RDCEP WORKING PAPER SERIES

The Center for Robust Decision Making
on Climate and Energy Policy

DSICE: A Dynamic Stochastic Integrated Model of Climate and Economy

Yongyang Cai, Kenneth L. Judd and Thomas S. Lontzek

Working Paper No.12-02

January, 2012

DSICE: A Dynamic Stochastic Integrated Model of Climate and Economy*

Yongyang Cai
Hoover Institution, 424 Galvez Mall, Stanford University, Stanford, CA 94305
yycai@stanford.edu

Kenneth L. Judd
Hoover Institution, 424 Galvez Mall, Stanford University, Stanford, CA 94305
kennethjudd@mac.com

Thomas S. Lontzek
University of Zurich, Moussonstrasses 15, 8044 Zurich
thomas.lontzek@business.uzh.ch

January 18, 2012

Abstract

This paper introduces a dynamic stochastic integrated model of climate and economy (DSICE), and a numerical dynamic programming algorithm for its solution. More specifically, we solve an example with annual time periods, a six hundred year horizon, and shocks to the economic and climate system. Our dynamic programming methods solve such models on a laptop in about an hour, and do so with good accuracy. This decisively refutes the pessimism one often hears about the possibility of solving such models.

Keywords: numerical dynamic programming, value function iteration, tipping points, stochastic IAM

JEL Classification: C63, Q54, D81

*Cai, Judd, and Lontzek gratefully acknowledge NSF support (SES-0951576). Furthermore, financial support for Lontzek was provided by the Zürcher Universitätsverein and the Ecosciencia Foundation. Part of this study was done while Lontzek was visiting the Hoover Institution.

1 Introduction

There is great uncertainty about the future of the climate and the impact of economic activity on the climate. There is always great uncertainty about future economic conditions. Therefore, any analysis of how society should respond to possible climate change must consider the uncertainties any decision maker faces when choosing policies. This paper demonstrates that there is no difficulty in adding uncertainty and risk to basic IAM models.

We use Nordhaus' DICE2007 model as a starting point for our model. It is well-known in the IAM community and widely used in the IAM literature. Furthermore, it is well-documented. We extend it by adding both economic and climate shocks to the framework of DICE2007.

The IAM community, despite numerous attempts, has so far not been able to produce a stochastic IAM flexible enough to represent uncertainty in a quantitatively realistic manner. In particular, the representation of time should be compatible with the natural frequencies of both the natural and social processes related to climate change. Models that assume long time periods, such as ten years, represent neither social nor physical processes because nontrivial dynamics and feedbacks may occur in either system during a single decade.

Dynamic stochastic general equilibrium (DSGE) models in economics use relatively short time periods, always at most a year. DICE2007 instead uses a ten-year time period. Ten year time periods are too long for serious, quantitative analysis of policy questions. For example, if one wants to know how carbon prices should react to business cycle shocks, the time period needs to be at most a year. No one would accept a policy that takes ten years to respond to current shocks to economic conditions. Therefore, we first develop DICE-CJL, an extension of DICE2007 that can handle any time period length. We demonstrate that many substantive results depend critically on the time step, strongly supporting our contention that short time periods are necessary for quantitatively reliable analysis.

We then take the DICE-CJL and add economic and climate shocks. We solve it using dynamic programming (DP) methods. In particular, we use the methods presented in Judd (1998), Cai (2009), and Cai and Judd (2010, 2012a, 2012b, 2012c).

There are many different types of uncertainty that are discussed in the IAM literature. First, many people examine parametric uncertainty, because we do not know with precision the value of key parameters. Second, economic models have substantial amounts of intrinsic uncertainty, meaning that even if one knew the parameter, there would still be uncertainty due to random exogenous events.

DSICE is a model that focuses on intrinsic uncertainty, as is done in the DSGE literature in economics. However, the speed of our DSICE solution algorithm is fast enough that we could also do wide-ranging parameter sweeps to address parameter uncertainty.

We demonstrate the performance of DSICE in a few examples. Those examples show that our algorithm is fast (about an hour on a single-processor laptop to solve a one-year, 600-period stochastic model with eight state variables), and passes basic accuracy tests.

The results show decisively that climate change issues can be examined with the same complexity used in standard dynamic stochastic models in economics.

2 A Brief Summary of DICE2007

In DICE2007, Nordhaus (2008) uses a social welfare-maximizing problem to model the tradeoffs between CO_2 abatement, consumption, and investment. He assumes ten-year time periods, and maximizes social utility subject to economic and climate constraints. Nordhaus (2008) solves the problem

$$\begin{aligned}
\max_{c_t,\mu_t} \quad & \sum_{t=0}^{60} \beta^{10t} 10 u(c_t, l_t) \\
\text{s.t.} \quad & k_{t+1} = (1-\delta)^{10} k_t + 10\left(\Omega_t (1-\Lambda_t) Y_t - c_t\right), \\
& \mathbf{M}_{t+1} = \mathbf{\Phi}^{\mathrm{M}} \mathbf{M}_t + (E_t, 0, 0)^{\top}, \\
& \mathbf{T}_{t+1} = \mathbf{\Phi}^{\mathrm{T}} \mathbf{T}_t + (\xi_1 F_t, 0)^{\top}.
\end{aligned}$$

where t is the time in units of decades. At each time t, the social planner has two control variables, annual consumption c_t and emission control rate μ_t. The annual utility function, $u(c,l)$, is the power utility function,

$$u(c,l) = \frac{(c/l)^{1-\gamma} - 1}{1-\gamma} l,$$

and $\beta = 1.015^{-1}$ is the annual discount factor. DICE2007 assumes $\gamma = 2$.

The production side of DICE2007 is a basic optimal growth model. Output in time t, denoted Y_t, is produced from capital, k_t (measured in trillions of 2005 U.S. dollars), and labor supply

$$l_t = 6514e^{-0.35t} + 8600(1 - e^{-0.35t})$$

which equals world population in millions of people. World gross output per year during decade t is

$$Y_t = A_t k_t^{\alpha} l_t^{1-\alpha},$$

where $\alpha = 0.3$ is the capital share and A_t is total productivity factor which is defined by

$$A_0 = 0.02722,$$

$$A_{t+1} = \frac{A_t}{1 - 0.092e^{-0.01t}}.$$

The annual rate of depreciation of capital is $\delta = 0.1$.

DICE2007 uses a simple box model of the climate. Total carbon emissions (billions of metric tons) per decade is

$$E_t = E_t^{\mathrm{Ind}} + E_t^{\mathrm{Land}},$$

where $E_t^{\mathrm{Land}} = 11 \times 0.9^t$ represents emissions from biological processes and

$$E_t^{\mathrm{Ind}} = \sigma_t(1 - \mu_t)Y_t$$

is the rate of emissions from industrial activity. Industrial emissions are affected by the technology, represented by σ_t, the ratio of industrial emissions to output (metric tons of carbon per output in 2005 prices) if there is no emission control. The technology factor σ_t follows the path

$$\sigma_0 = 0.13418,$$
$$\sigma_{t+1} = \frac{\sigma_t}{1 + 0.073e^{-0.03t}}.$$

Abatement efforts reduce emissions by a factor μ_t, which reduces output by

$$\Lambda_t \equiv \psi_t^{1-\theta_2}\theta_{1,t}\mu_t^{\theta_2},$$

where $\theta_2 = 2.8$, ψ_t is the participation rate, and

$$\theta_{1,t} = 1.17\sigma_t/\theta_2 \times (1 + e^{-0.05t})/2$$

is the adjusted cost for backstop. CO_2 concentration is modeled by a three-layer model, with

$$\mathbf{M}_t = (M_t^{\mathrm{AT}}, M_t^{\mathrm{UP}}, M_t^{\mathrm{LO}})^\top$$

representing carbon concentration (billions of metric tons) in the atmosphere (${M_t}^{\mathrm{AT}}$), upper oceans (${M_t}^{\mathrm{UP}}$) and lower oceans (${M_t}^{\mathrm{LO}}$) . $\mathbf{\Phi}^{\mathrm{M}}$ is the carbon diffusion matrix (flows per decade),

$$\mathbf{\Phi}^{\mathrm{M}} = \begin{bmatrix} 0.810712 & 0.097213 & 0 \\ 0.189288 & 0.852787 & 0.003119 \\ 0 & 0.05 & 0.996881 \end{bmatrix}.$$

The global mean temperature is represented by a two-layer model, with

$$\mathbf{T}_t = (T_t^{\mathrm{AT}}, T_t^{\mathrm{LO}})^\top$$

representing temperature (measured in degrees Celsius above the 1900 temperature) of atmosphere (T_t^{AT}) and lower oceans (T_t^{LO}). $\mathbf{\Phi}^{\mathrm{T}}$ is the climate temperature diffusion matrix per decade,

$$\mathbf{\Phi}^{\mathrm{T}} = \begin{bmatrix} 1-\xi_1\left(\frac{\eta}{3}-0.3\right) & 0.3\xi_1 \\ 0.05 & 0.95 \end{bmatrix},$$

where $\eta = 3.8$ and $\xi_1 = 0.22$ is the climate-equation coefficient for the upper level.

Temperature affects output. The damage function in DICE2007 is

$$\Omega_t \equiv \frac{1}{1+\pi_1 T_t^{\mathrm{AT}} + \pi_2 (T_t^{\mathrm{AT}})^2}$$

with $\pi_1 = 0$ and $\pi_2 = 0.0028388$. The total radiative forcing (watts per square meter from 1900) is

$$F_t = \eta \log_2\left(M_t^{\mathrm{AT}}/M_0^{\mathrm{AT}}\right) + F_t^{\mathrm{EX}},$$

where F_t^{EX} is a sequence of deterministic exogenous radiative forcing.

