Numerical Dynamic Programming with Verification and Uncertainty Quantification: An Application to Climate Policy

Yongyang Cai* Kenneth L. Judd†

March 1, 2018

This paper introduces novel techniques for solving multidimensional dynamic stochastic optimization problems which commonly occur in economics and related disciplines. We (i) present a new efficient method to approximate value functions, (ii) develop an error control scheme that allows us to verify the accuracy of value function iterations, and (iii) quantify the dependence on parameters by computing response surfaces, a basic tool for uncertainty quantification. This is all possible because our effecient use of parallel dynamic programming methods to solve those types of models. As an application of our methodology, we consider a prominent problem in environmental economics—determining the optimal policy for addressing potential adverse impacts of carbon emissions on economic output. Using plausible ranges for several key parameters we obtain an optimal carbon price ranging between $40 and $400 per ton. Our study shows that with the appropriate methods, complex problems of optimal decision making under various types of uncertainties can be addressed, and decisively refutes the pessimism one often hears about the possibility of solving such models.

*Ohio State University
†Hoover Institution, Stanford University & NBER, kennethjudd@mac.com.

1 Introduction

Dynamic programming (DP) is the foundation of dynamic economics and has been applied to a large range of fields in economics and finance (e.g., Bellman 1957; Lucas et al. 1989; Puterman 2005). Problems with a finite number of states and choices can be solved exactly, as can some with special functional form assumptions. However, this covers only a small portion of the range of DP problems in economics, finance and operations research, as well as in other fields. In general, the solution to a DP problem with continuous states can only be approximated. This paper combines modern numerical methods with the advanced massively parallel hardware to solve a DP problem from the climate change literature that has been viewed as impossible.

Our DP problems are in a finite horizon evolving system and have expanding state domains over time. Value function iteration is the traditional and most popular method for solving finite-horizon decision-making problems and it can also be used for solving infinite-horizon problems after setting a stopping criterion (see e.g., Judd 1998; Cai 2010). One basic method in value function iteration is to transform the problem into one with a finite number of states, but that method only works well for small problems. Another technique is to solve the DP problem at a finite grid of points, and using multilinear interpolation to approximate the value function at points not in the grid. This approximation method is not very efficient if the true value function is smooth- an observation which motivated Johnson et al. (1993) to instead interpolate using tensor-product cubic splines. To further reduce the computational costs, Chen et al. (1999) combined orthogonal arrays with multivariate adaptive regression splines (Friedman 1991) to efficiently solve multidimensional problems.

It is well-known that there there is a curse of dimensionality for general DP problems (see Rust 1997 and Rust et al. 2002). Fortunately, such worst-case analyses are not relevant for many problems in economics. A key task in solving DP problems is approximating the value function. There is a curse of dimensionality for the general problem of approximating multivariate continuous-functions but Griebel and Wozniakowski (2006) show that there is no curse of dimensionality in approximating functions with sufficient smoothness. Many DP problems in economics have smooth value functions, making

the Griebel-Wozniakowski approach more relevant than the approaches to general DP problems.

But for DP problems with continuous state and action variables and smooth value functions, one criticism for value function iteration in the literature is that it lacks an error control scheme if we use continuous function approximation on value functions, because the value function approximation may not preserve concavity of true value function (Cai and Judd 2013, 2014, 2015). This paper introduces basic verification methods for the value function iteration approach and provide tools to assess the accuracy of solution to address the criticism.

In economics, many studies rely on models that can be solved analytically. This has the advantage of allowing an accurate sensitivity analysis of these models. However, simple models that can be solved analytically are severely limited by whether they can realistically account for the problem they study. In contrast, numerical methods allow for formulating and solving much more complex models in economics, models which incorporate—for example—multidimensionality, non-linearities and uncertainty in the decision making process. However, typically in numerical work, economists employ strong simplifications of their models. Those simplifications include the log-linearization of the underlying dynamic systems, the reduction of the number of decision periods, the restriction to a simple steady-state analysis. One key reason for those simplifications is computational limitation. This is still a widely accepted argument for not engaging in more serious (computational) research. While it is true that past analyses have been hampered by both software and hardware limitations, advances over the past twenty years now make it possible to examine important economic questions without making unappealing assumptions for the sake of tractability. Currently, mainstream research in economics does not live up to the standards of of numerical algorithms and computational hardware.

However, Cai et al. (2015d) provide a parallel DP scheme on a computational grid for solving large-scale dynamic optimization problems that require a very large amount of computing resources including CPU time and memory, making them impossible to be solved on personal computers. It is run on a high-throughput system called HTCondor, a management tool for identifying, allocating and managing available resources to solve

large distributed computations on existing networks of computers that currently exist in department and college networks. While HTCondor is freely available, its latency in communication poses a limit in running huge applications requiring a large amount of compute cores and communication among them.

Supercomputing is a high-performance computing tool on a supercomputer allowing rapid communication among processors. Modern day supercomputers have petascale computing power and even exascale computing will be available in a near future, so that we can apply parallel DP algorithms on supercomputers to solve the large-scale DP problems. Our work is mainly implemented on the Blue Water supercomputer, located in the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We adapt the master-worker parallel DP algorithm in Cai et al. (2015d) to its supercomputing version. Our code is written in Fortran and uses MPI (message passing interface) for communication among processes.

Moreover, parallelism makes it possible to conduct extensive explorations of the parameter space and ascertain the sensitivity of any results to parameters about which we have limited information, as we often do not know the true parameters of a real world problem, a fact which calls for examining more than just one case. This method—also called uncertainty quantification—is standard in many disciplines, such as engineering and applied physics.[1] In economics, numerical studies have been limited (for computational reasons) in their analysis of how their results depend on uncertain parameters. Yet, addressing parametric uncertainty is important for any economic study as in most cases, we do not know the true parameter values. Disagreements over models, data, and estimation procedures tell us that the uncertainty is greater than the standard error computed for any single model. We show that the combination of hardware and software used in this paper has the ability to do nontrivial multidimensional uncertainty quantification by resolving the same DP problem over an efficiently chosen set of points in the parameter space. Finally, the tools do not exploit any special structure of the example and clearly can be used to solve a large array of DP problems that arise in economics.

[1]There are alternative methods to deal with parameter uncertainty under different assumptions, such as Bayesian learning (see e.g., Kelly and Kolstad 1999) and robust decision making (see e.g., Hansen and Sargent 2005, 2007a,b; Brock et al. 2007a,b; Anderson et al. 2014; Cai and Sanstad 2015). But these methods are more time-consuming and challenging and also out of the scope of this paper, we do not pursue them.

The main purpose of this study is to offer a set of tools for modelers interested in solving complex dynamic models of various types of uncertainties. For that purpose we provide various computational techniques and implement them with software and algorithms that can handle a wide rage of problems. We introduce a novel simplicial complete Chebyshev approximation that is much more efficient than complete Chebyshev approximation. We also present novel methods of verifying the accuracy of the computations, such as a method of a posteriori error measurement, for solving thousands of multidimensional DP problems with high accuracy, a large state space, and hundreds of decision periods. In addition, we address the analysis of substantial uncertainty about the key parameters, and extend the DP algorithm to allow performing an extensive uncertainty quantification. This will allow researchers to ascertain the implications of their preferred parameter choices. Furthermore, we demonstrate how to utilize the finding from an extensive uncertainty quantification to fit a multi-dimensional response surface. This is conducted by a Smolyak approximation method (Smolyak 1963) which we find to be efficient. In contrast to Bayesian methods for which moments from a single belief are interesting only to those who have those beliefs, a response surface can be used by anyone using their own beliefs and computing the corresponding moments.

The novel methods presented in this paper are highly modular at many levels, such as the length of a time period, the degree of aggregation of the model (sub)systems, and the computational components which can be utilized. As a consequence, these methods are ready to be basically applied to any DP problems, including far more complex problems of optimal decision making in various areas of economics, finance, and decision science in general. One field of application is macroeconomics, in which most analyses work with models where none of the primitives depend on calendar time. Our methods allow researcher to also solve non-stationary problems such as ones with unit roots (e.g., Cai et al. 2015a). Other examples of applications of our methods include multi-dimensional stochastic optimal growth models (e.g., Den Haan et al. 2011), stochastic life-cycle consumption problems (e.g., Cocco et al. 2005), and dynamic portfolio optimization problems (e.g., Cai et al. 2013; Gupta and Murray 2005; Infanger 2006).

Another important feature is the flexibility of our algorithm. It might be possible to solve a specific application with a simpler and model-specific algorithm. That algorithm

however, will have little value for any substantively different applications. For example we only assume concave utility and general laws of motion for states. Our software thus, may be more than what is needed for our application below. However, we have borne a large fixed cost and created a tool that can now be utilized to solve a vast array of problems. Different models then become just different parameter choices and different simple functions for utility and output.

We next turn to a specific application of our methods. Our objective is to determine optimal carbon pricing in a canonical dynamic integrated model of the climate and economy. One crucial factor of our application is that we incorporate intrinsic and parametric uncertainty.

Typically cost-benefit computations of carbon emissions are conducted with the help of so-called integrated assessment models that explicitly model the interactions between climate and economic systems. Integrated assessment models usually have many state variables and thus dimensions. It is often argued that a stochastic analysis of a 6-dimensional integrated assessment model over several hundred decision periods are computationally too expensive—if not impossible—due to the curse of dimensionality or other computational complexities (e.g., Webster et al. 2012; Jensen and Traeger 2014). In fact, recently Newbold et al. (2013) define low-dimensional analyses of uncertainty as the frontier of integrated assessment modeling. Such a clear assessment of the research frontier in climate economics makes the latter a suitable application for our methodology. With the example of computing optimal climate policies under uncertainty we challenge this assessment and show that when we combine modern methods from numerical mathematics with the state-of-the art hardware, it is now possible to solve dynamic programming problems that are currently viewed as impossible.

For our analysis, we take an integrated assessment model, DSICE (Cai et al. 2015a), a full-dimensional dynamic stochastic general equilibrium version of DICE (Nordhaus 2008), a well-known model currently used by the United States government to design climate policy (IWG 2010). Not only is the DSICE framework a stochastic extension of earlier versions of DICE, but also DSICE can be used to solve many integrated assessment models of comparable dimensionality. No computational framework is infinitely powerful, but it is clear that DSICE is far less limited by tractability concerns than earlier integrated

assessment models, because our computational method is flexible enough to solve far larger models when using parallel computing. In that context, our algorithm is fast enough and will enable us to do wide-ranging parameter sweeps to address parameter uncertainty.

We run several tests to measure the performance of our solution algorithm. Our first step is to test if our numerical DP algorithm can replicate the true solution of the degenerated deterministic DSICE example. We find that the true solution is approximated with high accuracy. For the stochastic model version with default parameter settings we compute stepwise errors for the policy functions and the value function. We find that the relative errors are small.

We are limited by the current state of knowledge about the critical parameters about the economics and climate system. Also, different decision-makers usually have different beliefs about the implications of parameter vectors. In a recent assessment of of the literature, Pindyck (2013) points out that uncertainty in integrated assessment models, if incorporated at all, is usually analyzed by running Monte Carlo simulations in which probability distributions are attached to one or more parameters. Therefore, in a next step we perform an uncertainty quantification analysis to deal with parametric uncertainty. In accordance with the literature, we identify four key features which are not known with precision: climate sensitivity, utility discount rate, growth rate of productivity and the damage function. Our uncertainty quantification analysis produces a range of the social cost of carbon between \$21 per ton and \$1,141 per ton (in 2005 USD). Moreover, from uncertainty quantification we produce a response surface function to give an estimated initial social cost of carbon (SCC) over the four-dimensional hypercube, so that policy makers with various beliefs about the uncertain parameters can immediately know the corresponding initial SCC.

The structure of this paper is as follows: In Section 2 we briefly review numerical dynamic programming methods. Section 3 introduces the new simplicial complete Chebyshev approximation method. Section 4 presents the new error control methods . Section 5 describes the algorithms for uncertainty quantification and surface response function. Section 6 reviews DSICE briefly and presents its dynamic programming problem. Section 7 describes various uncertain parameters in DSICE for uncertainty quantification.

Section 8 illustrates the results of the benchmark example of DSICE. Section 9 shows the results of uncertainty quantification for DSICE. Section 10 concludes.

2 Numerical Dynamic Programming

A dynamic programming (DP) problem is expressed in the Bellman equations that define value functions (Bellman 1957). Let $\mathbf{x}$ be a vector of continuous states. Let $V_t(\mathbf{x})$ denote the value function at time $t \leq T$, where the terminal value function $V_T(\mathbf{x})$ is known. Let F_t be the transition laws of $\mathbf{x}$ at time t, and the next-stage state is denoted $\mathbf{x}^+$. Let $\mathbf{a}$ be a vector of action/control variables and it should be in a feasible set $\mathcal{D}(\mathbf{x}, t)$ at time t. Let $u_t(\mathbf{x}, \mathbf{a})$ be the utility function at time t. The Bellman equation is

$$\begin{aligned} V_t(\mathbf{x}) = \mathcal{G}(V_{t+1})(\mathbf{x}) \quad \equiv \quad & \max_{\mathbf{a}\in\mathcal{D}(\mathbf{x},t)} \quad u_t(\mathbf{x}, \mathbf{a}) + \beta \mathcal{H}_t \left(V_{t+1}(\mathbf{x}^+) \right), \\ & \text{s.t.} \quad \mathbf{x}^+ = F_t(\mathbf{x}, \mathbf{a}, \omega), \end{aligned}$$

where $\mathcal{G}$ is the Bellman operator, ω is a vector of random variables, β is a discount factor, and $\mathcal{H}_t$ is a functional operator at time t. A typical functional operator for $\mathcal{H}_t$ is $\mathbb{E}_t$, the expectation operator conditional on time-t information. In our model, it is

$$\mathcal{H}_t \left(V_{t+1} \right) = \left[\mathbb{E}_t \left\{ V_{t+1}^{\frac{1-\gamma}{1-\frac{1}{\psi}}} \right\} \right]^{\frac{1-\frac{1}{\psi}}{1-\gamma}}.$$

We first present a general overview of the numerical methods we use. For DP problems, if state variables and control variables are continuous such that value functions are also continuous, then we have to use some approximation for the value functions, since computers cannot model the entire space of continuous functions. We focus on using a finitely parameterized collection of functions to approximate value functions, $V(\mathbf{x}) \approx \hat{V}(\mathbf{x}; \mathbf{b})$, where $\mathbf{b}$ is a vector of parameters. The functional form $\hat{V}$ may be a linear combination of polynomials, or it may represent a rational function or neural network representation, or it may be some other parameterization especially designed for the problem. After the functional form is fixed, we focus on finding the vector of parameters, $\mathbf{b}$, such that $\hat{V}(\mathbf{x}; \mathbf{b})$ approximately satisfies the Bellman equation. Numerical DP with

value function iteration can solve the Bellman equation approximately (Judd 1998).

The following is the algorithm of parametric dynamic programming with value function iteration for finite horizon problems.

