612. [655, 658]

Cai, Yongyang (2019), "Computational methods in environmental and resource economics." *Annual Review of Resource Economics*, 11, 59–82. [655, 658, 663]

Cai, Yongyang and Kenneth L. Judd (2013), "Shape-preserving dynamic programming." *Mathematical Methods of Operations Research*, 77 (3), 407–421. [658]

Cai, Yongyang, and Kenneth L. Judd (2023), "Supplement to 'A simple but powerful simulated certainty equivalent approximation method for dynamic stochastic problems'." *Quantitative Economics Supplemental Material*, 14, https://doi.org/10.3982/QE1835. [666]

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, https://www.hoover.org/research/social-cost-carbon-economic-and-climate-risk. [652, 655, 662, 665, 675, 676, 677, 678, 681]

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. [654, 656, 662, 663, 665, 679]

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. [652, 655, 662, 665, 675, 676, 677, 678, 681]

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. [651]

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. [663]

Czyzyk, Joseph, Michael P. Mesnier, and Jorge J. More (1998), "The NEOS server." *IEEE Computational Science & Engineering*, 5, 68–75. [659]

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. [670]

Drud, Arne S. (1994), "CONOPT—A large scale GRG code." *ORSA Journal on Computing*, 6, 207–216. [659, 663, 667]

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. [659, 676]

Fair, Ray C. and John B. Taylor (1983), "Solution and maximum likelihood estimation of dynamic nonlinear rational expectations models." *Econometrica*, 51 (4), 1169–1185. [653]

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. [679]

Fernandez-Villaverde, Jesus and Oren Levintal (2018), "Solution methods for models with rare disasters." *Quantitative Economics*, 9, 903–944. [655]

Fernandez-Villaverde, Jesus, Juan Rubio-Ramirez, and Frank Schorfheide (2016), "Solution and estimation methods for DSGE models." In *Handbook of Macroeconomics*, Vol. 2 (John B. Taylor and Harald Uhlig, eds.), 527–724, Chapter 9. Elsevier, Amsterdam. [655]

Fisher, Paul G., Sean Holly, and Andrew J. Hughes Hallett (1986), "Efficient solution techniques for dynamic non-linear rational expectations models." *Journal of Economic Dynamics and Control*, 10 (1), 139–145. [653]

Fourer, Robert, David M. Gay, and Brian W. Kernighan (1990), "A modeling language for mathematical programming." *Management Science*, 36 (5), 519–554. [667]

Gali, Jordi (2008), *Monetary Policy, Inflation, and the Business Cycle: An Introduction to the New Keynesian Framework*. Princeton University Press, Princeton, NJ. [679]

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. [659]

Gilli, Manfred and Giorgio Pauletto (1997), "Sparse direct methods for model simulation." *Journal of Economic Dynamics and Control*, 21 (6), 1093–1111. [653]

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. [659, 662]

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. [655, 679]

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. [651]

Hall, Stephen G. (1985), "On the solution of large economic models with consistent expectations." *Bulletin of Economic Research*, 37 (2), 157–161. [653]

Jin, He-Hui and Kenneth L. Judd (2002), "Perturbation methods for general dynamic stochastic models." Working paper, Stanford University. [651, 655, 665, 666]

Judd, Kenneth L. (1992), "Projection methods for solving aggregate growth models." *Journal of Economic Theory*, 58 (2), 410–452. [651]

Judd, Kenneth L. (1998), *Numerical Methods in Economics*. MIT Press, Cambridge. [651, 655]

Judd, Kenneth L. and Sy-Ming Guu (1993), "Perturbation solution methods for economic growth models." In *Economic and Financial Modeling With Mathematica* (Hal R. Varian, ed.), 80–103, Springer, New York, NY. [651, 655]

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. [651, 658]