3 DICE-CJL

In all versions of DICE, the time interval of one period is 10 years, i.e., governments could only have one chance per decade to adjust their economic and climate policy. Moreover, a 10-year time difference is too large in the finite difference method for discretizing the continuous time differential system.

We modify it into a general model with any smaller time interval h by adjusting the parameters accordingly. In the following, t is the time in units of h years, where the time interval h could be smaller than 1. The carbon cycle transition matrix (flows per period) becomes

$$\mathbf{\Phi}_h^{\mathrm{M}} = \begin{bmatrix} 1-0.0189288h & 0.0097213h & 0 \\ 0.0189288h & 1-0.0147213h & 0.0003119h \\ 0 & 0.005h & 1-0.0003119h \end{bmatrix},$$

the ratio of uncontrolled industrial emissions to output per period is

$$\sigma_{t+1,h} = \frac{\sigma_{t,h}}{1+0.0073e^{-0.003th}h},$$

and the total carbon emissions (billions of metric tons) from land per period is

$$E_{t,h}^{\mathrm{Land}} = 1.1h \times (1-0.01h)^t.$$

The climate temperature transfer matrix per period becomes

$$\mathbf{\Phi}_h^{\mathrm{T}} = \begin{bmatrix} 1-\xi_{1,h}\left(\frac{\eta}{3}-0.3\right) & 0.3\xi_{1,h} \\ 0.005h & 1-0.005h \end{bmatrix},$$

where $\xi_{1,h} = 0.022h$. The population and labor input in millions of people at time t is

$$l_{t,h} = 6514e^{-0.035th} + 8600(1 - e^{-0.035th}).$$

World gross output per year at time t is

$$Y_{t,h} = A_{t,h} k_{t,h}^{\alpha} l_{t,h}^{1-\alpha},$$

where

$$A_{t+1,h} = \frac{A_{t,h}}{1 - 0.0092e^{-0.001th}h}.$$

The abatement-cost function is

$$\Lambda_{t,h} \equiv \psi_{t,h}^{1-\theta_2} \theta_{1,t,h} \mu_{t,h}^{\theta_2}$$

where

$$\theta_{1,t,h} = 1.17\sigma_{t,h}/\theta_2 \times (1 + e^{-0.005th})/2.$$

This model with a flexible time interval is called DICE-CJL (Cai, Judd and Lontzek).

In order to simplify the notations, in the following we will always assume that h is the time interval of one period and we will cancel h in the subscript of the above notations.

4 Terminal Value Function in DICE-CJL

A finite horizon with 60 periods (each period has 10 years) approximates the infinite horizon problem in DICE2007, and it assumes that the terminal value function is 0 everywhere. This is not a reasonable assumption, as this implies that people will consume the entire capital stock before the terminal time and do not control carbon emission or temperature at the last period. In order to avoid this, DICE2007 imposes an additional constraint on terminal capital by assuming that investment at the terminal time must be at least 2% of the capital stock at the terminal time.

In DICE-CJL, we cancel the additional constraint and use a more meaningful terminal value function instead. We assume that at the terminal time (the 600th year), the capital is k_{600}, the three-layer CO_2 concentration is $\mathbf{M}_{600}$, the two-layer global mean temperature is $\mathbf{T}_{600}$, the abatement cost factor of emissions reduction is Λ_{600}, and the world gross output per year is Y_{600}. We assume that $h = 1$ for the dynamic system after the terminal time. DICE-CJL assumes that at the terminal time, the world reaches a partial equilibrium: after the terminal time, capital, population, world gross output per year and the abatement cost factor of emissions reduction will be the same, emission control rate will always be 1, and emission of carbon will always be 0 so that total amount of carbon will stay the same. That is, for any year t after the terminal time, $k_t = k_{600}$, $l_t = 8600$, $Y_t = Y_{600}$, $\Lambda_t = \Lambda_{600}$, $\mu_t = 1$, and $E_t = 0$. Thus, for any year t after the terminal time, the dynamics of the climate system becomes

$$\mathbf{M}_{t+1} = \mathbf{\Phi}^{\mathrm{M}}\mathbf{M}_t,$$

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

where

$$F_t = \eta \log_2 \left(M_t^{\mathrm{AT}}/M_0^{\mathrm{AT}}\right) + 0.3.$$

To keep the above partial equilibrium, the consumption will be

$$c_t = \Omega_t(1-\Lambda_t)Y_t - \delta k_t,$$

where

$$\Omega_t \equiv \frac{1}{1 + \left(\pi_1 T_t^{\mathrm{AT}} + \pi_2 \left(T_t^{\mathrm{AT}}\right)^2\right) I_{T_t^{\mathrm{AT}}>0}}$$

where $I_{T_t^{\mathrm{AT}}>0}$ is the indicator function (1 if $T_t^{\mathrm{AT}} > 0$, and 0 elsewhere), for the years after terminal time.

Therefore, we have our terminal value function:

$$V(k_{600}, \mathbf{M}_{600}, \mathbf{T}_{600}) = \sum_{t=601}^{\infty} \beta^{t-601} u(c_t, l_t).$$

To compute the terminal value function, we will use the summation of discounted utilities over 800 years from $t = 601$ to $t = 1400$ with one year as the time interval for each period instead. It will be a very good approximation of the summation of the infinite sequence, because $\beta^{800} = 6.7 \times 10^{-6}$ is small enough. That is,

$$V(k_{600}, \mathbf{M}_{600}, \mathbf{T}_{600}) = \sum_{t=601}^{1400} \beta^{t-601} u(c_t, l_t).$$

It would be too time-consuming to use the terminal value function of the above formula in optimizers to compute optimal solutions, so we will use its approximation to save computational time. In our examples, we will use complete Chebyshev polynomials over the 6-dimensional state space, where we let $k_{600} \in [40000, 160000]$, $M_{600}^{\mathrm{AT}} \in [720, 1350]$, $M_{600}^{\mathrm{UP}} \in [1280, 2400]$, $M_{600}^{\mathrm{LO}} \in [16000, 30000]$, $T_{600}^{\mathrm{AT}} \in [1, 3]$, and $T_{600}^{\mathrm{LO}} \in [1.5, 3.5]$.

5 DICE-CJL Numerical Results

Numerically, if h is too large, then the numerical errors may become so large that the solutions may be not trusted. We use various time intervals to see the difference of solutions of DICE-CJL with these different h.

5.1 Solving DICE-CJL with a Reliable Optimizer

We use CONOPT in the GAMS environment (McCarl, 2011) to solve DICE-CJL with different time period lengths in our GAMS code. The terminal value function is approximated by the degree-4 complete Chebyshev polynomial. When the time interval is smaller, it becomes more challenging. In DICE-CJL, if $h = 10$ years, neither the number of variables nor the number of constraints is larger than 1000 in our GAMS code, but if $h = 0.25$ year, both the number of variables and the number of constraints are larger than 20000.

We will solve this large-scale problem using a good initial guess. The good initial guess is the linear interpolation of optimal solutions of DICE-CJL with a larger time interval. For example, we will use the linear interpolation of optimal solutions of the 2-year DICE-CJL as the initial guess for the annual DICE-CJL model.

Figure 1 plots the optimal capital paths over the first 200 years of DICE-CJL with various time intervals $h = 10$, 8, 4, 2, 1, 0.5, and 0.25 years. We see that DICE-CJL with a smaller h has a smaller capital stock, because more money is spent in CO_2 emission control (particularly in the early years), see Figure 4 for the optimal emission control rate. Therefore, DICE-CJL with a smaller h has a smaller amount of carbon concentration in the atmosphere (Figure 2), and also a lower global mean surface temperature (Figure 3). Figure 2 tells us that carbon concentration in the atmosphere will reach its peak after around 200 years, and then drop slowly over the latter 400 years on the optimal paths. And Figure 3 shows that surface temperature will reach its peak after around 230 years, and then drop slowly. Moreover, after around 230 years, emission control rate will always be 1 so that CO_2 emission from industry is 0.

The running time for each DICE-CJL with $h \geq 0.25$ years is less than 1 minute in the GAMS environment on a Mac with a processor of 2.8 GHz Intel Core 2 Duo and 8GB memory.