Algorithm 1. *Numerical Dynamic Programming with Value Function Iteration for Finite Horizon Problems*

Initialization. *Choose the approximation nodes,* $\mathbb{X}_t = \{\mathbf{x}_{i,t} : 1 \leq i \leq N_t\} \subset \mathbb{R}^d$ *for every* $t < T$*, and choose a functional form for* $\hat{V}(\mathbf{x}; \mathbf{b})$*. Let* $\hat{V}(\mathbf{x}; \mathbf{b}_T) \equiv V_T(\mathbf{x})$*. Then for* $t = T-1, T-2, \ldots, 0$*, iterate through steps 1 and 2.*

Step 1. *Maximization step. Compute*

$$v_i = \max_{\mathbf{a} \in \mathcal{D}(\mathbf{x}_i, t)} u_t(\mathbf{x}_i, \mathbf{a}) + \beta \mathcal{H}_t \left(\hat{V} \left(\mathbf{x}^+; \mathbf{b}_{t+1} \right) \right) \tag{1}$$
$$\text{s.t.} \quad \mathbf{x}^+ = F_t(\mathbf{x}_i, \mathbf{a}, \omega),$$

for each $\mathbf{x}_i \in \mathbb{X}_t$*,* $1 \leq i \leq N_t$*.*

Step 2. *Fitting step. Using an appropriate approximation method, compute the* $\mathbf{b}_t$ *such that* $\hat{V}(\mathbf{x}; \mathbf{b}_t)$ *approximates* $(\mathbf{x}_i, v_i)$ *data.*

There are three main components in numerical DP: optimization, approximation, and numerical integration. In the following we focus on discussing approximation and omit the introduction of optimization and numerical integration. Detailed discussion of numerical DP can be found in Cai (2010), Cai and Judd (2014, 2015), Judd (1998), and Rust (2008). See Appendix B for an additional discussion of approximation. In the next section, we will introduce a multidimensional simplicial complete Chebyshev approximation method.

There are some alternative approaches for solving DP problems. For example, approximatie dynamic programming methods (see e.g. Bertsekas 2012; Powell 2011; Powell and Van Roy 2004) are designed to solve huge-dimensional problems by losing some accuracy based on simulation. An approximate DP initializes a value function approximation $V_t^0(\mathbf{x})$ at first. In its n-th iteration, with a given initial state $\mathbf{x}_0^n$, it computes $v_t^n = \mathcal{G}(V_{t+1}^{n-1})(\mathbf{x}_t^n)$ and its corresponding optimal action, and then get the next state using a sample realization of innovation. For a problem with discrete states, a basic

approximate DP lets $V_t^n(\mathbf{x}) = V_t^{n-1}(\mathbf{x})$ for all $\mathbf{x} \neq \mathbf{x}_t^n$ and $V_t^n(\mathbf{x}_t^n) = v_t^n$. However, for a problem with continuous actions, if it uses a continuous value function approximation $V_t^n(\mathbf{x})$, then we need to solve a nonlinear programming problem for the Bellman equation $V_t^{n+1} = \mathcal{G}(V_t^n)$, but it is possible that $V_t^n(\mathbf{x})$ does not preserve the shape of value function (i.e., $V_t^n(\mathbf{x})$ may be not concave), so that it is hard to get the global maximizer of $\mathcal{G}(V_t^n)$ (see Cai and Judd (2014) for a discussion about the importance of shape-preservation in DP). That is, the approximate DP may not converge for a problem with continuous actions if it uses a continuous value function approximation, while our numerical DP works well for problems with continuous actions (Cai 2010; Cai and Judd 2014, 2015). Thus, approximate DP methods are typically used for solving problems with discrete actions. See Bertsekas (2012) and Powell (2011) for detailed discussion.

Other approaches for DP problems include policy function iteration (see e.g. Judd 1998) for infinite-horizon problems; stochastic programming methods (see e.g. Kall and Wallace 1994) for several-stage problems; linear programming methods (De Farias and Van Roy 2003; Trick and Zin 1997) and nonlinear programming methods (Cai et al. 2015c) for low-dimensional problems. Due to the limits of these approaches, we apply parallel value function iteration (Cai et al. 2015d) in this paper as it is the most efficient one for our climate policy problems to the best of our knowledge.

3 Simplicial Complete Chebyshev Approximation

The complete degree n Chebyshev polynomial approximation method (described in Appendix B) is an isotropic approximation method; that is, the polynomial complexity of the function is the same in any direction. In a d-dimensional state space, the number of terms of the complete Chebyshev approximation is only $\binom{n+d}{d}$, a polynomial of d with the maximal degree n. Thus, it has no so-called curse-of-dimensionality and then it is tractable for some large problems. For example, the number of terms for quadratic polynomial approximation in 100-dimensional state space is only 5,151. Unfortunately, the growth in n limits the practicality of the complete polynomials. For example, a complete cubic polynomial in 100 dimensions has 176,851 terms. In the following we introduce a simplicial complete approximation to reduce the number of terms significantly.

3.1 Basis for Simplicial Complete Chebyshev Approximation

In many problems the value function is significantly nonlinear in some dimensions but much less nonlinear in others. This allows us to use different polynomial degrees in different dimensions, leading to an anisotropic approximation that is a subset of complete polynomials.

We describe a particular kind of anisotropic approximation, which we call the simplicial complete approximation, that works well in our climate change policy examples in later sections, and has substantial potential for more general dynamic programming problems. Suppose the maximal degree for dimension i is n_i, then the simplicial complete Chebyshev approximation for $V(\mathbf{x})$ is

$$\hat{V}(\mathbf{x};\mathbf{b}) = \sum_{\alpha \geq 0, \sum_{i=1}^{d} \alpha_i / n_i \leq 1} b_\alpha \phi_\alpha \left(Z(\mathbf{x}) \right),$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$ represents a nonnegative integer vector, and $\phi_\alpha(Z(\mathbf{x}))$ is a Chebyshev basis function in $\mathbb{R}^d$ where $Z(\mathbf{x})$ is a linear transformation of $\mathbf{x} \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}] \subset \mathbb{R}^d$ so that $Z(\mathbf{x}) \in [-1,1]^d$ (see Appendix C for details). Here, $\sum_{i=1}^{d} \alpha_i / n_i \leq 1$ represents those points with nonnegative integer values below or on the hyperplane $\sum_{i=1}^{d} x_i / n_i = 1$ that contains points $(n_1, 0, ..., 0)$, ..., $(0, ..., 0, n_d)$.

3.2 Regression for Simplicial Complete Chebyshev Approximation

After we choose the simplicial complete Chebyshev approximation for $V(\mathbf{x})$ with degrees $(n_1, ..., n_d)$, we know how many terms are used for the approximation. We denote the number of approximation terms by J. In the numerical dynamic programming algorithm, we should choose approximation nodes in the approximation domain, and then use the values on the nodes to compute the Chebyshev coefficients $\mathbf{b}$. For the number of approximation nodes N, we only require $N \geq J$ for Lagrange fitting, or $N \geq J/(d+1)$ for Hermite fitting (Cai and Judd 2015). But usually N should be much bigger than J (or $J/(d+1)$ for Hermite fitting) in order to have a globally good approximation. A general fitting method is the least squares method, but it will be too time-consuming when N and/or J are large.

To compute the Chebyshev coefficients $\mathbf{b}$ quickly, we can choose m_i Chebyshev nodes $\left\{x_i^{(k_i)} : k_i = 1, \ldots, m_i\right\}$ in the i-th dimension interval $[x_{\min,i}, x_{\max,i}]$ to construct the simplicial complete Chebyshev polynomial approximation directly. Thus, the number of approximation nodes is $N = \prod_{i=1}^{d} m_i$. Usually it is good to let $m_i = n_i + 1$ if the degree for the i-th dimension is n_i. With the nodes $\mathbf{x_k} = \left(x_1^{(k_1)}, \ldots, x_d^{(k_d)}\right)$ for $\mathbf{k} = (k_1, \ldots, k_d)$ and $k_i = 1, \ldots, m_i$, we can compute the coefficients of the simplicial complete Chebyshev approximation for $V(\mathbf{x})$ on $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$:

$$b_\alpha = \frac{2^{\tilde{d}}}{\prod_{i=1}^{d} m_i} \sum_{1 \le k_i \le m_i, 1 \le i \le d} v_{\mathbf{k}} \phi_\alpha(Z(\mathbf{x_k})), \tag{2}$$

where $v_{\mathbf{k}} = V(\mathbf{x_k})$, and $\tilde{d} = \sum_{i=1}^{d} 1_{\alpha_i > 0}$ with

$$1_{\alpha_i > 0} = \begin{cases} 1, & \text{if } \alpha_i > 0, \\ 0, & \text{if } \alpha_i = 0. \end{cases}$$

3.3 Efficiency of Simplicial Complete Chebyshev Approximation

From the maximization step of Algorithm 1, we see that the efficiency of the numerical dynamic programming algorithm depends on the number of approximation nodes, N. Moreover, the objective function of the maximization problem contains evaluation of the next-period approximated value function, so the efficiency of the algorithm also depends on the number of approximation terms, J. With a rough estimate, the computational load of the algorithm is proportional to NJ.

Since every approximation term $b_\alpha \phi_\alpha (Z(\mathbf{x}))$ has a product of d one-dimensional basis functions for ϕ_α (see Appendix C), the computational complexity of the algorithm seems to be proportional to $N \times J \times d$. However, the one-dimensional basis functions include a degenerated degree-zero function which is one everywhere, so the number of non-degenerated multiplications for $b_\alpha \phi_\alpha (Z(\mathbf{x}))$ will also not exceed $n = \max_{i=1}^{d} n_i$ for the simplicial complete Chebyshev approximation. Therefore, the computational complexity is proportional to $N \times J \times \min(n, d)$. This means that if we had a high-dimensional problem but could use the same number of basis functions and nodes for a low-dimensional problem, then the high-dimensional problem could have less computational complexity

than the low-dimensional problem. Since we only require $N \geq J$ for Lagrange fitting, or $N \geq J/(d+1)$ for Hermite fitting (Cai and Judd 2015), we can choose sparse grid to break the curse-of-dimension in choosing approximation nodes. That is, our numerical DP method could have no curse-of-dimension in both approximation methods and nodes.

Table 1 lists the numbers of approximation terms and nodes (i.e., J and N) for some cases in a six-dimensional continuous state space (i.e., $d = 6$) which is used in this paper. The first six columns are the degrees $n_1, ..., n_6$ for each dimension. For the approximation nodes, we choose the tensor grid of $(n_i + 1)$ Chebyshev nodes in the i-the dimension for simplicity, so $N = \prod_{i=1}^{6}(n_i + 1)$. The last column lists the speedup relative to corresponding complete Chebyshev polynomials with the highest degree among $n_1, ..., n_6$, i.e.,

$$\frac{(n+1)^6 \times \binom{n+6}{6}}{NJ} = \frac{(n+6)!(n+1)^6}{n!6!NJ} = \frac{(n+6)(n+5)(n+4)(n+3)(n+2)(n+1)^7}{720NJ}$$

where $n = \max_{i=1}^{6} n_i$. Here we also use the tensor approximation grid (each dimension has $(n+1)$ Chebyshev nodes) for fitting the degree-n complete Chebyshev polynomials with the formula (2).

$\max\{n_1, ..., n_6\}$	n_1	n_2	n_3	n_4	n_5	n_6	J	N	relative speedup
4	4	4	4	4	4	4	210	15,625	1
	4	2	2	2	2	2	35	1,215	77
6	6	6	6	6	6	6	924	117,649	1
	6	**6**	**6**	**4**	**4**	**2**	**267**	**25,725**	**16**
	6	6	4	4	4	2	204	18,375	29
	6	4	4	4	4	2	165	13,125	50
	6	4	4	4	2	2	116	7,875	119
	6	4	4	2	2	2	81	4,725	284
	6	4	2	2	2	2	57	2,835	673
	6	2	2	2	2	2	42	1,701	1,522
8	8	8	8	8	8	8	3,003	531,441	1
	8	6	6	4	4	2	310	33,075	156
10	10	10	10	10	10	10	8,008	1,771,561	1
	10	6	6	4	4	2	352	40,425	997

Table 1: Numbers of terms/nodes for simplicial complete Chebyshev approximations. Here, $n_1, ..., n_6$ are degrees in six dimensions, J is the number of approximation term, and N is the number of approximation nodes.

The row with bold numbers in Table 1 gives the degrees we use for our numerical examples (we use a decreasing order for convenience because of the symmetry, while the

order used in our examples is $(6, 6, 4, 2, 6, 4)$), and the number of approximation terms is only $J = 267$. But the number of terms of a degree-6 complete Chebyshev polynomial in the six-dimensional space is 924 (the row with degrees (6,6,6,6,6,6)), about 3.5 times the number of the simplicial complete Chebyshev approximation we use. Moreover, we will only need to solve $N = 25,752$ optimization problems (1) for each discrete state and stage, but if we use the same Chebyshev regression algorithm for computing coefficients of the complete degree-6 Chebyshev polynomials, it needs to solve $7^6 = 117,649$ optimization problems (1), about 4.6 times the number for the simplicial polynomial. Thus, the relative speedup of the simplicial polynomial is $3.5 \times 4.6 \approx 16$. That is, the numerical dynamic programming method using the simplicial complete Chebyshev approximation is roughly 16 times faster than the one using the complete degree-6 Chebyshev polynomials.

Table 1 tells us that for any given number of dimensions, if only a few dimensions need a high-order degree expansion, the number of approximation terms is not big and the relative speedup is large. For example, in the row with the degrees (10,6,6,4,4,2) (we have $n_1 = 10$ while the other dimensions are the same with the row with bold numbers), the number of approximation terms just increases to 352, but a degree-10 complete polynomial needs 8,008 terms (the last row), about 23 times higher. The slow growth of the number of approximation terms can also be seen from the other rows. Moreover, its relative speedup is up to 997.

Furthermore, the benefit from simplicial approximation is greater for higher-dimensional functions. For example, if we want to approximate a ten-dimensional function, and only one dimension needs a degree-10 approximation while the other nine dimensions will be well approximated with quadratic polynomials, then the number of terms of this simplicial approximation is only 110. But the number of terms for the degree-10 complete approximation is up to 184,756. Moreover, if we use the tensor grid for quick computation of coefficients with the formula (2), then the number of approximation nodes for this simplicial approximation is $11 \times 3^9 = 216,513$, but the number of tensor grid for the degree-10 complete approximation is up to $11^{10} = 2.6 \times 10^{10}$. Thus, the relative speedup of the simplicial approximation is up to 2×10^8.

4 Error Control Methods

Numerical DP algorithms are often criticized by lacking accuracy estimation and error control. We use three steps to estimate the errors of solutions from our numerical DP methods. The first step is testing if the computational method can replicate the solution of the corresponding deterministic model, the second step estimates stepwise approximation errors of policy/value functions in the backward value function iteration, and the last step checks errors in the forward simulation.

