Computational Models for Economics on Blue Waters

Kenneth Judd[1], Hoover Institution

May 7, 2014

[1]Senior representative for Blue Waters GLCPC project team with PI Lars Hansen

Introduction

Economics is a complex system.
Economics research ignores this

- Economists analyze simple stylized models of pieces of the system
- Pencil and paper preferred to computers and code

We are trying to change that

- Create robust and general tools that can use state-of-the art numerical methods on modern computer architectures
- Climate change policy is the application

Climate Change Policy Analysis

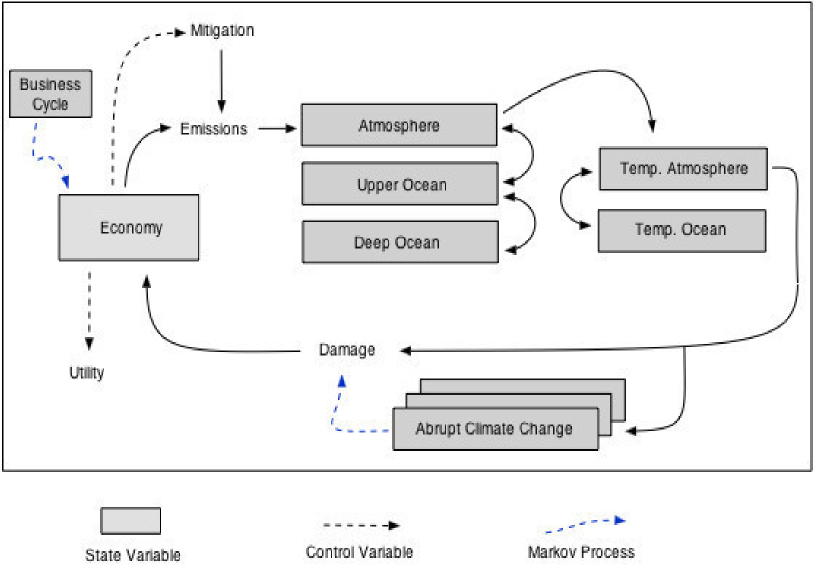
Question: What can and should be the response to rising CO2 concentrations?

- Analytical tools in the literature: IAMs (Integrated Assessment Models)
  - Two components: economic model and climate model
  - Interaction is often limited: Economy emits CO2 which affects world average temperature which affects economic productivity.
- Existing IAMs cannot study dynamic decision-making in an evolving and uncertain world
  - Most are deterministic where economic actors know perfectly future economic and climate events.
  - Limitations are due to economists' aversion to modern computational tools

Nordhaus' DICE: The Prototypical Model

- DICE2007 was the only dynamic economic model used by the US Interagency Working Group on the Cost of Carbon
- Economic system
  - gross output: $Y_t \equiv f(k_t, t) = A_t k_t^{\alpha} l_t^{1-\alpha}$
  - damage factor: $\Omega_t \equiv \left(1 + \pi_1 T_t^{\mathrm{AT}} + \pi_2 (T_t^{\mathrm{AT}})^2\right)^{-1}$
  - emission control cost: $\Lambda_t \equiv \psi_t^{1-\theta_2} \theta_{1,t} \mu_t^{\theta_2}$, where μ_t is policy choice
  - output net of damages and emission control: $\Omega_t(1 - \Lambda_t) Y_t$
- Climate system
  - Carbon mass: $\mathbf{M}_t = (M_t^{\mathrm{AT}}, M_t^{\mathrm{UP}}, M_t^{\mathrm{LO}})^{\top}$
  - Temperature: $\mathbf{T}_t = (T_t^{\mathrm{AT}}, T_t^{\mathrm{LO}})^{\top}$
  - Carbon emission: $E_t = \sigma_t(1 - \mu_t) Y_t + E_t^{\mathrm{Land}}$
  - Radiative forcing: $F_t = \eta \log_2 \left(\left(M_t^{\mathrm{AT}} + M_{t+1}^{\mathrm{AT}} \right) / \left(2 M_0^{\mathrm{AT}} \right) \right) + F_t^{\mathrm{EX}}$

Figure : DSICE Framework



Uncertainty and Risk

All agree that uncertainty needs to be a central part of any IAM analysis
Multiple forms of uncertainty

- Risk: productivity shocks, taste shocks, uncertain technological advances, weather shocks
- Parameter uncertainty: policymakers do not know parameters that characterize the economic and/or climate systems
- Model uncertainty: policymakers do not know the proper model or the stochastic processes

Abrupt, Stochastic, and Irreversible Climate Change

Question: What is the optimal carbon tax when faced with abrupt and irreversible climate change?