5.2 Richardson Extrapolation

The true model is a continuous-time model. We hope that the solutions converge to the continuous-time solution as we reduce the time period. Also, our optimizer may not give a good optimal solution with a very small time step. In fact, if we choose one month as the time interval of each period over

Figure 1: Capital in DICE-CJL with Various Time Intervals

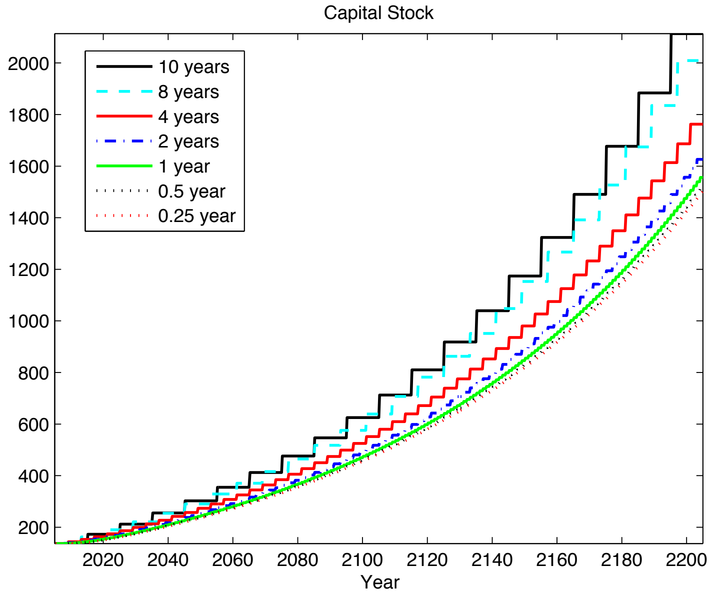


Figure 2: Carbon Concentration in the Atmosphere in DICE-CJL with Various Time Intervals

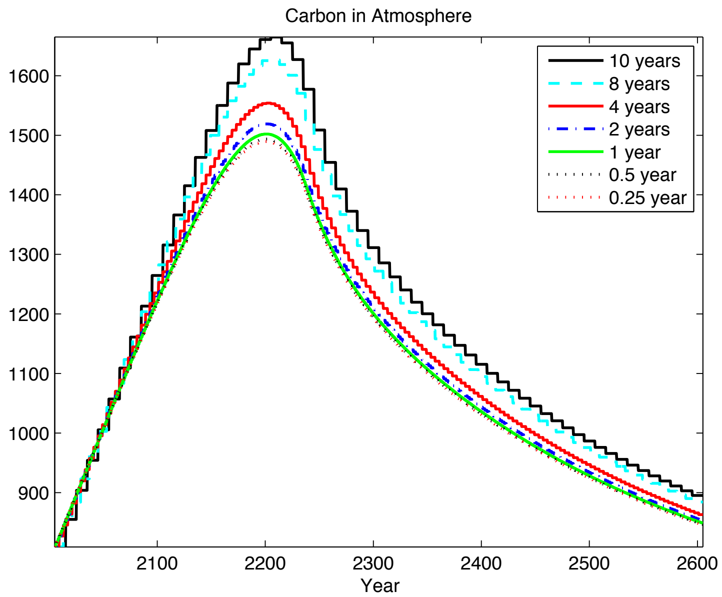


Figure 3: Surface Temperature in DICE-CJL with Various Time Intervals

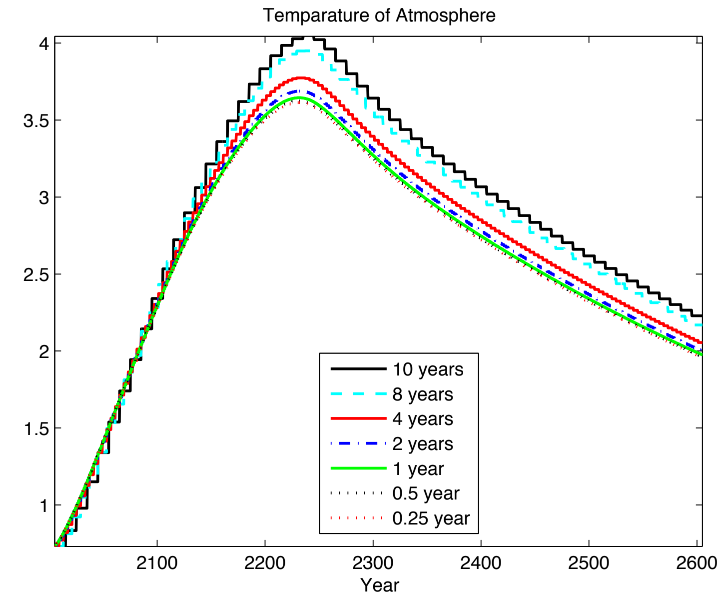


Figure 4: Emission Control Rate in DICE-CJL with Various Time Intervals

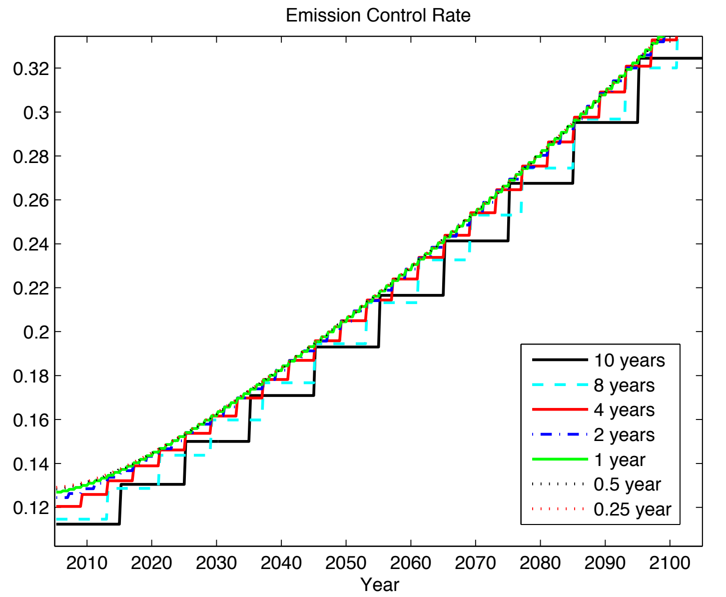
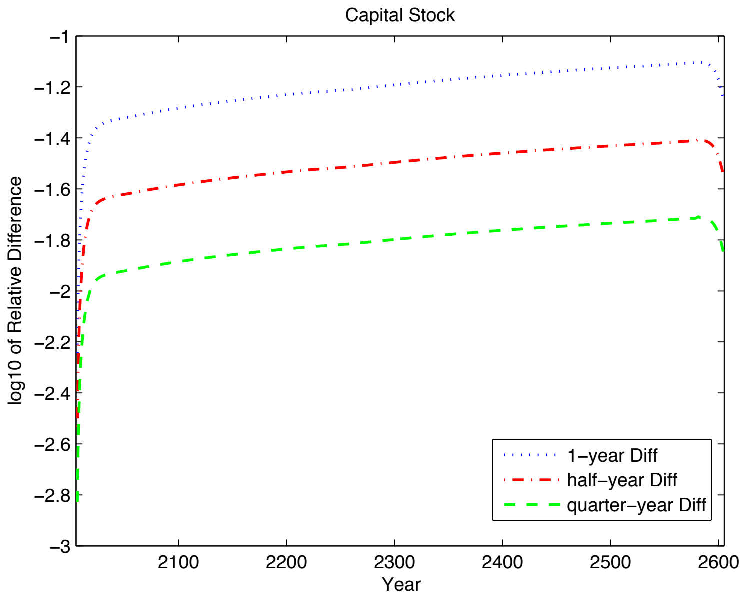


Figure 5: Richardson Extrapolation for Capital



600 years, CONOPT fails to find an optimal solution. For both reasons, we want to check if our discrete time solutions converge to a common limit.

Richardson extrapolation (Richardson and Gaunt, 1927) is a standard way to check if our solutions are consistent with convergence. We applied Richardson extrapolation to our solutions and found that they are consistent with convergence. The next figures show that our solutions are good. The vertical axis in each figure is

$$\log_{10}\left(\left|\frac{x_{t,h}^{*}-x_{t}^{\mathrm{R}}}{x_{t}^{\mathrm{R}}}\right|+10^{-6}\right),$$

where $x_{t,h}^{*}$ is the optimal solution at year t of DICE-CJL with h as the time interval for $h = 1$, 0.5, or 0.25 year, x_{t}^{R} is the 3-point Richardson extrapolation of $x_{t,1}^{*}$, $x_{t,0.5}^{*}$, and $x_{t,0.25}^{*}$, i.e.

$$x_{t}^{\mathrm{R}}=\frac{1}{3}\left(8x_{t,0.25}^{*}-6x_{t,0.5}^{*}+x_{t,1}^{*}\right).$$

Each line represents the difference between the solution for a time step and the Richardson extrapolant. We see that each variable at each time is moving uniformly towards the extrapolant as we reduce the time step.