4.1 Replication of the Deterministic Solution

The deterministic model can be solved directly by an optimization solver as a large-scale optimal control problem. When we solve the deterministic problem with our numerical DP methods, we use the same approximation domains and methods, which are used for the stochastic problems, on purpose. Thus, if our numerical DP methods can replicate the solution of the deterministic model, then it tells us that our code is reliable. Moreover, for a stochastic problem, we can also let its volatility be very small so we know that its solution should be close to the deterministic one, and then we apply our methods to solve it for checking if it also works well.

4.2 Error Control in Value Function Iteration

While the above routines show that our methods work well for deterministic or nearly-deterministic problems, we also estimate the approximation errors of policy/value functions at every value function iteration step for the real stochastic cases. At time t, for every vector in the discrete state space, we approximate the policies as functions of the smooth states using the solutions for the optimal policy at the approximation nodes in Algorithm 1 as data. That is, we construct a vector of policy function approximations, $\widehat{\mathbf{a}}_t(\mathbf{x})$, from the optimal solutions $\mathbf{a}^*(\mathbf{x}_i)$ at nodes $\mathbf{x}_i \in \mathbb{X}_t$.

For each time t, the approximation error over the state space is obtained in the following way: for each discrete state, we draw 1,000 random points in the continuous state space, solve the maximization step for each point $\widetilde{\mathbf{x}}$ in the sample set X_t and get the optimal action vector $\mathbf{a}_t^*(\widetilde{\mathbf{x}})$, and then compare it with the values of the approximated

policy functions at the point, $\widehat{\mathbf{a}}_t(\widetilde{\mathbf{x}})$.

After we compute the approximation errors for all points (the number is 1,000 times the number of discrete state vectors), we compute the global approximation error for each policy function or the value function in some norm. The following formula gives the $\mathcal{L}^\infty$ global error for the i-th policy function $a_{i,t}(\mathbf{x})$ (the i-the element of the policy function vector $\mathbf{a}_t(\mathbf{x})$)

$$\max_{\widetilde{\mathbf{x}}\in X_t} \frac{\left|\widehat{a}_{i,t}(\widetilde{\mathbf{x}}) - a^*_{i,t}(\widetilde{\mathbf{x}})\right|}{1 + \left|a^*_{i,t}(\widetilde{\mathbf{x}})\right|} \tag{3}$$

Here we use an adjusted relative error measure as some solutions could be 0. We can use the relative error measure directly if we know that the policy function is not close to 0. The corresponding $\mathcal{L}^1$ global error is the average value of the objective of the above discrete maximization problem over $\widetilde{\mathbf{x}} \in X_t$.

Similarly, we compute the approximation errors for the value functions. In dynamic economics problems, an affine transformation of utility functions for all times will not change the optimal policy functions if the linear coefficient is positive. This implies that value functions could be arbitrarily large or small in magnitude after such an affine transformation. Thus an absolute or relative error measure for value function approximation is not enough to tell the approximation efficiency. To overcome the problem, we could use the following unit-free error measure:

$$\max_{\widetilde{\mathbf{x}}\in X_t} \frac{\left|\widehat{V}_t(\widetilde{\mathbf{x}}) - V^*_t(\widetilde{\mathbf{x}})\right|}{|\widetilde{\mathbf{x}}| \cdot |\nabla V^*_t(\widetilde{\mathbf{x}})|} \tag{4}$$

where $\nabla V^*_t(\widetilde{\mathbf{x}})$ is the gradient vector of V^*_t over $\widetilde{\mathbf{x}}$, and $|\widetilde{\mathbf{x}}| \cdot |\nabla V^*_t(\widetilde{\mathbf{x}})|$ is the inner product of $|\widetilde{\mathbf{x}}|$ and $|\nabla V^*_t(\widetilde{\mathbf{x}})|$.[2] However, in many economics problems, we will have a positive capital stock, K, as one state variable, and it makes more sense to give an error measure relative to capital. That is, (4) can be replaced by

$$\max_{\widetilde{\mathbf{x}}\in X_t} \frac{\left|\widehat{V}_t(\widetilde{\mathbf{x}}) - V^*_t(\widetilde{\mathbf{x}})\right|}{K \cdot \left|\frac{\partial}{\partial K} V^*_t(\widetilde{\mathbf{x}})\right|} \tag{5}$$

where the capital K is one variable in the vector $\widetilde{\mathbf{x}}$. In this paper, we use (5) for our

[2]When $\widetilde{\mathbf{x}}$ is near or at the origin point, this error measure will not work well, but in our examples and most economics problems, $\widetilde{\mathbf{x}}$ will not be close to the origin point.

error measure of value function approximation.

5 Uncertainty Quantification and Response Surface Function

In many applications, some parameter values are uncertain. Economists often use sensitivity analysis to address it. While each case is a DP problem, it will be very time consuming if we want to run extensive sensitivity analysis. Moreover, even after the solutions for a set of m values of uncertain parameters are provided for sensitivity analysis, sometimes policy makers want to know solutions at one specific parameter value which is not included in the set of m cases. We introduce the following response surface function method to give them a quick answer.

In the space of uncertain parameters, we choose a set of m approximation points which could be tensor grid if the number of uncertain parameters is small, solve their corresponding DP problems (these DP problems are independent each other, so they are naturally parallelizable), and then use their solutions (e.g., the initial optimal social cost of carbon in our climate policy examples) to construct an approximation to fit them. In this paper, we use a sparse Smolyak grid and Chebyshev-Smolyak polynomials (see e.g., Smolyak 1963, Malin et al. 2011, and Judd et al. 2014), whileother efficient approximation methods like adaptive sparse grid methods (Brumm and Scheidegger 2014; Brumm et al. 2015) can also be implemented.

Solving solutions for each approximation point in the space of uncertain parameters is independent each other, so it is naturally parallelizable. Similarly with Subsection 4.2, we can estimate an approximation error for the response surface function. That is, we randomly draw m' points in the space of uncertain parameters, solve their corresponding DP problem in parallel, and then use their solutions to estimate the approximation errors. For example, if $\mathbf{p}$ is the vector of uncertain parameters, $R(\mathbf{p})$ is the fitted response surface function, $\{\mathbf{p}_i : 1 \leq i \leq m'\}$ is the set of randomly drawn points, and $\{y_i : 1 \leq i \leq m'\}$ is the set of solutions from solving the corresponding DP problem of $\mathbf{p}_i$, then we can use the following formula to get an $\mathcal{L}^\infty$ relative error of the approximation of the response

surface function:

$$\max_{i=1,...,m'} \frac{|R(\mathbf{p}_i) - y_i|}{|y_i|}. \tag{6}$$

Similarly, we can get an $\mathcal{L}^1$ relative error by averaging the objective values of (6) for $i = 1, ..., m'$.

6 Application - The Climate-Economy model: DSICE

In the following we apply the methodology presented in the previous sections to an environmental economics problem which entails much uncertainty. We address the problem of optimally pricing carbon emissions under intrinsic and parametric uncertainty. Typically analyses of climate policies are conducted with the help of high-dimensional integrated assessment models. Most large-scale integrated assessment models have a complex representation of the climate system but consider economic activity as exogenous; examples are MAGICC (Wigley and Raper 1997), ICAM (Dowlatabati and Morgan 1993), FUND (Anthoff et al. 2009) and IMAGE (Batjes and Goldewijk 1994). These models neglect any endogenous economic forward-looking decision-making process, making them unable to model economic responses to future climate policies or to beliefs about the future evolution of the world. A few models are based on dynamic optimization formulations of decisions by economic agents; examples include DICE (Nordhaus 2008), MERGE (Manne and Richels 2005), and RICE (Nordhaus and Yang 1996). Nevertheless, current integrated assessment models are deterministic, assuming the decision maker knows exactly how the economy and the climate will evolve over the next centuries. Therefore, they are not capable of representing the uncertainty surrounding future climate and the economy. A few exceptions include Brock et al. (2013), Jensen and Traeger (2014), and Cai et al. (2015a).

DICE-2007 (Nordhaus 2008) is the most widely used integrated assessment model. It solves for a social optimum in the presence of tradeoffs between CO_2 abatement, consumption, and investment. In DICE-2007, the time interval of one period is 10 years. Cai et al. (2012a, 2012b) extend DICE-2007 to DICE-CJL, a continuous-time model, and solve it using various finite difference methods allowing any length for the "time period", and then find that a 10-year time step is far too large so that it produces results

significantly different than ones with short time steps, while annual time steps have acceptable accuracy.

In order to account for uncertainties, Cai et al. (2015a) develop DSICE, a stochastic integrated assessment model. DSICE builds on the annual DICE-CJL model, incorporating stochasticity.

6.1 Stochastic Production

We assume that gross world product is described by the Cobb–Douglas production function

$$f_t(K_t, L_t, \widetilde{A}_t) = \widetilde{A}_t K_t^{\alpha} L_t^{1-\alpha},$$

where $\widetilde{A}_t$ is factor productivity at time t, K_t is capital (measured in trillions of 2005 U.S. dollars), L_t is labor supply, and α is capital share. Productivity has two components: a deterministic trend A_t, and a stochastic factor ζ_t, implying that $\widetilde{A}_t \equiv \zeta_t A_t$. The deterministic trend is taken from DICE-2007 as

$$A_t = A_0 \exp\left(\Lambda(1 - e^{-0.001t})/0.001\right), \tag{7}$$

where Λ is the initial growth rate of productivity.

The stochastic productivity process is determined by two state variables: ζ_t and χ_t. If they were continuous, we would assume they follow the process

$$\log(\zeta_{t+1}) = \lambda \log(\zeta_t) + \chi_t + \varrho\omega_{\zeta,t} \tag{8}$$

$$\chi_{t+1} = r\chi_t + \varsigma\omega_{\chi,t} \tag{9}$$

where $\omega_{\zeta,t}, \omega_{\chi,t} \sim i.i.d.\,\mathcal{N}(0,1)$, and λ, ϱ, r, and ς are parameters. This is motivated by Bansal and Yaron (2004) where λ is typically set to equal one. The process in (8) and (9) is unbounded at all times, a property which makes numerical solutions very difficult. We make two adjustments. First, we discretize ζ_t and χ_t with time-dependent finite-state Markov chain (see Appendix C), with Markov transition processes denoted by $\zeta_{t+1} = g_\zeta(\zeta_t, \chi_t, \omega_{\zeta,t})$ and $\chi_{t+1} = g_\chi(\chi_t, \omega_{\chi,t})$, and driven by two serially independent stochastic

processes, $\omega_{\zeta,t}$ and $\omega_{\chi,t}$. Second, we allow λ to be a free parameter in calibration.[3] Appendix C describes our calibration of the $(\lambda, \varrho, r, \varsigma)$ parameter vector. Our calibrated value of $\lambda = 0.998$ is close to 1, implying mean reversion and a significantly narrower domain for ζ_t. The result is tractable representation of stochastic productivity that is still consistent with empirical evidence.

6.2 Epstein–Zin Preferences

We use Epstein–Zin preferences (Epstein and Zin 1989) to address the willingness of people to pay to avoid risk, because Epstein–Zin preferences are flexible specifications of decision-makers' preferences regarding uncertainty, and allow us to distinguish between risk preference and the desire for consumption smoothing. Let C_t be the stochastic consumption process, and let

$$u(C_t, L_t) = \frac{(C_t/L_t)^{1-1/\psi}}{1-1/\psi} L_t$$

where ψ is the inter-temporal elasticity of substitution. The social welfare under the Epstein–Zin preferences is defined recursively by

$$U_t = \left\{ (1-\beta)\, u(C_t, L_t) + \beta \left[\mathbb{E}_t \left\{ U_{t+1}^{1-\gamma} \right\} \right]^{\frac{1-1/\psi}{1-\gamma}} \right\}^{\frac{1}{1-1/\psi}}, \tag{10}$$

where $\mathbb{E}_t\{\cdot\}$ is the expectation conditional on the states at time t, β is the discount factor, and γ is the risk aversion parameter. Epstein–Zin preferences are special cases of Kreps–Porteus preferences (Kreps and Porteus 1978). For the special case where $\psi\gamma = 1$, we have the separable utility case used in Nordhaus (2008). In this paper, we choose $\psi = 1.5$ and $\gamma = 10$ from Bansal and Yaron (2004).[4]

6.3 DSICE

The CO_2 concentrations for the carbon cycle are modeled by a three-layer model, with

$$\mathbf{M}_t = (M_{\mathrm{AT},t}, M_{\mathrm{UO},t}, M_{\mathrm{LO},t})^\top,$$

[3] This is an extension of the long-run risk model in Cai et al. (2015a), where λ is set as one.

[4] A detailed discussion of an sensitivity analysis about the impact of ψ and γ on the optimal climate policy is given in Cai et al. (2015a).

representing carbon concentration (billions of metric tons) in the atmosphere ($M_{\mathrm{AT},t}$), upper oceans ($M_{\mathrm{UO},t}$) and lower oceans ($M_{\mathrm{LO},t}$). The CO_2 concentrations impact the surface temperature of the globe through the radiative forcing (watts per square meter):

$$\mathcal{F}_t\left(M_{\mathrm{AT}}\right) = \eta \log_2\left(M_{\mathrm{AT}}/M_{\mathrm{AT},0}\right) + F_{\mathrm{EX},t}, \tag{11}$$

where $\eta = 3.8$, and $F_{\mathrm{EX},t}$ is the exogenous radiative forcing.

The global mean temperature (measured in degrees Celsius) is represented by a two-layer model,

$$\mathbf{T}_t = \left(T_{\mathrm{AT},t}, T_{\mathrm{OC},t}\right)^\top,$$

of the atmosphere ($T_{\mathrm{AT},t}$) and oceans ($T_{\mathrm{OC},t}$).

The annual total carbon emissions (billions of metric tons) during year t is stochastic and dependent on the economic shock:

$$\mathcal{E}_t\left(K, \mu, \zeta\right) = \sigma_t(1-\mu) f_t(K, L_t, \zeta) + E_{\mathrm{Land},t}, \tag{12}$$

where $E_{\mathrm{Land},t}$ is the rate of emissions from biological processes, and σ_t is the technology factor. Therefore, the carbon cycle and temperature transition system becomes

$$\begin{aligned}
\mathbf{M}_{t+1} &= \mathbf{\Phi}_{\mathrm{M}}\mathbf{M}_t + \left(\mathcal{E}_t\left(K_t, \mu_t, \zeta_t\right), 0, 0\right)^\top, \\
\mathbf{T}_{t+1} &= \mathbf{\Phi}_{\mathrm{T}}\mathbf{T}_t + \left(\xi_1 \mathcal{F}_t\left(M_{\mathrm{AT},t}\right), 0\right)^\top,
\end{aligned}$$

where $\mathbf{\Phi}_{\mathrm{M}}$ is the carbon cycle transition matrix, and $\mathbf{\Phi}_{\mathrm{T}}$ is the climate temperature transfer matrix with the following form

$$\mathbf{\Phi}_{\mathrm{T}} = \begin{bmatrix} 1 - \xi_1\eta/\xi_2 - \xi_1\xi_3 & \xi_1\xi_3 \\ \xi_4 & 1 - \xi_4 \end{bmatrix},$$

where ξ_2 is the climate sensitivity parameter, and the values of ξ_1, ξ_3, and ξ_4 are given in Appendix A.