- Common assumption in IAMs: damages depend only on contemporaneous temperature
- Our criticism: this cannot analyze the permanent and irreversible damages from tipping points
- We show that
  - Abrupt climate change can be modeled stochastically
  - The policy response to the threat of tipping points is very different from the policy response to standard damage representations.

Cai-Judd-Lontzek DSICE Model

DSICE (**D**ynamic **S**tochastic **I**ntegrated Model of **C**limate and **E**conomy)

$$
\begin{aligned}
DSICE &= DICE2007 \\
&+ \text{uncertainty regarding the future climate} \\
&+ \text{stochastic economic system} \\
&+ \text{parameter uncertainty and learning} \\
&+ \text{flexible period length} \\
&+ \textit{advanced computational methods}
\end{aligned}
$$

DSICE: new features

- Economic system: $Y_t \equiv f(k_t, \zeta_t, t) = \zeta_t A_t k_t^{\alpha} l_t^{1-\alpha}$ where the productivity state $\zeta_{t+1} = g^{\zeta}(\zeta_t, \chi, \omega_t^{\zeta})$ is a stochastic process dependent on a long run risk process χ_t
- Climate system: $\Omega_t \equiv J_t \left(1 + \pi_1 T_t^{AT} + \pi_2 (T_t^{AT})^2\right)^{-1}$ where $J_{t+1} = g^J(J_t, \omega_t^J)$ is a Markov process for the damage factor state J

DSICE with Epstein-Zin Preferences

- Epstein-Zin Preferences: recursive utility function
  - ψ: IES: "dynamic consumption flexibility"
  - γ: risk aversion
- State: $Z = (k, \mathbf{M}, \mathbf{T}, \zeta, \chi, J)$
- Bellman equation ($V_{600}(Z)$ is fixed, and is the terminal condition)

$$
\begin{aligned}
V_t(Z) = \max_{c,\mu} \quad & u(c_t, l_t) + \beta \left[\mathbb{E}_t \left\{ \left(V_{t+1}\left(Z^+\right)\right)^{\frac{1-\gamma}{1-\frac{1}{\psi}}} \right\} \right]^{\frac{1-\frac{1}{\psi}}{1-\gamma}}, \\
\text{s.t.} \quad & k^+ = (1-\delta)k_t + \mathcal{Y}_t(k, T^{AT}, \mu, \zeta, J) - c_t, \\
& \mathbf{M}^+ = \Phi^{\mathrm{M}}\mathbf{M} + \left(\mathcal{E}_t\left(k, \mu, \zeta\right), 0, 0\right)^\top, \\
& \mathbf{T}^+ = \Phi^{\mathrm{T}}\mathbf{T} + \left(\xi_1 \mathcal{F}_t\left(M^{AT}\right), 0\right)^\top, \\
& \zeta^+ = g_\zeta(\zeta, \chi, \omega_\zeta), \\
& \chi^+ = g_\chi(\chi, \omega_\chi), \\
& J^+ = g_J(J, \mathbf{T}, \omega_J), \qquad (1)
\end{aligned}
$$

Numerical Dynamic Programming

- Initialization. Choose the approximation grid, $X = \{x_i : 1 \le i \le m\}$, and choose functional form for $\hat{V}(x; b)$. Let $\hat{V}(x; b^T) = V_T(x)$. Iterate through steps1 and 2 over $t = T-1, ..., 1, 0$.
- Step 1. Maximization step: Compute

$$v_i = \max_{a_i \in \mathcal{D}(x_i, t)} u_t(x_i, a_i) + \beta \mathbb{E}\{\hat{V}(x_i^+; \mathbf{b}^{t+1})\},$$

  for each $x_i \in X$, $1 \le i \le m$.
- Step 2. Fitting step: Using the appropriate approximation method, compute the b^t such that $\hat{V}(x; b^t)$ approximates (x_i, v_i) data.

Parallelization in Dynamic Programming

- Parallelization in maximization step: Compute

$$v_i = \max_{a_i \in \mathcal{D}(x_i,t)} u_t(x_i,a_i) + \beta \mathbb{E}\{\hat{V}(x_i^+;\mathbf{b}^{t+1})\},$$

  for each $x_i \in X_t$, $1 \le i \le m_t$.
- Master-Worker system: Master processor, Worker processors.
- Master-Submaster-Worker system
  - use the Cartesian virtual topology for communicator
  - one submaster corresponds to one discrete state value and its group of workers solve all optimization problems on the continuous state approximation nodes for the discrete state value
  - submasters reduce the data-intensive communication between the master and the workers