Figure 6: Richardson Extrapolation for Carbon in Atmosphere

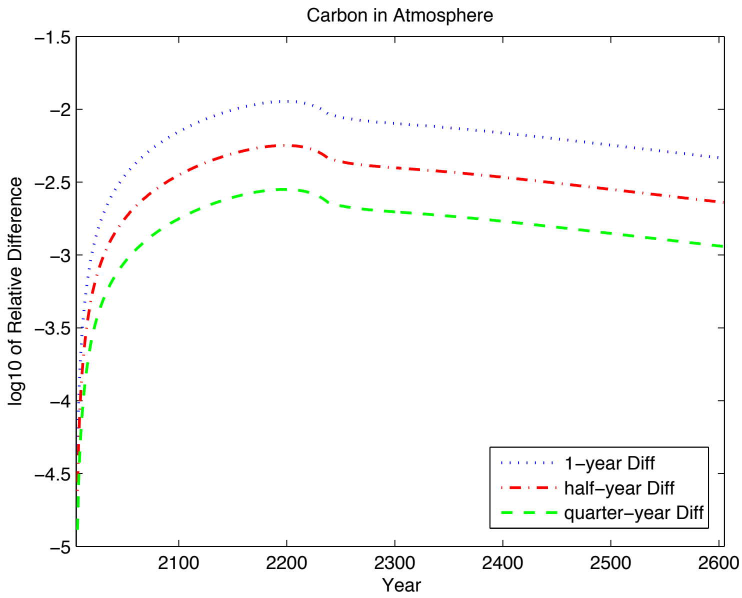


Figure 7: Richardson Extrapolation for Temperature in Atmosphere

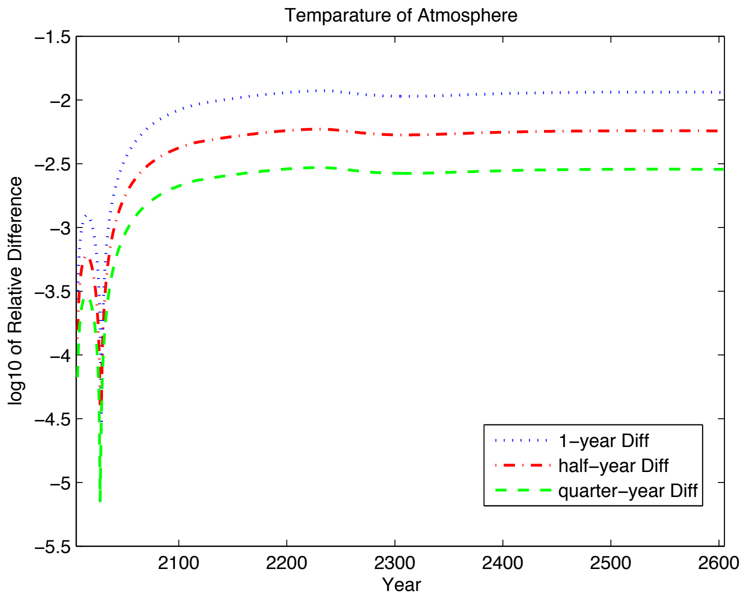
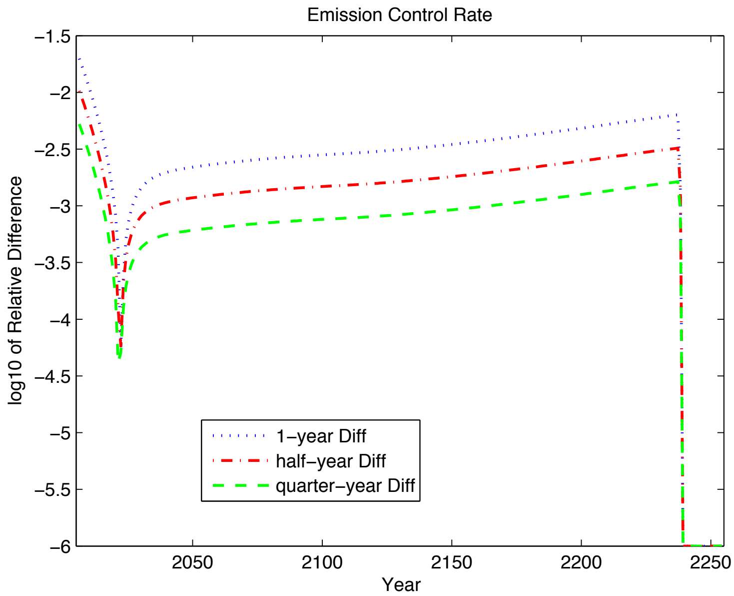


Figure 8: Richardson Extrapolation for Emission Control Rate



6 DSICE

Now that we have a solid understanding of the deterministic model, we are ready to extend it to include stochastic elements. We add two stochastic shocks to DICE-CJL, one representing an economic shock and the other representing a climate event that occurs at some random time. The resulting optimization problem is

$$
\begin{aligned}
\max_{c_t,\mu_t} \quad & \mathbb{E}\left\{\sum_{t=0}^{\infty}\beta^{th}u(c_t,l_t)h\right\} \\
\text{s.t.} \quad & k_{t+1}=(1-\delta)^h k_t+\left(\Omega_t(1-\Lambda_t)Y_t-c_t\right)h, \\
& \mathbf{M}_{t+1}=\mathbf{\Phi}^{\mathrm{M}}\mathbf{M}_t+(E_t,0,0)^\top, \\
& \mathbf{T}_{t+1}=\mathbf{\Phi}^{\mathrm{T}}\mathbf{T}_t+(\xi_1 F_t,0)^\top, \\
& \zeta_{t+1}=g^\zeta(\zeta_t,\omega_t^\zeta), \\
& J_{t+1}=g^J(J_t,\omega_t^J)
\end{aligned}
$$

where $\mathbb{E}\{\cdot\}$ is the expectation operator,

$$Y_t \equiv \zeta_t A_t k_t^\alpha l_t^{1-\alpha}$$

ζ_t is a stochastic productivity shock corresponding to economic fluctuations, and

$$\Omega_t \equiv \frac{J_t}{1 + \pi_1 T_t^{\mathrm{AT}} + \pi_2 (T_t^{\mathrm{AT}})^2}$$

is a new component of the damage function with a stochastic shock J_t, where ω_t^ζ and ω_t^J are two independent random variables, and g^ζ and g^J are transition functions of ζ_t and J_t, respectively.

The novel element is the shock J_t. It is a jump process that initially equals 1 but then may fall at some future time with the hazard rate of decline related to the contemporaneous temperature. See Lontzek, Cai, and Judd (2012) for a more complete discussion of the details.

The DP model for DSICE is

$$\begin{aligned}
V_t(k, \mathbf{M}, \mathbf{T}, \zeta, J) = \max_{c,\mu} \quad & u(c, l_t)h + \beta \mathbb{E}\{V_{t+1}(k^+, \mathbf{M}^+, \mathbf{T}^+, \zeta^+, J^+)\} \\
\text{s.t.} \quad & k^+ = (1-\delta)^h k + (\Omega_t(1-\Lambda_t) f(k, l_t, \zeta, t) - c)\, h, \\
& \mathbf{M}^+ = \mathbf{\Phi}^{\mathrm{M}} \mathbf{M} + (E_t, 0, 0)^\top, \\
& \mathbf{T}^+ = \mathbf{\Phi}^{\mathrm{T}} \mathbf{T} + (\xi_1 F_t, 0)^\top, \\
& \zeta^+ = g^\zeta(\zeta, \omega^\zeta), \\
& J^+ = g^J(J, \omega^J),
\end{aligned}$$

where V_t is value function at stage t, consumption c and emission control rate μ are two control variables, $(k, \mathbf{M}, \mathbf{T}, \zeta, J)$ is 8-dimensional state vector at stage t (where $\mathbf{M} = (M^{\mathrm{AT}}, M^{\mathrm{UP}}, M^{\mathrm{LO}})^\top$ is the three-layer CO_2 concentration and $\mathbf{T} = (T^{\mathrm{AT}}, T^{\mathrm{LO}})^\top$ is the two-layer global mean temperature), and $(k^+, \mathbf{M}^+, \mathbf{T}^+, \zeta^+, J^+)$ is its next-stage state vector.

7 Numerical Methods for DP

Before discussing examples of DSICE, we summarize the numerical methods we used to solve the dynamic programming problem. In 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 parameterizable collection of functions to approximate value functions, $V(x, \theta) \approx \hat{V}(x, \theta; \mathbf{b})$, where x is the continuous state vector (in DSICE, it is the 6-dimensional vector $(k, \mathbf{M}, \mathbf{T})$), θ is the discrete state vector (in DSICE, it is the 2-dimensional vector (ζ, J)), and $\mathbf{b}$ 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 specially designed for the problem. After the functional form is fixed, we focus on finding the vector of parameters, $\mathbf{b}$, such that $\hat{V}(x,\theta;\mathbf{b})$ approximately satisfies the Bellman equation (Bellman, 1957). Numerical DP with value function iteration can solve the Bellman equation approximately (Judd, 1998).

A general DP model is based on the Bellman equation:

$$\begin{aligned} V_t(x,\theta) = \max_{a\in\mathcal{D}(x,\theta,t)} \quad & u_t(x,a)+\beta\mathbb{E}\left\{V_{t+1}(x^+,\theta^+)\mid\theta\right\}, \\ \text{s.t.} \quad & x^+ = f(x,\theta,a), \\ & \theta^+ = g(\theta,\omega), \end{aligned}$$

where $V_t(x,\theta)$ is called the value function at time $t \le T$ (the terminal value function $V_T(x,\theta)$ is given), (x^+,θ^+) is the next-stage state, $\mathcal{D}(x,\theta,t)$ is a feasible set of a, ω is a random variable, and $u_t(x,a)$ is the utility function at time t. The following is the algorithm of parametric DP 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, $X_t = \{x_{it} : 1 \le i \le m_t\}$ for every $t < T$, and choose a functional form for $\hat{V}(x,\theta;\mathbf{b})$, where $\theta \in \Theta$. Let $\hat{V}(x,\theta;\mathbf{b}_T) \equiv V_T(x,\theta)$. Then for $t = T-1, T-2, \ldots, 0$, iterate through steps 1 and 2.*

Step *1. Maximization step. Compute*

$$\begin{aligned} v_{i,j} = \max_{a_{i,j}\in\mathcal{D}(x_i,\theta_j,t)} \quad & u_t(x_i,a_{ij})+\beta\mathbb{E}\left\{\hat{V}(x_{i,j}^+,\theta_j^+;\mathbf{b}_{t+1})\mid\theta_j\right\} \\ \text{s.t.} \quad & x_{i,j}^+ = f(x_i,\theta_j,a_{i,j}), \\ & \theta_j^+ = g(\theta_j,\omega), \end{aligned}$$

for each $\theta_j \in \Theta$, $x_i \in X_t$, $1 \le i \le m_t$.