The output is affected by the global average surface temperature, T_{AT} . Climate factors reduce output by $1 - \Omega\left(T_{\mathrm{AT}}\right)$, where $\Omega\left(T_{\mathrm{AT}}\right)$ is a decreasing function on T_{AT}.

Abatement effort with an emission control rate μ will reduce total net CO2 emissions at some cost as a share of output. The abatement cost is proportional to μ^{θ_2}, while the adjusted cost for backstop is given by $\theta_{1,t}$. Thus, the net output is

$$\mathcal{Y}_t(K, T_{\mathrm{AT}}, \mu, \zeta) = \left(1 - \theta_{1,t}\mu^{\theta_2}\left(1 + \theta_3 e^{\theta_4(\mu-1)}\right)\right)\Omega\left(T_{\mathrm{AT}}\right) f_t(K, L_t, \zeta), \tag{13}$$

where $\theta_3 = 0.1$ and $\theta_4 = 100$ are used because much higher marginal mitigation costs will be required when μ is close to 1. Therefore, the next-stage capital is

$$K_{t+1} = (1-\delta)K_t + \mathcal{Y}_t\left(K_t, T_{\mathrm{AT},t}, \mu_t, \zeta_t\right) - C_t,$$

where δ is the annual rate of depreciation of capital.

The social planner's problem is represented as a dynamic programming problem. It has eight states including six continuous state variables (the capital stock K, the three-dimensional carbon system $\mathbf{M}$, and the two-dimensional temperature vector $\mathbf{T}$) and two discrete state variables (the stochastic productivity state ζ, and the persistence of its growth rate, χ). Let $\mathbf{x} \equiv (K, \mathbf{M}, \mathbf{T}, \zeta, \chi)$ denote the eight-dimensional state variable vector and let $\mathbf{x}^+$ denote its next period's state vector.

Let ρ be the utility discount rate. The terminal value function given in Appendix D is an approximation of sum of discounted deterministic utility over the infinite horizon. We use 300 years as the horizon for our dynamic stochastic optimization problem, as we find a larger horizon has almost no impact on the solution in the first century. We make a nonlinear change of variables[5] and express the Bellman equation in terms of utils,

[5]That is, $V_t\left(\mathbf{x}\right) = \left[U_t\left(\mathbf{x}\right)\right]^{1-\frac{1}{\psi}}/(1-\beta)$.

$(U_t)^{1-\frac{1}{\psi}}$. That is, the Bellman equation is

$$
\begin{aligned}
V_t\left(\mathbf{x}\right) = \max_{C,\mu} \quad & u(C_t, L_t) + \beta \left[\mathbb{E}_t \left\{ \left(V_{t+1}\left(\mathbf{x}^+\right)\right)^{\Gamma} \right\}\right]^{\frac{1}{\Gamma}}, \\
\text{s.t.} \quad & K^+ = (1-\delta)K + \mathcal{Y}_t(K, T_{\mathrm{AT}}, \mu, \zeta) - C_t, \\
& \mathbf{M}^+ = \mathbf{\Phi}_{\mathrm{M}}\mathbf{M} + \left(\mathcal{E}_t\left(K, \mu, \zeta\right), 0, 0\right)^{\top}, \\
& \mathbf{T}^+ = \mathbf{\Phi}_{\mathrm{T}}\mathbf{T} + \left(\xi_1 \mathcal{F}_t\left(M_{\mathrm{AT}}\right), 0\right)^{\top}, \\
& \zeta^+ = g_{\zeta}(\zeta, \chi, \omega_{\zeta}), \\
& \chi^+ = g_{\chi}(\chi, \omega_{\chi}),
\end{aligned} \tag{14}
$$

for $t = 0, 1, \ldots, 299$, where $\beta = e^{-\rho}$ is the discount factor and

$$
\Gamma = \frac{1-\gamma}{1-\frac{1}{\psi}}
$$

is the composite factor for preference. In the model, consumption C and emission control rate μ are two control variables. The definitions of parameters and exogenous paths, L_t, $\theta_{1,t}$, $F_{\mathrm{EX},t}$, $E_{\mathrm{Land},t}$, and σ_t, are given in Appendix A.

7 Parameters for Uncertainty Quantification

Many important model parameters are uncertain. Here we choose four parameters for the uncertainty quantification analysis. Those parameters are: climate sensitivity, utility discount rate, growth rate of productivity and a damage factor adjustment parameter.

Climate Sensitivity The climate sensitivity parameter ξ_2 refers to the long-run change in atmospheric temperature (in degrees Celsius) that would result from a doubling of the atmospheric stock of carbon. The mostly used climate sensitivity value in the integrated assessment models is $\xi_2 = 3$. For the uncertainty quantification analysis we study a range of climate sensitivity between 1.5 and 4.5, thus following the most recent assessment of the IPCC, expressing high confidence that 1.5 - 4.5 is a likely range (IPCC 2013). We use the central point $\xi_2 = 3$ as our benchmark level.

Damage Factor Probably the most uncertain component of any assessment of climate change are the impacts resulting from it. As an example, consider a global warming of 4 degrees Celsius (that is $T_{AT} = 4$). While studies built on the DICE model (Nordhaus 2008; Nordhaus and Sztorc 2013) assume that in this case the impacts on the economy will be about 4% of gross world product, other studies assume much higher impacts, in the order of 9% (Dietz and Stern 2015; Millner et al. 2014; Weitzman 2012) and even up to 50% (Dietz and Stern 2015) of gross world product. Here, we want to cover that large spectrum of parameter uncertainty regarding the impacts of climate change. We therefore utilize the large range of possible damage factors from the literature and specify our damage impact as a convex combination of those.

The low damage factor in the DICE-2013R model (Nordhaus and Sztorc 2013) is quadratic in the level of atmospheric temperature. We denote by $\underline{\Omega}$ the lower envelope damage factor which is defined as

$$\underline{\Omega}(T_{\mathrm{AT}}) = \frac{1}{1 + 0.00267(T_{\mathrm{AT}})^2},$$

Similarly, we use the (high) damage factor from (Dietz and Stern 2015) and denote the upper envelope damage factor by

$$\overline{\Omega}(T_{\mathrm{AT}}) = \frac{1}{1 + 0.00284(T_{\mathrm{AT}})^2 + 0.0000819(T_{\mathrm{AT}})^{6.754}}.$$

Then, our damage factor function is defined as

$$\Omega(T_{\mathrm{AT}}) = (1 - q)\underline{\Omega}(T_{\mathrm{AT}}) + q\overline{\Omega}(T_{\mathrm{AT}})$$

where $q \in [0, 1]$ is an uncertain parameter. We use the central point $q = 0.5$ as our benchmark level. Considering the previous example of global warming of 4 degrees Celsius (that is $T_{AT} = 4$) and using $q = 0.5$, our benchmark specification suggest a damage factor of 25%. Setting $q = 0$, we obtain 4%, with $q = 0.1$ we obtain 9%, and with $q = 1$ we obtain 50%. In Figure 1 we illustrate the range of damage factors considered in this study and compare it some more recent studies in the literature.

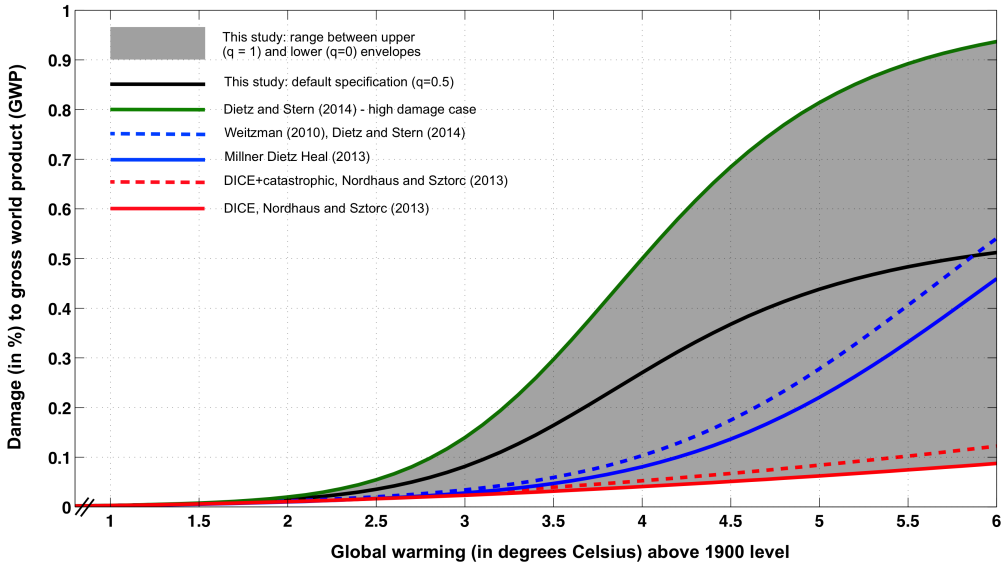

Figure 1: Damage Functions

Utility Discount Rate The utility discount rate ρ, also known as the pure rate of time preference, is an important parameter in the cost-benefit assessment of climate change as it denotes how much the welfare of future generations is valued. Moral arguments are made to justify low pure-time discounting (Dasgupta 2008; Stern 2007). Recent studies (Arrow et al. 2014; Barro 2009; Gollier 2012; Goulder and Williams 2012; Lontzek et al. 2015; Pindyck and Wang 2013) suggest that uncertainty about future damages should imply a low discount rate. However, most integrated assessment studies use higher pure-time discounting. The value for the utility discount rate in DICE is 0.015 (that it 1.5% per year), which is the upper limit of values used in IWG (2010). For the uncertainty quantification analysis we study a range of ρ between 0.001 and 0.015. We use the central point $\rho = 0.008$ as our benchmark level.

Growth Rate of Productivity In the DICE model (Nordhaus 2008), the initial time growth rate of productivity (the deterministic trend), Λ, is estimated to be 0.0092 and its standard deviation is 0.004. We choose [0.0072, 0.0112] as the range for the uncertainty quantification analysis. We use the central point $\Lambda = 0.0092$ as our benchmark level.

8 Results for Benchmark Example

In this section we apply numerical DP algorithms to solve DSICE with the benchmark parameter values: $\xi_2 = 3$, $q = 0.5$, $\rho = 0.008$, and $\Lambda = 0.0092$. We define this as our benchmark example. The numbers of nodes for discrete states ζ and χ are 33 and 19 respectively, so in additional to the 6 continuous states this model version has $33 \times 19 = 627$ discrete states.

For each discrete state, we choose a simplicial complete Chebyshev polynomial (see Appendix B) to approximate its value function. Moreover, we compute the values of the value function on the multidimensional tensor grids in the continuous state approximation domains, and then compute the Chebyshev coefficients. Since the approximation domain of K is very wide except the first periods, we use $\log(K)$ as the state variable for approximation. For all examples in this paper, the degrees of six continuous states $(\log(K), \mathbf{M}, \mathbf{T})$ for the simplicial complete Chebyshev polynomials are $(6, 6, 4, 2, 6, 4)$, and the numbers of nodes for continuous states are $(7, 7, 5, 3, 7, 5)$, so the number of coefficients per discrete state is 267 and the number of approximation nodes per discrete state is 25,725.[6]

In the maximization step of DP, we use NPSOL (Gill et al. 1994), a set of Fortran subroutines for minimizing a smooth function subject to linear and nonlinear constraints. In order to improve the efficiency of the optimization solver, we also use consumption-output ratio (i.e., $C_t/f_t(K_t, L_t, \zeta_t)$) as the control variable for consumption C_t, so NPSOL can use the "warm start" option more efficiently, because the consumption-output ratio will not change significantly over different states while C_t could. We use annual time steps over a 300-year horizon, choose a terminal value function, and use value function iteration to solve for the value function at each time t. The major computational effort is in solving the Bellman optimization problem at each approximation node at each discrete state and each time; the total number of optimization problems for 300 value function iterations is $300 \times 627 \times 25725 \approx 4.8 \times 10^{10}$.

We solve the benchmark example with the master-worker parallel DP algorithm (Cai et al. 2015d) over 628 cores of Blue Waters, a modern supercomputer, and get the

[6] We choose the degrees so that our solutions have two-digit accuracy (a higher degree approximation can get a higher accuracy, but it will also be more time-consuming).

solutions in 1.7 wall clock hours. The total number of floating-point operations per second (FLOPS) is 5.1×10^{11} while the theoretical peak performance of these compute cores is 6.3×10^{12} FLOPS,[7] so the parallel efficiency of our algorithm is more than 8% for this example. This paper uses modest-sized problems to illustrate the basic ideas. Cai et al. (2015a) apply these methods to larger problems, even some using 69,170 cores on the Blue Waters supercomputer with even greater efficiency.

Moreover, our methods have displayed nearly linear speedup in scaling tests using up to tens of thousands of cores on the Blue Waters supercomputer (Hansen et al. 2014). That is, if the problem size stays fixed, the work units per unit time are almost linear to the number of cores. This linear scaling parallel performance comes from our algorithms' efficiency with low demands for memory and communication.

Because of the efficiency of the algorithm and the computational power on Blue Waters, we are also able to do uncertainty quantification. In the next section we will solve hundreds of examples in parallel for uncertainty quantification, that is infeasible on a desktop.

After computing these Chebyshev coefficients on all discrete state values for all stages along the 300 years using the backward value function iteration method, we use a simulation method to generate the optimal paths by the forward iteration method. That is, when the state at the current stage is given, since the next stage value function approximation has been computed by the previous numerical DP algorithm, we can apply the optimization solver to get the optimal policy and the next-stage continuous state $(K, \mathbf{M}, \mathbf{T})$. Then we simulate to get the next stage stochastic state. We start this process with the given initial continuous state and $(\zeta_0, \chi_0) = (1, 0)$, and run it until the terminal time. In the following examples, we will compute 10,000 paths in parallel by the simulation method, and then plot their distribution.

8.1 Numerical results

Figure 2 shows the distribution of simulation paths of surface temperature, atmospheric carbon, and capital, and their approximation domains over the first 200 years. The domains for surface temperature and atmospheric carbon looks too wide, but they are

[7] A typical single-core 2.5 GHz processor has a theoretical performance of 10^{10} FLOPS.

necessary for approximating value functions over the stochastic productivity state with very small probabilities, as we do not want to use unstable extrapolation.