Judd, Kenneth L., Lilia Maliar, and Serguei Maliar (2012), "Merging simulation and projection approaches to solve high-dimensional problems." Tehcnical Report w18501, National Bureau of Economic Research, https://www.nber.org/papers/w18501. [651, 658, 675]

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. [656, 658]

Juillard, Michel, Douglas Laxton, Peter McAdam, and Hope Pioro (1998), "An algorithm competition: First-order iterations versus Newton-based techniques." *Journal of Economic Dynamics and Control*, 22 (8), 1291–1318. [653]

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. [655]

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. [655, 665, 666]

Kydland, Finn E. and Edward C. Prescott (1982), "Time to build and aggregate fluctuations." *Econometrica*, 50 (6), 1345–1370. [651]

Lenton, Timothy M., Hermann Held, Elmar Kriegler, Jim W. Hall, Wolfgang Lucht, Stefan Rahmstorf, and Hans J. Schellnhuber (2008), "Tipping elements in the Earth's climate system." *Proceedings of the National Academy of Sciences*, 105 (6), 1786–1793. [676]

Levintal, Oren (2018), "Taylor projection: A new solution method for dynamic general equilibrium models." *International Economic Review*, 59, 1345–1373. [655]

Ljungqvist, Lars and Thomas J. Sargent (2018), *Recursive Macroeconomic Theory*. MIT Press, Cambridge. [655]

Magill, Michael J. P. (1977), "A local analysis of N-sector capital accumulation under uncertainty." *Journal of Economic Theory*, 15 (1), 211–219. [651]

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. [657, 658, 665, 675, 679]

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. [662]

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. [656]

Marcet, Albert and Guido Lorenzoni (1998), "Parameterized expectations approach: Some practical issues." In *Computational Methods for Study of Dynamic Economies* (Ramon Marimon and Andrew Scott, eds.), 143–171, Oxford University Press, New York. [651]

McCarl, Bruce, Alex Meeraus, Paul van der Eijk, Michael Bussieck, Steven Dirkse, and Franz Nelissen (2016), McCarl Expanded GAMS User Guide Version 24.6, https://www.gams.com/mccarlGuide. Accessed on December 19, 2020. [659, 667]

Miranda, Mario J. and Paul L. Fackler (2002), *Applied Computational Economics and Finance*. MIT Press, Cambridge. [651, 655]

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. [679]

Nordhaus, William D. (2008), *A Question of Balance: Weighing the Options on Global Warming Policies*. Yale University Press, New Haven. [652, 662, 676]

Nordhaus, William D. (2017), "Revisiting the social cost of carbon." *Proceedings of the National Academy of Sciences of the United States of America*, 114 (7), 1518–1523. [676, 677]

Rust, John (1996), "Numerical dynamic programming in economics." In *Handbook of Computational Economics*, Vol. 1 (Hans M. Amman, David A. Kendrick, and John Rust, eds.), 619–729, Elsevier, Amsterdam. [655]

Rust, John (1997), "Using randomization to break the curse of dimensionality." *Econometrica*, 65 (3), 487–516. [658]

Rust, John (2019), "Has dynamic programming improved decision making?" *Annual Review of Economics*, 11 (1), 833–858. [658]

Scheidegger, Simon and Ilias Bilionis (2019), "Machine learning for high-dimensional dynamic stochastic economies." *Journal of Computational Science*, 33, 68–82. [658]

Smets, Frank and Rafael Wouters (2007), "Shocks and frictions in US business cycles: A Bayesian DSGE approach." *American Economic Review*, 97 (3), 586–606. [679]

Smolyak, Sergei A. (1963), "Quadrature and interpolation formulas for tensor products of certain classes of functions." *Soviet Mathematics Doklady*, 4, 240–243. [656]

Woodford, Michael (2003), *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton University Press, Princeton, NJ; Woodstock, Oxfordshire, England. [679]

Co-editor Tao Zha handled this manuscript.

Manuscript received 8 February, 2021; final version accepted 19 December, 2022; available online 31 January, 2023.