Step *2. Fitting step. Using an appropriate approximation method, compute the $\mathbf{b}_t$ such that $\hat{V}(x,\theta_j;\mathbf{b}_t)$ approximates $(x_i, v_{i,j})$ data for each $\theta_j \in \Theta$.*

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 (2009), Judd (1998) and Rust (2008).

8 Approximation

An approximation scheme consists of two parts: basis functions and approximation nodes. Approximation nodes can be chosen as uniformly spaced nodes, Chebyshev nodes, or some other specified nodes. From the viewpoint of basis functions, approximation methods can be classified as either spectral methods or finite element methods. A spectral method uses globally nonzero basis functions $\phi_j(x)$ such that $\hat{V}(x;\mathbf{c}) = \sum_{j=0}^{n} c_j\phi_j(x)$ is a degree-n approximation. Examples of spectral methods include ordinary polynomial approximation, ordinary Chebyshev polynomial approximation, and shape-preserving Chebyshev polynomial approximation (Cai and Judd, 2012c). In contrast, a finite element method uses local basis functions $\phi_j(x)$ that are nonzero over sub-domains of the approximation domain. Examples of finite element methods include piecewise linear interpolation, shape-preserving rational function spline interpolation (Cai and Judd, 2012b), cubic splines, and B-splines. See Cai (2009), Cai and Judd (2010), and Judd (1998) for more details.

8.1 Chebyshev Polynomial Approximation

Chebyshev polynomials on $[-1, 1]$ are defined as $T_j(z) = \cos(j\cos^{-1}(z))$, while general Chebyshev polynomials on $[a, b]$ are defined as $T_j((2x - a - b)/(b - a))$ for $j = 0, 1, 2, \ldots$. These polynomials are orthogonal under the weighted inner product: $\langle f, g\rangle = \int_a^b f(x)g(x)w(x)dx$ with the weighting function $w(x) = \left(1 - ((2x - a - b)/(b - a))^2\right)^{-1/2}$. A degree n Chebyshev polynomial approximation for $V(x)$ on $[a, b]$ is

$$\hat{V}(x;\mathbf{c}) = \frac{1}{2}c_0 + \sum_{j=1}^{n} c_j T_j\left(\frac{2x - a - b}{b - a}\right), \tag{1}$$

where c_j are the Chebyshev coefficients.

If we choose the Chebyshev nodes on $[a, b]$: $x_i = (z_i + 1)(b - a)/2 + a$ with $z_i = -\cos\left((2i - 1)\pi/(2m)\right)$ for $i = 1, \ldots, m$, and Lagrange data $\{(x_i, v_i) : i = 1, \ldots, m\}$ are given (where $v_i = V(x_i)$), then the coefficients c_j in (1) can be easily computed by the following formula,

$$c_j = \frac{2}{m}\sum_{i=1}^{m} v_i T_j(z_i), \qquad j = 0, \ldots, n. \tag{2}$$

The method is called the Chebyshev regression algorithm in Judd (1998).

It is often more stable to use the expanded Chebyshev polynomial interpolation (Cai and Judd, 2012a), as the above standard Chebyshev polynomial interpolation gives poor approximation in the neighborhood of end points. That is, we use the following formula to approximate $V(x)$,

$$\hat{V}(x;\mathbf{c}) = \frac{1}{2}c_0 + \sum_{j=1}^{n} c_j T_j\left(\frac{2x - \widetilde{a} - \widetilde{b}}{\widetilde{b} - \widetilde{a}}\right), \tag{3}$$

where $\widetilde{a} = a - \delta$ and $\widetilde{b} = b + \delta$ with $\delta = (z_1 + 1)(a - b)/(2z_1)$. Moreover, if we choose the expanded Chebyshev nodes on $[a, b]$: $x_i = (z_i+1)(\widetilde{b}-\widetilde{a})/2+\widetilde{a}$, then the coefficients c_j can also be calculated easily by the expanded Chebyshev regression algorithm (Cai, 2009), which is similar to (2).

8.2 Multidimensional Complete Chebyshev Approximation

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

$$\{x = (x_1, \ldots, x_d) : a_i \le x_i \le b_i,\, i = 1, \ldots d\},$$

for some real numbers a_i and b_i with $b_i > a_i$ for $i = 1, \ldots, d$. Let $a = (a_1, \ldots, a_d)$ and $b = (b_1, \ldots, b_d)$. Then we denote $[a, b]$ as the domain. Let $\alpha = (\alpha_1, \ldots, \alpha_d)$ be a vector of nonnegative integers. Let $T_\alpha(z)$ denote the product $T_{\alpha_1}(z_1) \cdots T_{\alpha_d}(z_d)$ for $z = (z_1, \ldots, z_d) \in [-1, 1]^d$. Let

$$Z(x) = \left(\frac{2x_1 - a_1 - b_1}{b_1 - a_1}, \ldots, \frac{2x_d - a_d - b_d}{b_d - a_d} \right)$$

for any $x = (x_1, \ldots, x_d) \in [a, b]$.

Using these notations, the degree-n complete Chebyshev approximation for $V(x)$ is

$$\hat{V}_n(x; \mathbf{c}) = \sum_{0 \le |\alpha| \le n} c_\alpha T_\alpha \left(Z(x) \right),$$

where $|\alpha| = \sum_{i=1}^d \alpha_i$ for the nonnegative integer vector $\alpha = (\alpha_1, \ldots, \alpha_d)$. So the number of terms with $0 \le |\alpha| = \sum_{i=1}^d \alpha_i \le n$ is $\binom{n+d}{d}$ for the degree-n complete Chebyshev approximation in $\mathbb{R}^d$.

Let

$$z^{(k)} = \left(z_1^{(k_1)}, \ldots, z_d^{(k_d)} \right) \in [-1, 1]^d,$$

where $k = (k_1, \ldots, k_d)$, $z_i^{(k_i)} = -\cos\left((2k_i - 1)\pi/(2m)\right)$ for $k_i = 1, \ldots, m$, and $i = 1, \ldots, d$. Let

$$x_i^{(k_i)} = (z_i^{(k_i)} + 1)(b_i - a_i)/2 + a_i,$$

for $i = 1, \ldots, d$, and then

$$x^{(k)} = \left(x_1^{(k_1)}, \ldots, x_d^{(k_d)} \right)$$

is a d-dimensional Chebyshev node in $[a, b]$. These $x^{(k)}$ (for all $k_i = 1, \ldots, m$ and $i = 1, \ldots, d$) forms the set of the d-dimensional Chebyshev nodes with m nodes in each dimension. For each $x^{(k)}$, let $v^{(k)} = V(x^{(k)})$ be computed by solving the Bellman equation at $x^{(k)}$. Then the coefficients of the degree-n complete Chebyshev approximation on $[a, b]$ are computed as

$$c_\alpha = \frac{2^{\tilde{d}}}{m^d} \sum_{1 \le k_i \le m, 1 \le i \le d} v_{(k)} T_\alpha(z^{(k)}),$$

where $\tilde{d} = \sum_{i=1}^{d} 1_{\alpha_i > 0}$ with $1_{\alpha_i > 0}$ as the indicator

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

for all nonnegative integer vectors α with $0 \le |\alpha| \le n$.

We can easily extend this multidimensional complete Chebyshev approximation and the formula to compute the Chebyshev coefficients to its expanded version over $[a, b]$.

9 Choosing Domains of Value Functions

In our examples, $(k, \mathbf{M}, \mathbf{T})$ is the 6-dimensional vector of continuous states, and (ζ, J) is 2-dimensional vector of discrete states. All of our examples assume that we solve a 600-year horizon problem. As is assumed in DICE, we believe that this solution of the first few centuries are a good approximation of the solution of the first few centuries to the infinite horizon problem in the model description.

It is important to set an appropriate domain for approximating the value functions. We use the solution of DICE-CJL to tell us how to construct the domain of the value function at each time. Over these 600 years, the optimal solution of the annual DICE-CJL tells us that the minimal capital is 137, and the maximal capital is 75535 along the optimal path of capital. Therefore, if we use a fixed domain along the time path, then the domain will be too large. The problem becomes more difficult when we include the stochastic states, particularly the tipping point shock, in DSICE.

To overcome this difficult problem, we let the domains vary along the time path. We use the optimal solution of DICE-CJL to generate the domains of the value functions along time t, and keep the optimal state variables of DICE-CJL at around the center of the domain. When there is a tipping point shock in the model, the optimal state variables of the adjusted DICE-CJL should also be around the center of the domain, where the adjusted DICE-CJL assume that the tipping point happens at the first year.