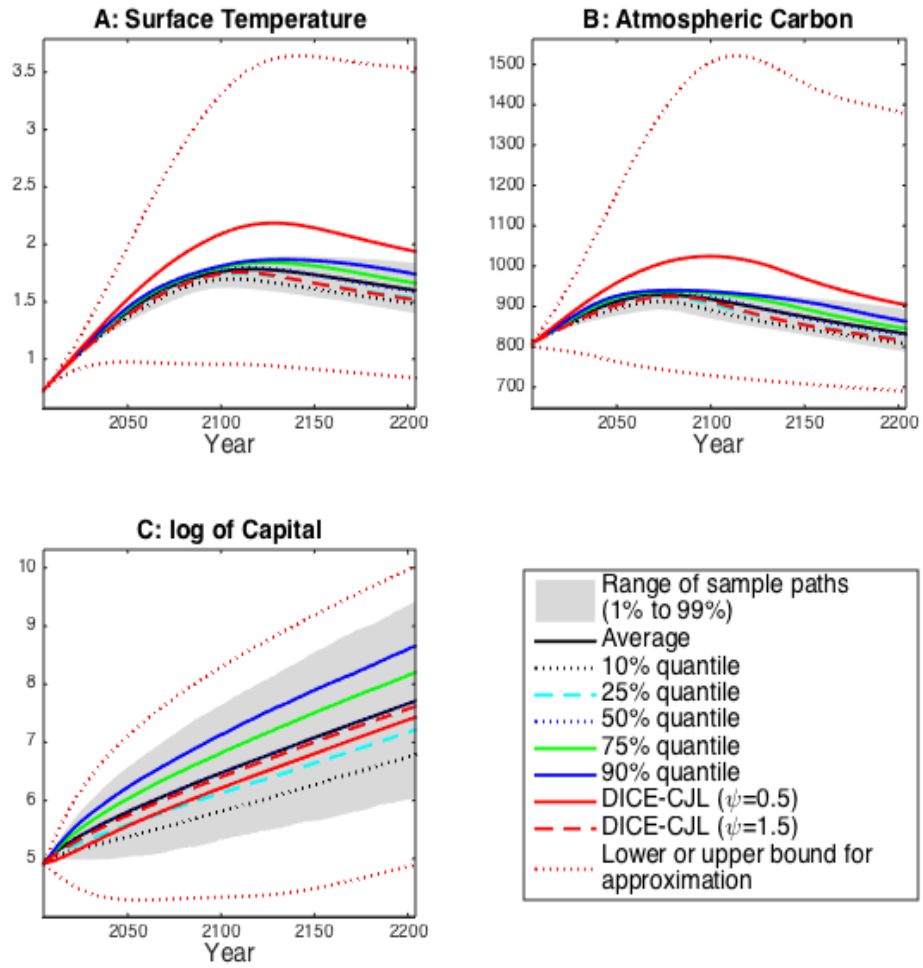

Figure 2: Simulation paths of states and their approximation domains for the benchmark example. The solid black lines are the average along time t, the solid red lines are the solutions of DICE-CJL with $\psi = 1/\gamma = 0.5$ (the utility preference used by DICE-2007), and the dashed red lines are the solutions of DICE-CJL with $\psi = 1/\gamma = 1.5$. The dotted red lines are the lower/upper bounds of the approximation domains in the corresponding state dimension at each time. The other lines are the quantiles of the 10,000 optimal paths.

Figure 3 shows the numerical results of DSICE using the distribution of simulation paths of SCC and output over the first 100 years. The initial SCC is 189 US dollars per ton, which is much larger than the DICE-CJL solution with $\psi = 0.5$ but slightly smaller than the DICE-CJL solution with $\psi = 1.5$. The most interesting observation is that the

future social cost of carbon and gross world output are both stochastic processes with variances expanding over time.

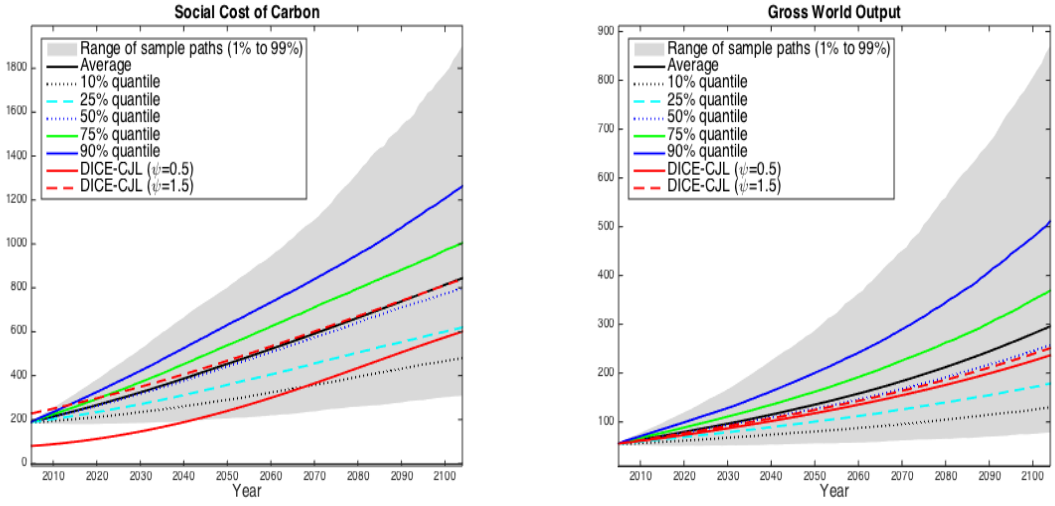

Figure 3: Social Cost of Carbon and Gross World Output of the benchmark example . The line styles are consistent with Figure 2.

8.2 Error Control

Error control is very important for any numerical algorithms. Numerical results should not be trusted if we have not examined their accuracy.

8.2.1 Test for the Deterministic Example

Our first step is to test if our numerical DP algorithm can replicate the true solution of the degenerated deterministic DSICE example, using the same parameters and algorithm details (including the approximation domains) as the stochastic problem except we kill variances . Here the DICE-CJL model helps us. It will be a natural way to compare the solutions given by the numerical DP algorithm and the solutions given by the GAMS (McCarl et al. 2014) code using a large-scale nonlinear optimizer (e.g., CONOPT (Drud 1996) and SNOPT (Gill et al. 2005)) for DICE-CJL in Cai et al. (2012b), because DICE-CJL is the deterministic degenerated case of DSICE.

Our Fortran code of the numerical DP algorithm (the deterministic version with six continuous state variables and a degenerated economic shock) is applied to solve the deterministic DSICE example. Our target is not only to replicate the solution from

DICE-CJL, but also to obtain some accuracy measure for the corresponding stochastic problem using the same algorithm. Thus, we use the approximation domains and methods that are the same ones used in solving the stochastic problems on purpose, although the deterministic example can be solved with much narrower domains and then higher accuracy. Note that it will be more efficient to run this accuracy test before we solve the stochastic problem, because it is much faster to solve the deterministic problem (we can get the approximation domains of the stochastic problem at first) and then we can test out the best choice of approximation methods and their specification such as degrees of simplicial complete Chebyshev polynomials.

After computing the Chebyshev coefficients for all stages using the backward value function iteration method, we generate the optimal path with the given initial state by the forward iteration method. That is, given the current stage's state, since we have the approximation of the next-stage value function, we can use the Bellman equation to compute the optimal consumption and emission control so that we can get the optimal next-stage state, and then go on until the terminal stage.

Now we use the solution of DICE-CJL from the GAMS code in Cai et al. (2012b) to verify the accuracy of the optimal path computed from the numerical DP algorithm. It is run on a desktop computer with 3.5 GHz 6-Core Intel Xeon E5 and the running time is less than two minutes.

Figure 4 plots the relative errors of the optimal paths of states, $(K_t, M_{\mathrm{AT},t}, T_{\mathrm{AT},t})$, and control variables, (C_t, μ_t), over the whole horizon. The relative errors of the other states, $(M_{\mathrm{UO},t}, M_{\mathrm{LO},t}, T_{\mathrm{LO},t})$, are even smaller so they are omitted. The errors are computed in the following formula:

$$\left| \frac{X^*_{t,\mathrm{DP}} - X^*_{t,\mathrm{GAMS}}}{X^*_{t,\mathrm{GAMS}}} \right|,$$

where $X^*_{t,\mathrm{DP}}$ is the optimal path at year t from our numerical DP algorithm, and $X^*_{t,\mathrm{GAMS}}$ is the optimal solution at year t of the GAMS code for the DICE-CJL model. We see that the relative errors are small, particularly those at the first half of this century are around or less than 0.1%.

To verify the accuracy of our method, we also use Figure 5 to show the relative approximation errors in $\mathcal{L}^1$ norm (the left panel) and $\mathcal{L}^\infty$ norm (the right panel) of two

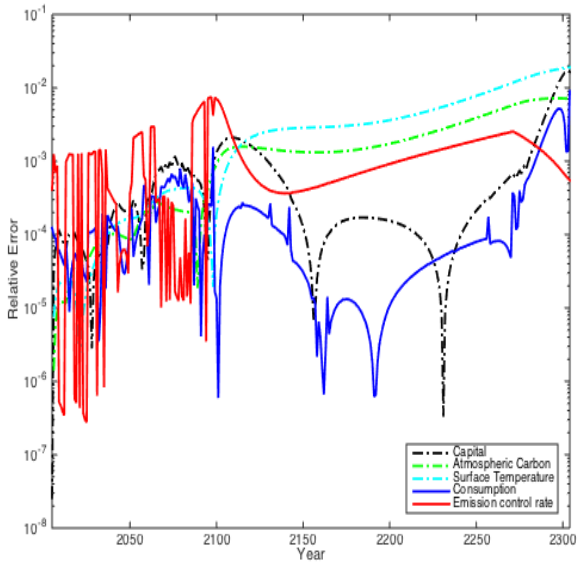

Figure 4: Relative errors for the deterministic example. The red and blue solid lines in the figure are respectively the errors of two policy functions—emission control rates and consumptions.

policy functions (consumption and emission control rate) and value function at each year for the deterministic case over the first 100 years. The errors are computed with the accuracy checking method introduced in Section 4.2. We see that all the stepwise errors are small, particularly for the value function approximation or the policy functions in the first 30 years. The $\mathcal{L}^1$ errors for emission control rates start to increase quickly after 30 years and then look "big". However, these "big" $\mathcal{L}^1$ errors are caused by the polynomial approximation to a function with kinks, where the emission control rate is near to or hits its upper limit. We can also see this from comparing with the corresponding $\mathcal{L}^\infty$ errors for emission control rates, where they do not increase as quickly as the $\mathcal{L}^1$ errors.

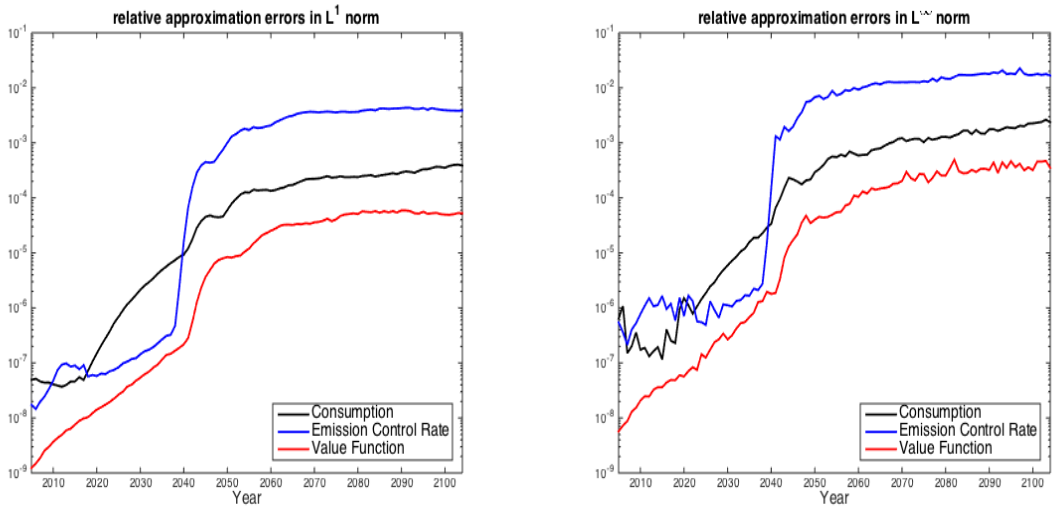

Figure 5: Stepwise errors of numerical DP for the deterministic example

8.2.2 Stepwise errors for the stochastic benchmark example

Figure 6 shows the relative approximation errors in $\mathcal{L}^1$ norm (the left panel) and $\mathcal{L}^\infty$ norm (the right panel) of two policy functions and value function at each year for the benchmark case over the first 100 years. By comparing Figure 6 and Figure 5, we see that the stochastic example has about one-order magnitude higher $\mathcal{L}^1$ errors than the deterministic example using the same approximation domains and methods.[8] Thus, we estimate that errors of the policy functions of the stochastic example are about one-order magnitude higher than the ones of the deterministic example. That is, since Figure 4 tells us that the errors of the policy functions are about 0.1% for the deterministic example, we estimate that the errors of the policy functions of the stochastic example are about 1%, an acceptable criterion in economics.

[8] Although the $\mathcal{L}^\infty$ errors of the stochastic example are about three-order magnitude higher than the $\mathcal{L}^\infty$ errors of the deterministic example in the first few decades, they are in the those extreme states that have almost zero probabilities to be reached.

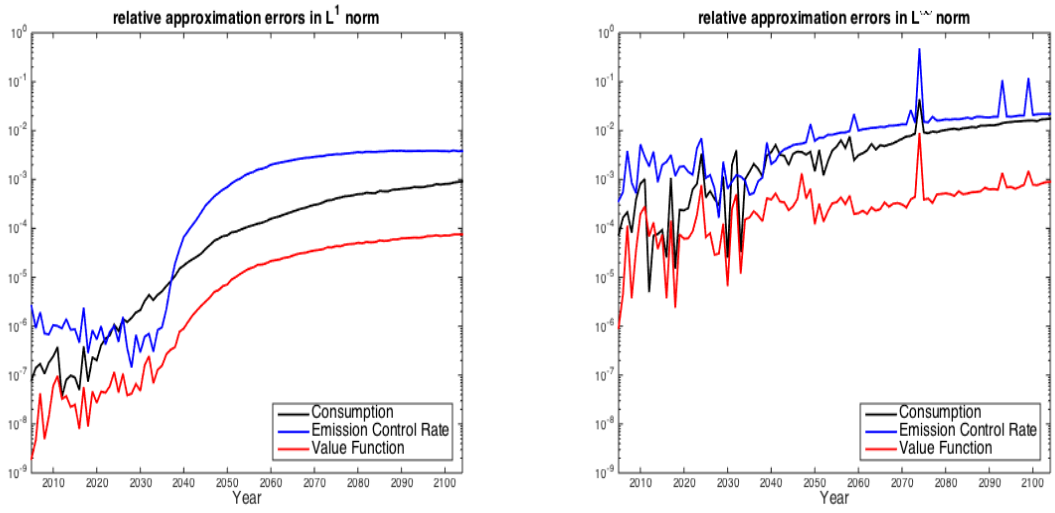

Figure 6: Stepwise errors of numerical DP for the benchmark example

8.2.3 Sensitivity to Terminal Value Functions

The solutions of value function iteration relies on the choice of the terminal value function. Appendix D gives our choice for the terminal value function V_{300}. We find that when we have 10% relative change in the whole terminal value function, it has 0.3% relative change on the initial SCC. In fact, the impact of terminal value function mainly relies on the horizon and discount rate. If we let the discount rate be $\rho = 0.015$, then the relative impact on the initial SCC is 0.07% , but if $\rho = 0.001$ then it increases to 2%. When we change the horizon from 300 years to 600 years, then its impact on the initial SCC is almost zero.

9 Results: Uncertainty Quantification & Surface Response Function

We are limited by the current state of knowledge about the critical parameters about the economics and climate system. Also, different decision-makers usually have different beliefs about the implications of parameter vectors which are represented by the four-dimensional $(\xi_2, \rho, q, \Lambda)$ here (we assume that the other two preference parameters, ψ and γ, are given for simplicity). This is partially solved by Table 2 including the results of 16 cases. Since we are only interested in the initial-year SCC here, we use 19 nodes

for ζ_t instead of 33 nodes in the benchmark case to save computational resource, as we find that they give almost the same initial-year SCC.

ξ_2	q	SCC			
		$\rho = 0.001$		$\rho = 0.015$	
		$\Lambda = 0.0072$	$\Lambda = 0.0112$	$\Lambda = 0.0072$	$\Lambda = 0.0112$
1.5	0	113	172	21	24
	1	129	191	24	30
4.5	0	604	827	101	120
	1	832	1141	253	281

Table 2: Social cost of carbon in various cases for sensitivity analysis

From Table 2, we know that SCC is sensitive to the values of the four uncertain parameters and has a wide range—from \$21 to \$1,141. Due to this wide range, we want to do a more complete examination of parameter sensitivity.

For a specific point in the space of the four uncertain parameters, if it is not quite close to one of points used in the sensitivity analysis, then its optimal carbon tax cannot be estimated well by simply using a multi-linear interpolation over the values in the table. For example, if we use the multi-linear interpolation to estimate the carbon tax for the benchmark case, then its estimation is \$304, about 60% higher than its true value, \$189.

Since the discretization on the parameter space with 16 cases is too coarse, a more accurate map from the parameter space to the policy space is desired. If we compute SCC over a tensor grid to get a good multi-linear interpolation, it will be too time-consuming as each case takes around 1.3 wall clock hours using 362 cores if we run the master-work parallel method to solve it. Thus, we construct a response surface function for uncertainty quantification over the four uncertain parameters using sparse Smolyak grid.

In this paper, we compute initial SCC at the level-3 Smolyak sparse grid (Smolyak 1963 and Malin et al. 2011) over the four-dimensional space $(\xi_2, q, \log(\rho), \Lambda)$ of four parameters: $\xi_2 \in [1.5, 4.5]$, $q \in [0, 1]$, $\log(\rho) \in [\log(0.001), \log(0.015)]$ and $\Lambda \in [0.0072, 0.0112]$. We then used a degree-8 Chebyshev-Smolyak polynomial (Smolyak 1963 and Malin et al. 2011) to fit the log of SCC over $(\xi_2, q, \log(\rho), \Lambda)$, giving us an approximation to the dependence of initial SCC to parameters over the four-dimensional hypercube. That is,

our response surface function has the following form:

$$\text{SCC} = \exp\left(P\left(\xi_2, q, \log(\rho), \Lambda\right)\right)$$

where P is the fitted degree-8 Chebyshev-Smolyak polynomial using the solution over the Smolyak sparse grid which has 137 different points. We solve these 137 cases in parallel on the supercomputer—Blue Waters.

We also do a verification test for the response surface function. For example, the approximated value from the response surface function over the benchmark case is \$184, about 2.7% errors with the true value. We have a global measurement for the approximation error. At first we randomly draw 100 points in the parameter space of $(\xi_2, q, \log(\rho), \Lambda)$, and solve the optimal SCC for each point (these 100 jobs are naturally parallelized). We then compare them with the approximated SCC on these out-of-sample points from the response surface function. We find that the $\mathcal{L}^\infty$ relative error using the formula (6) is 3.8% and the corresponding $\mathcal{L}^1$ relative error is 0.76%. Therefore, our degree-8 polynomial is good for the purposes of the discussion below.

Figure 7 displays the contours of the response surface functions with labelled SCC ranging from \$40 to \$550. We see that the bottom-right contour ($q = 1$ and $\Lambda = 0.0112$) has the largest SCC relative to other contours, and the largest SCC is more than \$550 for the cases with $\xi_2 > 4$ and $\rho < 0.004$, and the marginal SCC over ξ_2 or ρ is also large: the exact range of the bottom-right contour is from \$30 at its left-top corner point with $\xi_2 = 1.5$ and $\rho = 0.015$, to \$1,141 at its right-bottom corner point with $\xi_2 = 4.5$ and $\rho = 0.001$, which are also listed in Table 2. The top-left contour ($q = 0$ and $\Lambda = 0.0072$) has the smallest SCC relative to other contours; but the marginal SCC over ξ_2 or ρ is still large: its exact range is from \$21 to \$604 listed in Table 2 . From the figure, we see clearly that SCC increases as climate sensitivity ξ_2, damage factor q, or growth trend parameter Λ increases, or utility discount rate ρ decreases.

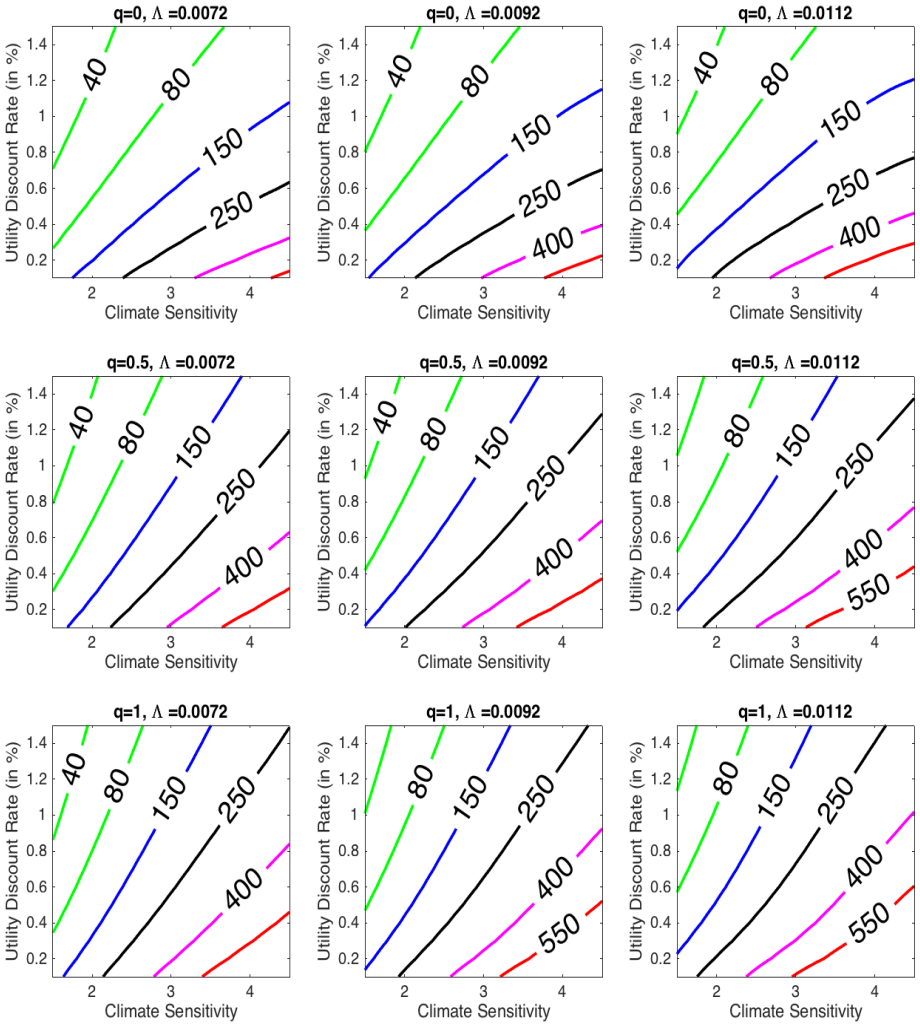

Figure 7: Contours of Response Surface Function. Here we show contours for $q \in \{0, 0.5, 1\}$ and $\Lambda \in \{0.0072, 0.0092, 0.0112\}$, each contour graph representing one combination of (q, Λ). For example, the bottom-right contour represents the case with $q = 1$ and $\Lambda = 0.0112$. The horizontal axis of each contour is the climate sensitivity ξ_2. The vertical axis of each contour is the utility discount rate ρ in percentage. In each contour of Figure 7, the labelled numbers on the lines represent their SCC on the points of (ξ_2, ρ).

10 Conclusion and Extensions

With the efficient parallel algorithms, we have done an extensive exploration of the parameter space in DSICE to determine the sensitivity of conclusions for parameters about which we have limited information. The accuracy tests indicate that the algorithms

are reliable as well as fast. Our results have shown decisively that climate change issues can be examined with the same complexity used in standard dynamic stochastic models in economics.

While we developed the DSICE framework incorporating both economic and climate risks in Cai et al. (2015a), we also implemented our methods discussed in this paper to DSICE for solving optimal climate policies with both economic and climate risks under various parameter uncertainties of utility preferences and climate risk specifications in Cai et al. (2015a). Moreover, DSICE has been applied in the literature. For example, Lontzek et al. (2015) use our methods and DSICE to examine the impact on the carbon tax under various continuous climate tipping points. Moreover, Cai et al. (2015b) apply our methods and DSICE to do a cost-benefit assessment of climate policies under environmental tipping points impacting on market and non-market goods and services.

Moreover, our DP methods can be applied to basically any dynamic programming problem. Most of macroeconomics looks at models where none of the primitives depend on calendar time. We can also solve non-stationary problems such as ones with unit roots (e.g., Cai et al. 2015a).

Acknowledgments

Cai acknowledges support from the Hoover Institution and the National Science Foundation grant SES-0951576. This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications. We also thank the HTCondor team of the University of Wisconsin-Madison for their support.

References

[1] Anderson, E., W. Brock, L. Hansen, and A. Sanstad (2014). Robust analytical and computational explorations of coupled economic-climate models with carbon-climate response. RDCEP working paper 13-05.

[2] Anthoff, D., R.S.J. Tol, and G.W. Yohe (2009). Discounting for Climate Change. *Economics: The Open-Access, Open-Assessment E-Journal*, 3(24).

[3] Arrow, K., M. Cropper, C. Gollier, B. Groom, G. Heal, R. Newell, W. Nordhaus, R. Pindyck, W. Pizer, P. Portney, T. Sterner, R.S.J. Tol, and M. Weitzman (2013). Determining benefits and costs for future generations. *Science* 341 (6144), 349–350.

[4] Bahn, O., Haurie, A., and R. Malhamé (2008). A Stochastic Control Model for Optimal Timing of Climate Policies. *Automatica*, 44(6):1545-1558.

[5] Barro, R.J. (2009). Rare disasters, asset prices, and welfare costs. *American Economic Review* 99, 243–264.

[6] Batjes, J.J., and C.G.M. Goldewijk (1994). The IMAGE 2 hundred year (1890–1990) database of the global environment (HYDE). RIVM, Bilthoven, 410100082

[7] Bellman, R. (1957). *Dynamic Programming*. Princeton University Press.

[8] Bertsekas, D. (2012). *Dynamic Programming and Optimal Control, Vol. II, 4th Edition: Approximate Dynamic Programming*. Athena Scientific.

[9] Brock, W.A., S.N. Durlauf, J.M. Nason, and G. Rondina (2007a). Simple versus optimal rules as guides to policy. *Journal of Monetary Economics* 54: 1372–1396.

[10] Brock, W.A., S.N. Durlauf, and K.D. West (2007b). Model uncertainty and policy evaluation: Some theory and empirics. *Journal of Econometrics* 136: 629–664.

[11] Brock, W.A., G. Engstrom, D. Grass, and A. Xepapadeas (2013). Energy Balance Climate Models and General Equilibrium Optimal Mitigation Policies. *Journal of Economic Dynamics and Control*, 37(12): 2371–2396.

[12] Brumm, J., D. Mikushin, Scheidegger, S., and O. Schenk (2015). Scalable High-Dimensional Dynamic Stochastic Economic Modeling. *Journal of Computational Science*, 11, 12–25.

[13] Brumm, J., and S. Scheidegger (2014). Using adaptive sparse grids to solve high-dimensional dynamic models. SSRN working paper 2349281.

[14] Cai, Y. (2010). *Dynamic Programming and Its Application in Economics and Finance*. PhD thesis, Stanford University.

[15] Cai, Y., K.L. Judd, and T.S. Lontzek (2012a). Open science is necessary. *Nature Climate Change*, Vol. 2, Issue 5, 299–299.

[16] Cai, Y., K.L. Judd, and T.S. Lontzek (2012b). Continuous-Time Methods for Integrated Assessment Models. NBER working paper 18365.

[17] Cai, Y., and K.L. Judd (2013). Shape-preserving dynamic programming. *Mathematical Methods of Operations Research*, 77(3): 407–421.

[18] Cai, Y., K.L. Judd, and R. Xu (2013). Numerical solution of dynamic portfolio optimization with transaction costs. NBER working paper No. 18709.

[19] Cai, Y., and K.L. Judd (2014). Advances in numerical dynamic programming and new applications. Chapter 8 in: *Handbook of Computational Economics*, Vol. 3, ed. by Karl Schmedders and Kenneth L. Judd, Elsevier.

[20] Cai, Y., and K.L. Judd (2015). Dynamic programming with Hermite approximation. *Mathematical Methods of Operations Research*, 81, 245–267.

[21] Cai, Y., K.L. Judd, and T.S. Lontzek (2015a). The social cost of carbon with economic and climate risks. Working paper, arXiv preprint arXiv:1504.06909.

[22] Cai, Y., K.L. Judd, T.M. Lenton, T.S. Lontzek, and D. Narita (2015b). Environmental tipping points significantly affect the cost-benefit assessment of climate policies. *Proceedings of the National Academy of Sciences*, 112(15), 4606–4611.

[23] Cai, Y., K.L. Judd, T.S. Lontzek, V. Michelangeli, and C.-L. Su (2015c). Nonlinear programming method for dynamic programming. Forthcoming in *Macroeconomic Dynamics*.

[24] Cai, Y., K.L. Judd, G. Thain, and S. Wright (2015d). Solving dynamic programming problems on a computational grid. *Computational Economics*, Vol. 45, No. 2, 261–284.

[25] Cai, Y., and A.H. Sanstad (2015). Model uncertainty and energy technology policy: the example of induced technical change. Forthcoming in *Computers & Operations Research*.

[26] Chen, V.C.P., D. Ruppert, and C.A. Shoemaker (1999). Applying Experimental Design and Regression Splines to High-Dimensional Continuous-State Stochastic Dynamic Programming. *Operations Research* 47(1):38–53.

[27] Cocco, J.F., F.J. Gomes, and P.J. Maenhout (2005). Consumption and portfolio choice over the life cycle. *The Review of Financical Studies,* 18(2): 491–533.

[28] Dantzig, G.B., R.P. Harvey, Z.F. Landowne, and R.D. McKnight (1974). *DYGAM—A Computer System for the Solution of Dynamic Programs*. Control Analysis Corporation, Palo Alto, California.