To choose the domains with the above properties, we set the range of capital k_t along time t to be

$$\left[0.8 \min_{i,j} \left\{\zeta_i k^*_{t,J_j}\right\}, 1.5 \max_{i,j} \left\{\zeta_i k^*_{t,J_j}\right\}\right],$$

where ζ_i and J_j are possible values of ζ and J respectively, k^*_{t,J_j} is the optimal capital at time t of the adjusted DICE-CJL with the damage function

$$\Omega_t \equiv \frac{J_j}{1 + \pi_1 T_t^{\mathrm{AT}} + \pi_2 (T_t^{\mathrm{AT}})^2}.$$

The ranges of the other continuous states are defined in the following way:

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

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

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

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

where $\underline{\mathbf{M}}_t$ and $\underline{\mathbf{T}}_t$ are the lower bounds of $\mathbf{M}_t$ and $\mathbf{T}_t$ respectively, $\overline{\mathbf{M}}_t$ and $\overline{\mathbf{T}}_t$ are the upper bounds of $\mathbf{M}_t$ and $\mathbf{T}_t$ respectively,

$$\underline{F}_t = \eta \log_2 \left(\underline{M}_t^{\mathrm{AT}} / M_0^{\mathrm{AT}} \right) + F_t^{\mathrm{EX}},$$

$$\overline{F}_t = \eta \log_2 \left(\overline{M}_t^{\mathrm{AT}} / M_0^{\mathrm{AT}} \right) + F_t^{\mathrm{EX}},$$

and $\underline{E}_t$ and $\overline{E}_t$ are the lower and the upper bounds of the optimal emission at time t.

10 Accuracy Test

An accuracy test is very important for any numerical algorithm. A numerical algorithm should not be trusted if we have not examined its accuracy. Here again our DICE-CJL model helps us. Since we can solve DICE-CJL by our GAMS code directly without using numerical DP algorithms, it will be a natural way to compare the solutions given by the numerical DP algorithm and the solutions given by the GAMS code using a large-scale nonlinear optimizer for DICE-CJL , because DICE-CJL are degenerated cases of DSICE.

Our Fortran code of numerical DP algorithm (the deterministic version of Algorithm 1) is applied to solve DICE-CJL. 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. For each dimension of the continuous state space, we choose 5 expanded Chebyshev nodes, and then use the tensor rule to generate all points. In our examples, the number of points is $5^6 = 15625$. Therefore, for each value function iteration, we compute 15625 values of the value function at these points, and then compute Chebyshev coefficients of a degree-4 expanded complete Chebyshev polynomial to approximate the value function.

After computing the Chebyshev coefficients for all stages along the 600 years using the backward value function iteration method, we generate the optimal path with the given initial state in the GAMS code 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.

Then we use the solution of DICE-CJL from our GAMS code to verify the accuracy of the optimal path computed from the numerical DP algorithm. Table 1 lists the relative errors of the optimal paths over the first 400 years. The errors are computed in the following formula:

$$\max_{th \leq 400} \left| \frac{x^*_{t,\mathrm{DP}} - x^*_{t,\mathrm{GAMS}}}{x^*_{t,\mathrm{GAMS}}} \right|,$$

where $x^*_{t,\mathrm{DP}}$ is the optimal path at stage t from our numerical DP algorithm, and $x^*_{t,\mathrm{GAMS}}$ is the optimal solution at stage t of our GAMS code for DICE-CJL with time interval h.

Table 1: Relative Errors of Optimal Paths from Numerical DP Algorithm

h	k	M^{AT}	T^{AT}	c	μ
10 years	9.1(−4)	9.7(−5)	1.0(−4)	1.6(−4)	1.7(−4)
8 years	9.9(−4)	9.3(−5)	9.7(−5)	1.8(−4)	2.1(−4)
4 years	1.1(−3)	1.1(−4)	1.2(−4)	2.5(−4)	1.5(−4)
2 years	1.3(−3)	1.2(−4)	1.4(−4)	4.6(−4)	1.9(−4)
1 year	1.4(−3)	1.3(−4)	1.5(−4)	3.8(−4)	8.6(−4)
0.5 year	1.4(−3)	1.4(−4)	1.6(−4)	4.1(−4)	1.8(−4)
0.25 year	1.5(−3)	1.4(−4)	1.6(−4)	4.2(−4)	3.3(−4)

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

Moreover, the relative errors of the other states including M^{UP}, M^{LO} and T^{LO}, are also small, varying from $O(10^{-4})$ to $O(10^{-6})$.

Table 2 lists the computational time of our numerical DP algorithm for DICE-CJL with various time intervals. They are run on a Mac with a processor of 2.8 GHz Intel Core 2 Duo and 8GB memory.

Table 2: Computational Time of Numerical DP Algorithm

degree of polynomial	time interval h	computational time
4	10 years	1.9 minutes
6	10 years	38 minutes
4	8 years	2.4 minutes
4	4 years	4.5 minutes
4	2 years	7.8 minutes
4	1 year	15.4 minutes
4	0.5 year	32 minutes
4	0.25 year	59.7 minutes

The main reason that the computational time of numerical DP with the degree-6 complete Chebyshev polynomial is much more than the computational time of numerical DP with the degree-4 one is that we need to compute values at $7^6 = 117649$ points for degree-6 complete Chebyshev polynomial approximation instead of $5^6 = 15625$ points for the degree-4 one. The second reason is that the degree-6 complete polynomial has $\binom{6+6}{6} = 924$ terms, while the degree-4 complete polynomial has only $\binom{4+6}{6} = 210$ terms, so that in the objective function of the maximization step of Algorithm 1, the computational time of next-stage value function approximation in degree-6 complete polynomial takes about 4 times more time than the degree-4 one.

The numerical DP algorithm with the degree-6 complete polynomial is much more time-consuming than the one with the degree-4 one to solve DICE-CJL. Furthermore, the degree-4 one has enough accuracy. Therefore, we will keep using the degree-4 complete Chebyshev polynomial approximation in the numerical DP algorithm in the later stochastic examples.

11 DSICE Results

In this section we apply numerical DP algorithms in our Fortran code to solve DSICE. For each discrete state value, we chose the degree-4 expanded complete Chebyshev polynomials to approximate the value functions. Moreover, we computed the values of the value function on the multidimensional tensor grids with 5 expanded Chebyshev nodes on each dimension in the continuous state ranges, and then compute the Chebyshev coefficients.

After computing these Chebyshev coefficients on all discrete state values for all stages along the 600 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 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 in the GAMS code and $(\zeta_0, J_0) = (1, 1)$, and run it until the terminal time.

In the following examples, we will compute 1000 optimal paths by simulation method, and then plot their distribution.

11.1 DSICE with an Economic Shock

In this example, we consider economic shock ζ_t in the model. Let ζ_t be a discrete Markov chain with 3 possible values of $0.96, 1.0, 1.04$, and its probability transition matrix is

$$\begin{bmatrix} 0.5 & 0.5 & 0 \\ 0.125 & 0.75 & 0.125 \\ 0 & 0.5 & 0.5 \end{bmatrix},$$

where its (i, j) element is the annual transition probability of ζ_t from state i to j.

The following figures show the numerical results of DSICE with a quarter of one year as the time interval of each period over 600 years. It takes about 3.1 hours to run the numerical DP algorithm on a Mac with a processor of 2.8 GHz Intel Core 2 Duo and 8GB memory.

Figure 9 shows the distribution of optimal paths of capital over the first 200 years. The solid line is the average optimal capital along time t, the dotted lines are the minimal or maximal optimal capital, the dashed line is the median optimal capital, the dash-dot lines are the 25% or 75% quantile of the 1000 optimal paths. Figures 10, 11 and 12 plot the distribution of optimal paths of carbon concentration in atmosphere, surface temperature and emission control rate respectively.

We see that the economic shock has significant impact on the capital stock and the emission control rate, but little impact on carbon concentration and temperature. This happens because the economic shock is not irreversible, in this example. No matter which state ζ_t is now, after about 10 years, it has about 1/6 probability to be in state 1, about 2/3 probability to be in state 2, and about 1/6 probability to be in state 3. Moreover, when we have more money from the economic shock with $\zeta_t > 1$, we will spend more money to reduce CO_2 emission, and when we have less money with $\zeta_t < 1$, we will spend less money in reducing CO_2 emission.

Figure 9: Capital in Quarterly Version of DSICE with Economic Shock

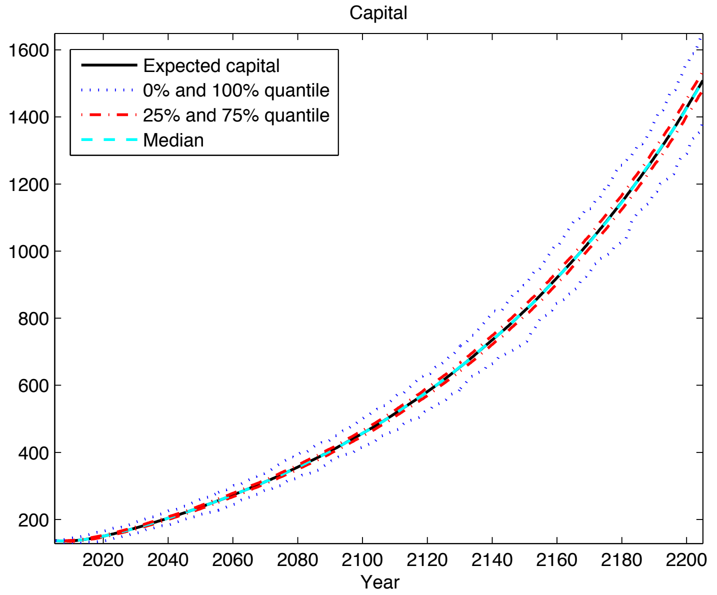


Figure 10: Carbon in Atmosphere in Quarterly Version of DSICE with Economic Shock