[29] Dasgupta, P. (2008). Discounting Climate Change. *Journal of Risk and Uncertainty* 37: 141–169.

[30] De Farias, D.P., and B. Van Roy (2003). The linear programming approach to approximate dynamic programming. *Operations Research*, 51(6), 850–865.

[31] Den Haan, W.J., K.L. Judd and M. Juillard (2011). Computational suite of models with heterogeneous agents II: Multi-country real business cycle models. *Journal of Economic Dynamics & Control*, 35: 175–177.

[32] Dietz, S., A. Millner, and G. Heal (2013). Scientific ambiguity and climate policy. *Environmental and Resource Economics*, 55(1), 21–46.

[33] Dietz, S., and N. Stern (2015). Endogenous growth, convexity of damages and climate risk: how Nordhaus' framework supports deep cuts in carbon emissions. *The Economic Journal* 125, 574–620.

[34] Dowlatabati, H., and M.G. Morgan (1993). A Model Framework for Integrated Studies of Climate Change. *Science* 259, 1813–1814.

[35] Drud, A.S. (1996). CONOPT: a system for large scale nonlinear optimization. ARKI Consulting and Development A/S, Bagsvaerd.

[36] Environmental Protection Agency (2010). Peer Review of ADAGE and IGEM.

[37] Epstein, L.G., and S.E. Zin (1989). Substitution, risk aversion, and the temporal behavior of consumption and asset returns: a theoretical framework. Econometrica, 57(4), 937–969.

[38] Fisher, A., and U. Narain (2003). Global Warming, Endogenous Risk, and Irreversibility. *Environmental and Resource Economics*, 25(4), 395–416.

[39] Friedman, J. H. (1991). Multivariate adaptive regression splines (with discussion). Ann. Statist. 19, 1–141.

[40] Gill, P., W. Murray, M.A. Saunders, and M.H. Wright (1994). User's Guide for NPSOL 5.0: a Fortran Package for Nonlinear Programming. Technical report, SOL, Stanford University.

[41] Gill, P., W. Murray, and M.A. Saunders (2005). SNOPT: an SQP algorithm for largescale constrained optimization. *SIAM Rev* 47(1): 99–131.

[42] Gollier, C. (2012). *Pricing the Planet's Future: The Economics of Discounting in an Uncertain World*. Princeton University Press.

[43] Goettle, R.J., M.S. Ho, D.W. Jorgenson, D.T. Slesnick, and P.J. Wilcoxen (2009). Analyzing Environmental Policies with IGEM, an Intertemporal General Equilibrium Model of U.S. Growth and the Environment. Cambridge, MA: Dale Jorgenson Associates.

[44] Griebel, M., H. Wozniakowski (2006). On the optimal convergence rate of universal and nonuniversal algorithms for multivariate integration and approximation. *Math Comput* 75(255):1259–1286.

[45] Gupta A., and W. Murray (2005). A framework algorithm to compute optimal asset allocation for retirement with behavioral utilities. *Computational Optimization and Applications*, 32(1/2): 91–113.

[46] Hansen L.P., K.L. Judd, Y. Cai, and S. Scheidegger (2014). Policy responses to climate changes in a dynamic stochastic economy. In: N. Gaynor, W. Kramer, C. Beldica (Eds.), Blue Waters 2014 Annual Report.

[47] Hansen, L.P., and T.J. Sargent (2005). Robust Estimation and Control under Commitment. *Journal of Economic Theory* 124 (2): 258–301.

[48] Hansen, L.P., and T.J. Sargent (2007a). Recursive Robust Estimation and Control without Commitment. *Journal of Economic Theory* 136: 1–27.

[49] Hansen, L.P., and T.J. Sargent (2007b). *Robustness*. Princeton University Press.

[50] Infanger, G. (2006). Dynamic asset allocation using a stochastic dynamic programming approach. In *Handbook of Asset and Liability Management*, volume 1, North Holland.

[51] IPCC (2013). Long-term Climate Change: Projections, Commitments and Irreversibility. *Climate Change 2013: The Physical Science Basis - IPCC Working Group I Contribution to AR5*. Geneva, Switzerland: Intergovernmental Panel on Climate Change. Retrieved 2015-02-02.

[52] IWG (2010). *Social Cost of Carbon for Regulatory Impact Analysis under Executive Order 12866*. United States Government. http://www.whitehouse.gov/sites/default/files/omb/inforeg/for-agencies/Social-Cost-of-Carbon-for-RIA.pdf

[53] Jensen, S., and C. Traeger (2014). Optimal Climate Change Mitigation under Long-Term Growth Uncertainty: Stochastic Integrated Assessment and Analytic Findings. *European Economic Review* 69: 104–125.

[54] Johnson, S. A., J. R. Stedinger, C. A. Shoemaker, Y. Li, J. A. Tejada-Guibert (1993). Numerical solution of continuous state dynamic programs using linear and spline interpolation. *Operations Research* 41(3):484–500.

[55] Judd, K.L. (1998). *Numerical Methods in Economics*. The MIT Press.

[56] Judd, K.L., L. Marlia, S. Marliar, and R. Valero (2014). Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics & Control*, 44, 92–123.

[57] Kelly, D.L., and C.D. Kolstad (1999). Bayesian learning, growth, and pollution. *Journal of Economic Dynamics and Control* 23, 491–518.

[58] Kall, P., and S.W. Wallace (1994). *Stochastic Programming*. John Wiley & Sons, Chichester.

[59] Lontzek, T.S., Y. Cai, K.L. Judd, and T.M. Lenton (2015). Stochastic integrated assessment of climate tipping points indicates the need for strict climate policy. *Nature Climate Change* 5, 441–444.

[60] Lucas, R.E., E.C. Prescott, and N.L. Stokey (1989). Recursive Methods in Economic Dynamics. Cambridge, MA: Harvard University Press.

[61] Malin, B.A., D. Krueger, and F. Kubler (2011). Solving the multi-country real business cycle model using a smolyak collocation method. *Journal of Economic Dynamics and Control* 35, 229–239.

[62] Manne, A., and R. Richels (2005). MERGE: An Integrated Assessment Model for Global Climate Change. *Energy and Environment* (175-189) edited by. Loulou, R., Waaub, J-P. and Zaccour, G.

[63] McCarl, B., et al. (2014). McCarl GAMS User Guide. GAMS Development Corporation.

[64] Nordhaus, W. (2008). *A Question of Balance: Weighing the Options on Global Warming Policies*. Yale University Press.

[65] Nordhaus, W. D., and Sztorc, P. (2013). DICE 2013 - Introduction and User's Manual. Available at http://www.econ.yale.edu/~nordhaus/homepage/documents/Dicemanualfull.pdf.

[66] Oberkampf, W.L., and C.J. Roy (2010). *Verification and Validation in Scientific Computing*. Cambridge University Press.

[67] Pindyck, R.S. (2013). Climate Change Policy: What Do the Models Tell Us? *Journal of Economic Literature*, 51(3): 860–872.

[68] Pindyck, R.S., and N. Wang (2013). The economic and policy consequences of catastrophes. *American Economic Journal: Economic Policy*, 5, 306–339.

[69] Powell, W.B. (2011). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. John Wiley & Sons.

[70] Powell, W.B., and B. Van Roy (2004). Approximate dynamic programming for high-dimensional dynamic resource allocation problems. In *Handbook of Learning and Approximate Dynamic Programming*, Wiley-IEEE Press, Hoboken, NJ.

[71] Puterman, M.L. (2005). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics.

[72] Rust, J. (1997). Using randomization to break the curse of dimensionality. *Econometrica*, 65(3): 487–516.

[73] Rust, J., J.F. Traub, and H. Wozniakowski (2002). Is there a curse of dimensionality for contraction fixed points in the worst case? *Econometrica*, 70(1): 285–329.

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

[75] Stern, N. (2007). *The Economics of Climate Change: The Stern Review*. Cambridge University Press.

[76] Tauchen, G. (1986). Finite state Markov-chain approximations to univariate and vector autoregressions. *Economic Letters*, 20, 177–181.

[77] Trick, M.A., and S.E. Zin (1997). Spline approximations to value functions — linear programming approach. *Macroeconomic Dynamics*, 1, 255–277.

[78] Webster, M., N. Santen, and P. Parpas (2012). An Approximate Dynamic Programming Framework for Modeling Global Climate Policy under Decision-Dependent Uncertainty. *Computational Management Science*.

[79] Weitzman, M. (2012). GHG targets as insurance against catastrophic climate damages. *Journal of Public Economic Theory* 14 (2), 221–244.

[80] Wigley, T.M.L., and S.C.B. Raper (1997). Model for the Assessment of Greenhouse-gas Induced Climate Change (MAGICC Version 2.3.), The Climate Research Unit, University of East Anglia, UK.

Appendix A—Definition of Trends and Parameters

DSICE (Cai et al. 2015b) has the following exogenous paths: L_t (population), σ_t (carbon intensity of output.), $\theta_{1,t}$ (mitigation cost parameter), $E_{\text{Land},t}$ (annual carbon emissions from biological processes), and $F_{\text{EX},t}$ (exogenous radiative forcing). We give their definitions here for readers' convenience:

$$L_t = 6514e^{-0.035t} + 8600(1 - e^{-0.035t}) \tag{15}$$

$$\sigma_t = \sigma_0 \exp\left(-0.0073(1 - e^{-0.003t})/0.003\right) \tag{16}$$

$$\theta_{1,t} = \frac{1.17\sigma_t\left(1 + e^{-0.005t}\right)}{2\theta_2} \tag{17}$$

$$E_{\text{Land},t} = 1.1e^{-0.01t} \tag{18}$$

$$F_{\text{EX},t} = \begin{cases} -0.06 + 0.0036t, & \text{if } t \leq 100 \\ 0.3, & \text{otherwise} \end{cases} \tag{19}$$

We also list the values of other parameters and/or definition of all parameters, variables, and symbols, in Tables 3–5.

$t \in \{0, 1, ..., 300\}$	time in years (t represents year $t + 2005$)
$\psi = 1.5$	inter-temporal elasticity of substitution
$\gamma = 10$	risk aversion parameter
Γ	the composite factor for preference
$\rho \in [0.001, 0.015]$	utility discount rate (default: 0.008)
$\beta = e^{-\rho}$	utility discount factor
A_t	productivity trend at time t, $A_0 = 0.0272$
L_t	population at time t
K_t	capital at time t (in \$ trillions), $K_0 = 137$
C_t	consumption at time t
$\alpha = 0.3$	capital share in output
$\Lambda \in [0.0072, 0.0112]$	initial growth rate of the productivity trend (default: 0.0092)
$\delta = 0.1$	annual depreciation rate
$\mathcal{Y}_t$	gross world output at time t
SCC_t	social cost of carbon in dollars per ton

Table 3: Parameters, variables, and symbols for the economic system of our model

χ_t	persistence of productivity shock at time t, $\chi_0 = 0$
ζ_t	stochastic productivity shock at time t, $\zeta_0 = 1$
$(\lambda, \varrho, r, \varsigma) = (0.998, 0.035, 0.65, 0.007)$	productivity process parameters
$\widetilde{A}_t = \zeta_t A_t$	stochastic productivity at time t
$\omega_{\zeta,t}$	i.i.d. shocks in transition of ζ_t
$\omega_{\chi,t}$	i.i.d. shocks in transition of χ_t

Table 4: Parameters, variables, and symbols for the stochastic growth specification

Carbon Cycle	$M_{\mathrm{AT},t}$	carbon concentration (billion tons) in atmosphere at time t, $M_{\mathrm{AT},0} = 808.9$
	$M_{\mathrm{UO},t}$	carbon concentration in upper ocean (billion tons) at time t, $M_{\mathrm{UO},0} = 1255$
	$M_{\mathrm{LO},t}$	carbon concentration in lower ocean (billions tons) at time t, $M_{\mathrm{LO},0} = 18365$
	$\mathbf{M}_t = (M_{\mathrm{AT},t}, M_{\mathrm{UO},t}, M_{\mathrm{LO},t})^\top$	carbon concentration vector at time t
	$\Phi_{\mathrm{M}} = \begin{bmatrix} 0.981 & 0.01 & 0 \\ 0.019 & 0.9846 & 0.00034 \\ 0 & 0.0054 & 0.99966 \end{bmatrix}$	transition matrix of carbon cycle
	$M^*_{\mathrm{AT}} = 596.4$	preindustrial atmospheric carbon concentration
	$E_{\mathrm{Land},t}$	carbon emissions (billions tons) from land use in year t
Temperature System	$T_{\mathrm{AT},t}$	global average surface temperature (degrees Celsius) at time t, $T_{\mathrm{AT},0} = 0.7307$
	$T_{\mathrm{OC},t}$	global average ocean temperature (degrees Celsius) at time t, $T_{\mathrm{OC},0} = 0.0068$
	$\mathbf{T}_t = (T_{\mathrm{AT},t}, T_{\mathrm{OC},t})^\top$	temperature vector at time t
	Φ_{T}	transition matrix of temperature system
	$(\xi_1, \xi_3, \xi_4) = (0.037, 0.277, 0.0048)$	temperature transition parameters
	$\xi_2 \in [1.5, 4.5]$	climate sensitivity (default: 3)
	$\mathcal{F}_t$	radiative forcing at time t
	$F_{\mathrm{EX},t}$	exogenous radiative forcing in year t
	$\eta = 3.8$	radiative forcing parameter
Interaction between the economic system and the climate system	$\mathbf{S} = (K, \mathbf{M}, \mathbf{T}, \zeta, \chi)$	state vector
	μ_t	emission control rate at time t
	$q \in [0, 1]$	damage factor parameter (default: 0.5)
	$(\theta_2, \theta_3, \theta_4) = (2.8, 0.1, 100)$	mitigation cost parameters
	σ_t	technology factor at time t, $\sigma_0 = 0.13418$
	$\theta_{1,t}$	adjusted cost for backstop at time t

Table 5: Parameters, variables, and symbols in the carbon and temperature systems and their interaction with the economic system

Appendix B—Approximation

An approximation scheme approximates the value function with $\hat{V}(\mathbf{x};\mathbf{b}) = \sum_{j=0}^{n} b_j \phi_j(\mathbf{x})$ for some vector of parameters $\mathbf{b}$. A spectral method uses globally nonzero basis functions $\phi_j(\mathbf{x})$. Examples of spectral methods include ordinary or Chebyshev polynomial approximation. In contrast, a finite element method uses local basis functions where for each j the basis function $\phi_j(\mathbf{x})$ is zero except on a small part of the approximation domain. Examples of finite element methods include piecewise linear interpolation, cubic splines, and B-splines. This paper uses spectral methods; specifically, we use Chebyshev polynomials after transforming the domain via a nonlinear change of variables. See Cai (2010), Cai and Judd (2014, 2015), and Judd (1998) for more details.

Chebyshev Polynomial Approximation

Chebyshev polynomials on $[-1,1]$ are defined as $\phi_j(z) = \cos(j\cos^{-1}(z))$. The Chebyshev polynomials on a general interval $[x_{\min}, x_{\max}]$ are defined as $\phi_j((2x - x_{\min} - x_{\max})/(x_{\max} - x_{\min}))$ for $j \geq 0$, and are orthogonal under the weighted inner product $\langle f, g\rangle = \int_{x_{\min}}^{x_{\max}} f(x)g(x)w(x)dx$ with the weighting function

$$w(x) = \left(1 - \left(\frac{2x - x_{\min} - x_{\max}}{x_{\max} - x_{\min}}\right)^2\right)^{-1/2}.$$

A degree n Chebyshev polynomial approximation for $V(x)$ on $[x_{\min}, x_{\max}]$ is

$$\hat{V}(x;\mathbf{b}) = \sum_{j=0}^{n} b_j \phi_j\left(\frac{2x - x_{\min} - x_{\max}}{x_{\max} - x_{\min}}\right), \tag{20}$$

where b_j are the Chebyshev coefficients.

The canonical Chebyshev nodes on $[-1,1]$ are $z_i = -\cos\left((2i-1)\pi/(2m)\right)$ for $i = 1,\ldots,m$, and the corresponding Chebyshev nodes adapted for the general interval $[x_{\min}, x_{\max}]$ are $x_i = (z_i + 1)(x_{\max} - x_{\min})/2 + x_{\min}$. If we have Lagrange data $\{(x_i, v_i) : i = 1,\ldots,m\}$ with $v_i = V(x_i)$, then the coefficients b_j in (20) are

$$b_j = \frac{2}{m}\sum_{i=1}^{m} v_i \phi_j(z_i), \qquad j = 1,\ldots,n, \tag{21}$$

and $b_0 = \sum_{i=1}^{m} v_i/m$. The method is called the Chebyshev regression algorithm in Judd (1998).

Multidimensional Complete Chebyshev Approximation

In a d-dimensional approximation problem, the domain of the approximation function will be

$$\{\mathbf{x} = (x_1, \ldots, x_d) : x_{\min,i} \leq x_i \leq x_{\max,i},\ i = 1, \ldots d\},$$

Let $\mathbf{x}_{\min} = (x_{\min,1}, \ldots, x_{\min,d})$ and $\mathbf{x}_{\max} = (x_{\max,1}, \ldots, x_{\max,d})$. We let $[\mathbf{x}_{\min}, \mathbf{x}_{\max}]$ denote the domain. Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ be a vector of nonnegative integers. Let $\phi_\alpha(\mathbf{z})$ denote the product $\prod_{i=1}^{d} \phi_{\alpha_i}(z_i)$ for $\mathbf{z} = (z_1, \ldots, z_d) \in [-1, 1]^d$. Let

$$Z(\mathbf{x}) = \left(\frac{2x_1 - x_{\min,1} - x_{\max,1}}{x_{\max,1} - x_{\min,1}}, \ldots, \frac{2x_d - x_{\min,d} - x_{\max,d}}{x_{\max,d} - x_{\min,d}} \right)$$

for any $\mathbf{x} = (x_1, \ldots, x_d) \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}]$. With this notation, the degree-n complete Chebyshev approximation for $V(\mathbf{x})$ is

$$\hat{V}(\mathbf{x}; \mathbf{b}) = \sum_{\alpha \geq 0,\, |\alpha| \leq n} b_\alpha \phi_\alpha \left(Z(\mathbf{x}) \right),$$

where $|\alpha| = \sum_{i=1}^{d} \alpha_i$. This is a degree n polynomial, and has $\binom{n+d}{d}$ terms.

Appendix C—Calibration of the Productivity Process

The productivity process, (ζ_t, χ_t), is a time-dependent, finite-state Markov chain and depends on four parameters: λ, ϱ, r, and ς. Our calibration target is to choose these parameters to match the conditional and unconditional moments of consumption growth rates from observed annual market data over the period from 1930 to 2008. For each choice of $(\lambda, \varrho, r, \varsigma)$ and several choices for the number of states, n_ζ and n_χ, we solved our model assuming damage from climate is zero (that presumably was the case for the years included in the date). The preference parameters are $\psi = 1.5$ and $\gamma = 10$. After computing the solution for each specific parameter guess, we computed 10,000 simulations

of the consumption process and computed the conditional and unconditional moments of the per capita consumption growth paths over the first century

The Markov chain approximates the following continuous processes[9]

$$\log\left(\hat{\zeta}_{t+1}\right) = \lambda \log\left(\hat{\zeta}_t\right) + \hat{\chi}_t + \varrho\hat{\omega}_{\zeta,t}, \tag{22}$$

$$\hat{\chi}_{t+1} = r\hat{\chi}_t + \varsigma\hat{\omega}_{\chi,t}, \tag{23}$$

where $\hat{\omega}_{\zeta,t}$, $\hat{\omega}_{\chi,t} \sim i.i.d.\mathcal{N}(0,1)$. The random variable $\widehat{\chi}_t$ is normal with mean 0. Denote $\Upsilon_t \equiv \mathrm{Var}\left\{\widehat{\chi}_t\right\}$. Equation (23) implies the recursion $\Upsilon_{t+1} = r^2\Upsilon_t + \varsigma^2$ and then

$$\Upsilon_t = \frac{\varsigma^2\left(1-r^{2t}\right)}{1-r^2} \tag{24}$$

for $t > 0$. The random variable $\log\left(\widehat{\zeta}_t\right)$ is also normal with mean 0, and can be expressed in terms of its disturbances and $\widehat{\chi}_t$:

$$\log\left(\widehat{\zeta}_t\right) = \sum_{s=1}^{t-1}\lambda^{t-1-s}\widehat{\chi}_s + \varrho\sum_{s=0}^{t-1}\lambda^{t-1-s}\hat{\omega}_{\zeta,s}.$$

Denote $\Delta_t \equiv \mathrm{Var}\left\{\log\left(\widehat{\zeta}_t\right)\right\}$. Since $\mathbb{E}\left\{\widehat{\chi}_t\widehat{\chi}_s\right\} = r^{|t-s|}\Upsilon_{\min\{t,s\}}$, we find that

$$\Delta_t = \mathbb{E}\left\{\left(\log\left(\widehat{\zeta}_t\right)\right)^2\right\} = \left(\sum_{s=1}^{t-1}\lambda^{2(t-1-s)}\Upsilon_s + 2\sum_{\tau=2}^{t-1}\sum_{s=1}^{\tau-1}\lambda^{2(t-1)-\tau-s}r^{\tau-s}\Upsilon_s\right) + \frac{\varrho^2\left(1-\lambda^{2t}\right)}{1-\lambda^2}, \tag{25}$$

for $t > 0$.

Our Markov chain for the bivariate process (ζ_t, χ_t) will be chosen to match the variances Υ_t in (24) and Δ_t in (25) . Our Markov chain will have bounded support, implying that we lose the tails. But the presence of the tails creates serious challenges in even proving existence of a solution to our dynamic programming problem. We choose the values of χ_t as $\{\chi_{t,j} : j = 1, ..., n_\chi\}$ where $\chi_{t,j}$ are equally spaced in $\left[-3\sqrt{\Upsilon_t}, 3\sqrt{\Upsilon_t}\right]$ (the probability that the continuously distributed random variable $\widehat{\chi}_t$ is in $\left[-3\sqrt{\Upsilon_t}, 3\sqrt{\Upsilon_t}\right]$ is

[9]To avoid confusion, we use $\widehat{\zeta}_t$ and $\widehat{\chi}_t$ to represent the continuous random variables at each time t, while ζ_t and χ_t represent the Markov chains.

99.7 percent), at each time t. Similarly, we choose the values of ζ_t as $\{\zeta_{t,i} : i = 1, ..., n_\zeta\}$ so that $\log(\zeta_{t,i})$ are equally spaced in $\left[-3\sqrt{\Delta_t}, 3\sqrt{\Delta_t}\right]$ (the probability that the continuously distributed random variable $\log\left(\widehat{\zeta}_t\right)$ is in $\left[-3\sqrt{\Delta_t}, 3\sqrt{\Delta_t}\right]$ is 99.7 percent).

Next we set the transition probability matrices for our Markov chains (ζ_t, χ_t). For the paired values $\{(\zeta_{t,i}, \chi_{t,j}) : i = 1, ..., n_\zeta, j = 1, ..., n_\chi\}$ ($\zeta_{0,i} \equiv 1$ and $\chi_{0,j} \equiv 0$), we apply the method in Tauchen (1986) to define the transition probability from $\chi_{t,j}$ to $\chi_{t+1,j'}$:

$$\begin{aligned}\Pr\left\{\chi_{t+1,j'} \mid \chi_{t,j}\right\} &= \Phi\left(\frac{1}{\varsigma}\left(\frac{\chi_{t+1,j'} + \chi_{t+1,j'+1}}{2} - r\chi_{t,j}\right)\right) \\ &\quad -\Phi\left(\frac{1}{\varsigma}\left(\frac{\chi_{t+1,j'-1} + \chi_{t+1,j'}}{2} - r\chi_{t,j}\right)\right),\end{aligned}$$

for $j' = 2, ..., n_\chi - 1$, and

$$\Pr\left\{\chi_{t+1,1} \mid \chi_{t,j}\right\} = \Phi\left(\frac{1}{\varsigma}\left(\frac{\chi_{t+1,1} + \chi_{t+1,2}}{2} - r\chi_{t,j}\right)\right),$$

$$\Pr\left\{\chi_{t+1,n} \mid \chi_{t,j}\right\} = 1 - \Phi\left(\frac{1}{\varsigma}\left(\frac{\chi_{t+1,n-1} + \chi_{t+1,n}}{2} - r\chi_{t,j}\right)\right),$$

where $\Phi(\cdot)$ is the cumulative normal distribution function, for any $j = 1, ..., n_\chi$. Similarly, the transition probability from $(\zeta_{t,i}, \chi_{t,j})$ to $\zeta_{t+1,i'}$ is defined as

$$\begin{aligned}\Pr\left\{\zeta_{t+1,i'} \mid (\zeta_{t,i}, \chi_{t,j})\right\} &= \Phi\left(\frac{1}{\varrho}\left(\frac{\log\left(\zeta_{t+1,i'}\right) + \log\left(\zeta_{t+1,i'+1}\right)}{2} - (\lambda\log(\zeta_{t,i}) + \chi_{t,j})\right)\right) \\ &\quad -\Phi\left(\frac{1}{\varrho}\left(\frac{\log\left(\zeta_{t+1,i'}\right) + \log\left(\zeta_{t+1,i'+1}\right)}{2} - (\lambda\log(\zeta_{t,i}) + \chi_{t,j})\right)\right),\end{aligned}$$

for $i' = 2, ..., n_\zeta - 1$, and

$$\Pr\left\{\zeta_{t+1,1} \mid (\zeta_{t,i}, \chi_{t,j})\right\} = \Phi\left(\frac{1}{\varrho}\left(\frac{\log(\zeta_{t+1,1}) + \log(\zeta_{t+1,2})}{2} - (\lambda\log(\zeta_{t,i}) + \chi_{t,j})\right)\right),$$

$$\Pr\left\{\zeta_{t+1,n} \mid (\zeta_{t,i}, \chi_{t,j})\right\} = 1 - \Phi\left(\frac{1}{\varrho}\left(\frac{\log(\zeta_{t+1,n-1}) + \log(\zeta_{t+1,n})}{2} - (\lambda\log(\zeta_{t,i}) + \chi_{t,j})\right)\right),$$

for any $i = 1, ..., n_\zeta, j = 1, ..., n_\chi$.

We aim to match conditional variances as well as unconditional variances. This demands a relatively fine discretization of the $\left(\widehat{\zeta}_t, \widehat{\chi}_t\right)$ space at each time. After some experimentation, we find that $n_\zeta = 33$ and $n_\chi = 19$ produce a good approximation (i.e., distributions of solutions of our benchmark example in the first 100 years are almost invariant to a larger n_ζ or n_χ) with the calibrated values at $\lambda = 0.998$, $\varrho = 0.034$, $r = 0.65$, and $\varsigma = 0.007$. Moreover, with $n_\zeta = 33$ and $n_\chi = 19$, it does not exist a transition probability that is nearly equal to one so that no transition is almost certain at any time.

Appendix D—Terminal Value Function

Value function iteration needs to specify a value function at the terminal time. We initially assume that the solutions from 2010 to 2100 is insensitive to the terminal value function, but then check that assumption. We assume that at the terminal time, the state vector is $(\tilde{K}, \tilde{\mathbf{M}}, \tilde{\mathbf{T}})$. For any time t after the terminal time, we assume that the population is $L_t = 8600$, the total production factor and the adjusted cost for backstop will be the same with the numbers at the terminal time respectively, i.e., $A_t = 0.295$ and $\theta_{1,t} = 0.008$. We assume that at the terminal time, the world reaches a partial equilibrium: after the terminal time, consumption is always 0.74 times of net ouput $\mathcal{Y}$, and the emission control rate will always be 1 so that the emission of carbon from the industry will always be 0, i.e., $\mu_t = 1$, for any year $t \geq 300$. Moreover, the economic shock disappears, i.e., $\zeta_t = 1$. Thus, the dynamics of the climate system become

$$\begin{aligned} \mathbf{M}_{t+1} &= \mathbf{\Phi}_{\mathrm{M}} \mathbf{M}_t + (E_{\mathrm{Land},t}, 0, 0)^\top, \\ \mathbf{T}_{t+1} &= \mathbf{\Phi}_{\mathrm{T}} \mathbf{T}_t + (\xi_1 \mathcal{F}_t, 0)^\top, \end{aligned}$$

for any year $t \geq 300$, where $\mathbf{M}_{300} = \tilde{\mathbf{M}}$, $\mathbf{T}_{300} = \tilde{\mathbf{T}}$, and

$$\mathcal{F}_t(M_{\mathrm{AT}}) = \eta \log_2 \left(M_{\mathrm{AT}}/M_{\mathrm{AT},0}\right) + 0.3.$$

The transition law for capital becomes

$$K_{t+1} = (1-\delta)K_t + 0.26\mathcal{Y}_t(K_t, T_{\mathrm{AT},t}, 1, 1)$$

for any year $t \geq 300$, where $K_{300} = \tilde{K}$.

Therefore, we have our terminal value function:

$$V_{300}(\tilde{K}, \tilde{\mathbf{M}}, \tilde{\mathbf{T}}, \zeta, \chi) = \sum_{t=300}^{\infty} e^{-\rho(t-300)} u(C_t, L_t).$$

The terminal value function is the sum of discounted utilities over 400 years from $t = 300$ to $t = 700$ with annual time steps. It is a good approximation of the summation of the infinite sequence.

It would be too time-consuming to use the terminal value function of the above formula in optimizers to compute optimal solutions, so we use its approximation to save computational time. In our examples, we use a simplicial complete Chebyshev polynomial approximation, $\widehat{V}_{300}(K, \mathbf{M}, \mathbf{T}, \zeta, \chi)$, over the terminal domain of the six-dimensional state space $(K, \mathbf{M}, \mathbf{T})$.