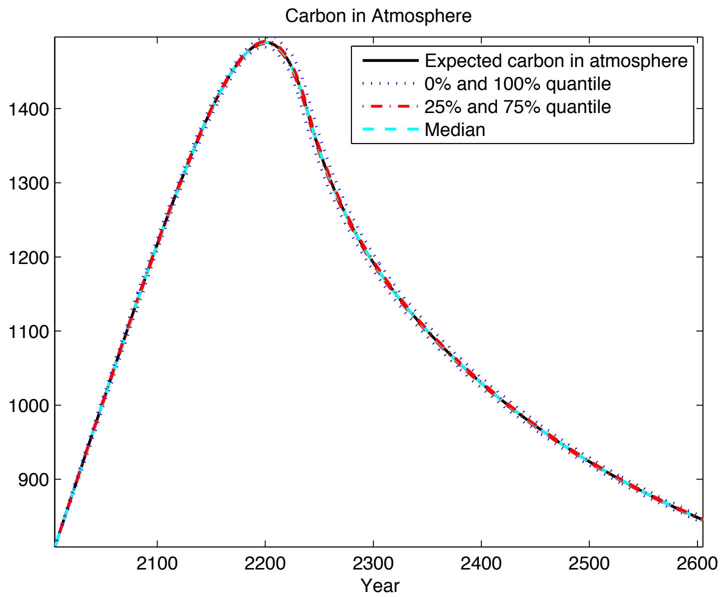


Figure 11: Temperature in Atmosphere in Quarterly Version of DSICE with Economic Shock

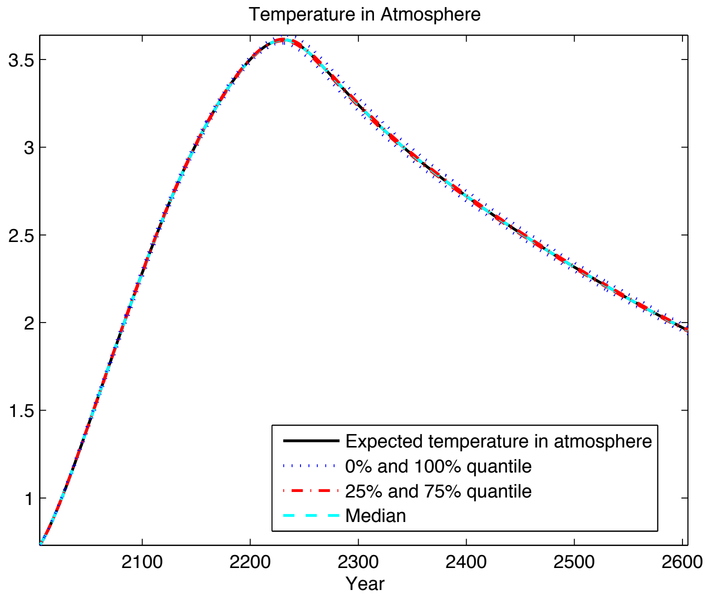
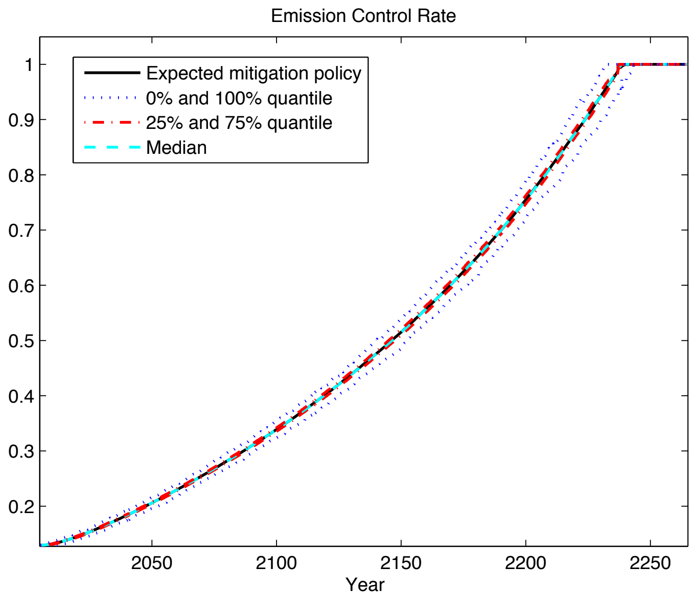


Figure 12: Emission Control Rate in Quarterly Version of DSICE with Economic Shock



11.2 DSICE with a Tipping Point

In this example, we consider a tipping point shock J_t in DSICE. Let J_t be a discrete Markov chain with 2 possible values of $1.0, 0.9$, and its probability transition matrix at time t is

$$\begin{bmatrix} 1-p_{1,2} & p_{1,2} \\ 0 & 1 \end{bmatrix},$$

where its (i,j) element is the transition probability from state i to j for J_t, and

$$p_{1,2} = \max\left\{0, \min\left\{1, \ \frac{h(T_t^{\mathrm{AT}}-1)}{100}\right\}\right\}.$$

So the probability $p_{1,2}$ is dependent on the surface temperature at time t, higher surface temperature implies higher probability to have an irreversible damage, as the transition probability of J_t from state 2 to 1 is 0. See Lontzek, Cai and Judd (2012) for more detailed discussion about DSICE with tipping points.

The following figures show numerical results of DSICE with one year as the time interval of each period over 600 years. It takes about half an hour to run the numerical DP algorithm on a Mac with a processor of 2.8 GHz Intel Core 2 Duo and 8GB memory.

Figure 13 shows the distribution of optimal paths of capital over the first 200 years. The solid line is the average optimal capital along time t, the dotted lines are the minimal or maximal optimal capital, the dashed line is the median optimal capital, the dash-dot lines are the 25% or 75% quantile of the 1000 optimal paths. Figures 14, 15 and 16 plot the distribution of optimal paths of carbon concentration in atmosphere, surface temperature and emission control rate respectively.

From these figures, we see that among these 1000 optimal paths, the tipping point has significant impact on all of these state variables and control variables. Once the tipping event happens, carbon concentration and temperature increase dramatically, and emission control rate has a big jump, and capital will stop growing in the first years and then start growing at a much smaller speed.

Figure 13: Capital in Annual Version of DSICE with Tipping Point

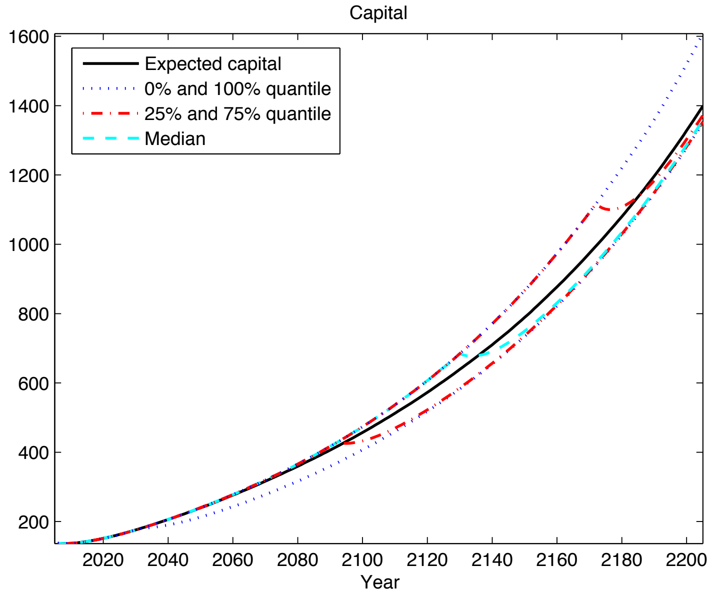


Figure 14: Carbon in Atmosphere in Annual Version of DSICE with Tipping Point

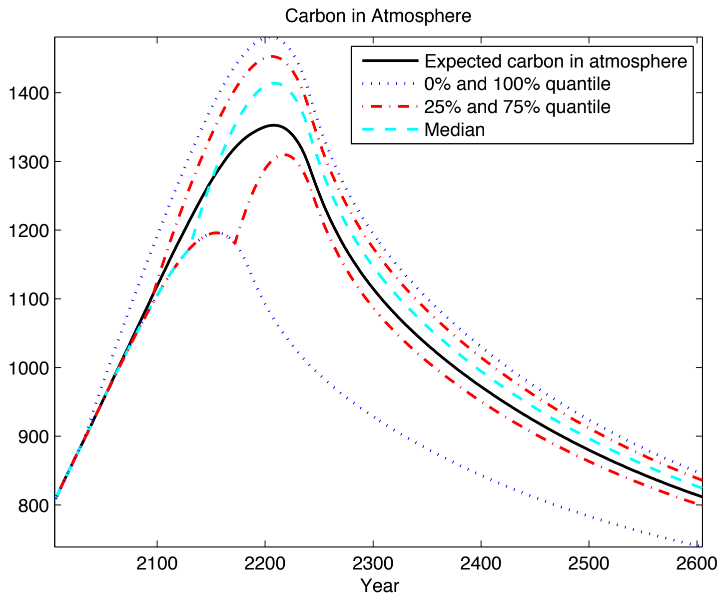


Figure 15: Temperature in Atmosphere in Annual Version of DSICE with Tipping Point

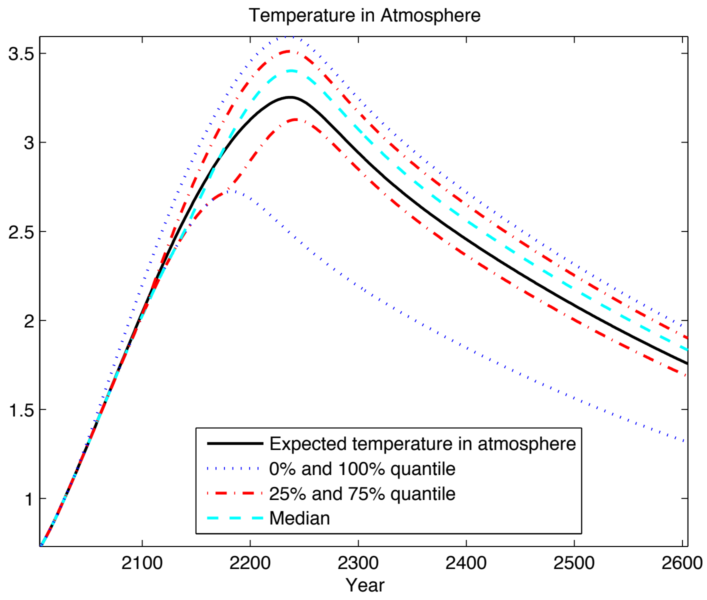


Figure 16: Emission Control Rate in Annual Version of DSICE with Tipping Point

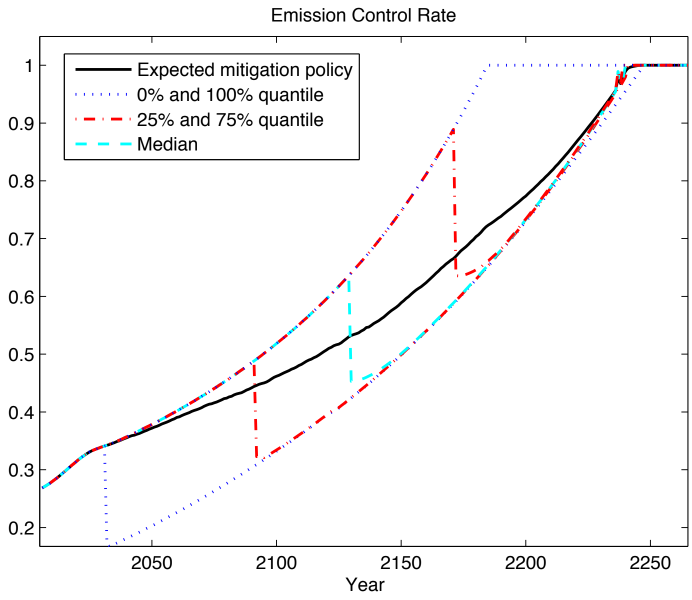
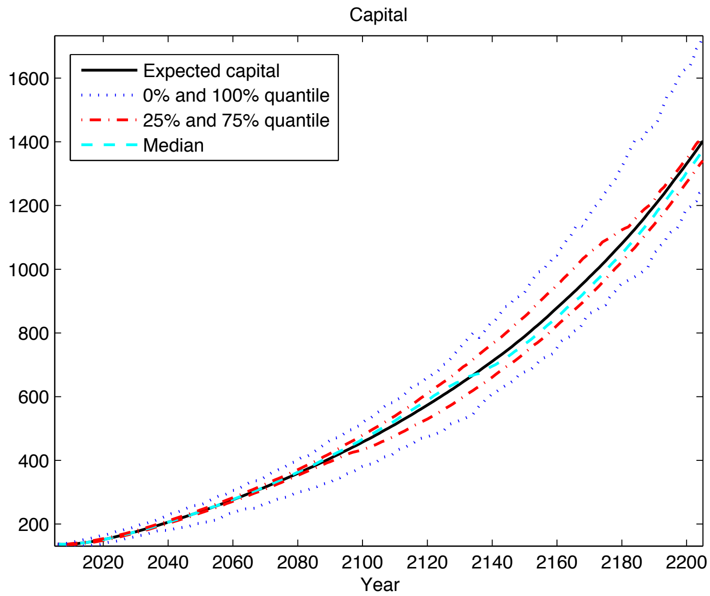


Figure 17: Capital in DSICE with Economic Shock and Tipping Point



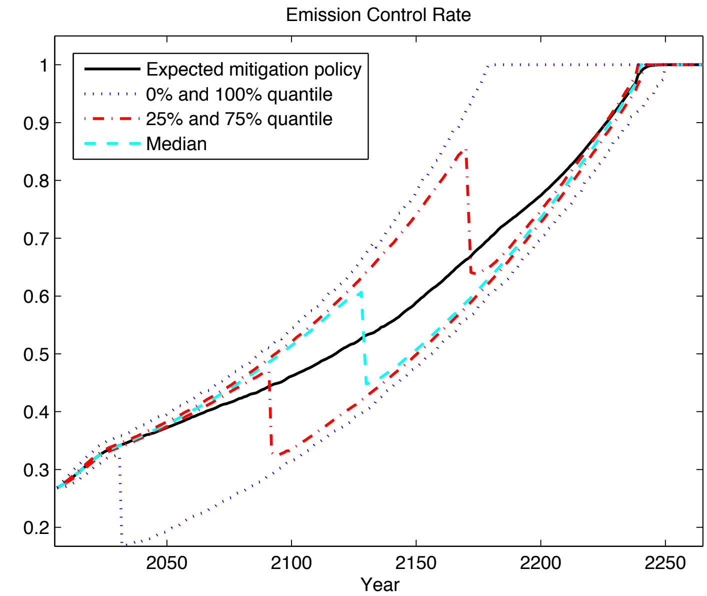
11.3 DSICE with Economic Shock and Tipping Point

Our last example considers the combination of economic shock and tipping point shock in the above two examples. The following figures show numerical results of DSICE with one year as the time interval of each period over 600 years. It takes about 1.5 hours to run the numerical DP algorithm on a Mac with a processor of 2.8 GHz Intel Core 2 Duo and 8GB memory.

Figure 17 shows the distribution of optimal paths of capital over the first 200 years. The solid line is the average optimal capital along time t, the dotted lines are the minimal or maximal optimal capital, the dashed line is the median optimal capital, the dash-dot lines are the 25% or 75% quantile of the 1000 optimal paths. Figure 18 plots the distribution of optimal paths of emission control rate. The distribution of carbon concentration in the atmosphere and the surface temperature are similar to the previous example of DSICE with an tipping point.

From the figures, we see that a tipping point has a significant impact on the state variables and control variables, and the economic shock also impacts significantly capital and emission control rate. Particularly for the emission control rate (Figure 18), we see that the dotted line with maximal emission control rate is not at the same path with the dash-dot or dashed lines before the tipping event happens, but they are the same in the previous example. This difference is caused by the impact of economic shock.

Figure 18: Emission Control Rate in DSICE with Economic Shock and Tipping Point



12 Conclusion

We have described DSICE, a basic dynamic stochastic extension of DICE2007. We have shown that it is quite feasible to combine annual (even sub-annual) time periods with economic and climate uncertainty. The speed of the algorithms means that we can do extensive exploration of parameter space to determine sensitivity of conclusions to parameters about which we have limited information. The accuracy tests indicate that the algorithms are reliable as well as fast. This paper has focused on describing the basic model and addressing basic issues relating to the feasibility of such a model. We have clearly refuted the pessimism one often hears about the possibility of such analyses.

References

[1] Bellman, Richard (1957). *Dynamic Programming.* Princeton University Press.

[2] Cai, Yongyang (2009). *Dynamic Programming and Its Application in Economics and Finance*. PhD thesis, Stanford University.

[3] Cai, Yongyang, and Kenneth Judd (2010). Stable and efficient computational methods for dynamic programming. *Journal of the European Economic Association*, Vol. 8, No. 2-3, 626–634.

[4] Cai, Yongyang, and Kenneth Judd (2012a). Dynamic programming with Hermite interpolation. Working paper.

[5] Cai, Yongyang, and Kenneth Judd (2012b). Dynamic programming with shape-preserving rational spline Hermite interpolation. Working paper.

[6] Cai, Yongyang, and Kenneth Judd (2012c). Shape-preserving dynamic programming. Working paper.

[7] Gill, Philip, Walter Murray, Michael Saunders, and Margaret Wright (1994). User's Guide for NPSOL 5.0: a Fortran Package for Nonlinear Programming. Technical report, SOL, Stanford University.

[8] Judd, Kenneth (1998). *Numerical Methods in Economics.* The MIT Press.

[9] Lontzek, Thomas, Yongyang Cai and Kenneth Judd (2012). Tipping points in a dynamic stochastic IAM. Working paper.

[10] McCarl, B., et al., 2011. McCarl GAMS User Guide. GAMS Development Corporation.

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

[12] Richardson, L. F., and Gaunt, J. A. (1927). The deferred approach to the limit. Philosophical Transactions of the Royal Society of London, Series A 226 (636-646): 299–349.

[13] Rust, J., 2008. Dynamic Programming. In: Durlauf, S.N., Blume L.E. (Eds.), New Palgrave Dictionary of Economics. Palgrave Macmillan, second edition.

About RDCEP

The Center brings together experts in economics, physical sciences, energy technologies, law, computational mathematics, statistics, and computer science to undertake a series of tightly connected research programs aimed at improving the computational models needed to evaluate climate and energy policies, and to make robust decisions based on outcomes.

RDCEP is funded by a grant from the National Science Foundation (NSF) through the Decision Making Under Uncertainty (DMUU) program.

For more information please contact us at
info-RDCEP@ci.uchicago.edu
or visit our website:
www.rdcep.org

RDCEP
Computation Institute
University of Chicago
5735 S. Ellis Ave.
Chicago, IL, 60637 USA
+1 (773) 834 1726