Parallelization of DSICE

- Discretized states (ζ, χ, J): $91 \times 19 \times 16 = 27,664$ points
- Six-dimensional continuous states $(k, \mathbf{M}, \mathbf{T})$: 56K approximation nodes per discrete point
- Total number of optimization problems: 372 billion
- Use Master-Submaster-Worker system for one VFI:
  - One submaster task solve Bellman at one discrete point
  - Its workers solve optimization problem for each of the 56K nodes in six continuous dimensions
  - Communication loads
    - Master receives 58MB totally from submasters
    - Each submaster receives 0.45MB from its group of workers
    - Master would juggle 12GB under simple MW

Num of Nodes	Wall Clock Time	Total CPU Time
2,612	8.1 hours	77 years

Parallelization of Uncertainty Quantification in DSICE

- Six uncertain parameter values
  - intertemporal elasticity of substitution
  - risk aversion
  - hazard rate of tipping
  - expected damages
  - variance of damages
  - expected duration of the tipping process
- Solve on Smolyak grid in parameter space
  - construct high-degree approximation to response surface on the parameters: 401 cases
  - Used 3618 cores

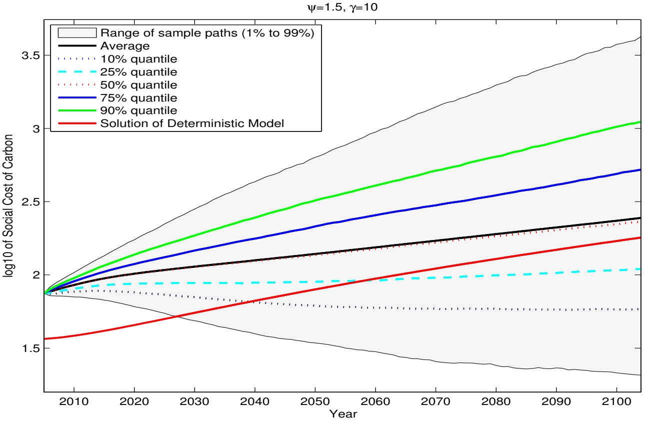
Num of Nodes	Wall Clock Time	Total CPU Time
113	15.5 minutes	39 days

SCC for DSICE with Tipping

Table : Initial Time SCC (US$/tC) for DSICE with Tipping

Hazard Rate Parameter	Mean Damage Level	Expected Duration	Relative Variance	SCC			
				$\psi = 0.5$		$\psi = 1.5$	
				$\gamma = 2$	$\gamma = 10$	$\gamma = 2$	$\gamma = 10$
0.0045	10%	5	0%	88	109	386	480
			40%	91	140	400	586
		200	0%	54.4	57.8	227	259
			40%	54.9	61.5	232	306
0.0025	10%	5	0%	67	83	274	364
			40%	69	103	285	467
		200	0%	47.2	49.6	174	195
			40%	47.5	51.9	176	224

Dynamics of Solutions for DSICE



- Uncertain Growth with Long Run Risk
- Tipping: hazard rate parameter = 0.0035; mean damage level = 5%; expected duration = 50 years; relative variance = 20%

Validation of Models

Economists uses econometric methods for relating data and models
Our view: validation is an enormous inverse optimal problem
Our approach: use parallelism on Blue Waters to solve these problems
Current project is first to construct necessary code and implement far more flexible methods

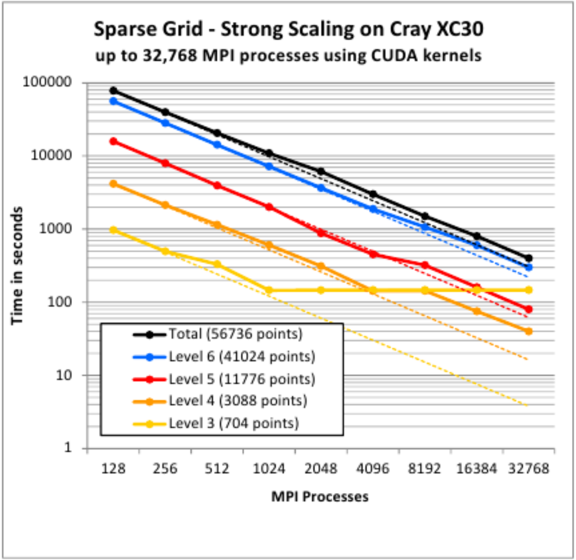
Parallel Iterations of General Operators in Economics Models

More generally, economic problems can be modeled as solutions to operator equations on Banach spaces

- $V(., t) = \mathfrak{F}_t V(., t + 1)$ where $V(Z, t)$ represents economic system at time t at state Z
- Approximate functions $V(., t) = \mathfrak{F}_t V(., t + 1)$ in appropriate Banach space
- Approximate $\mathfrak{F}_t$ operator

Example of Scalability

....



Social Decisions are Dynamic Games

Economic policy analysis focuses mainly on what is "optimal".
Real decisions are made by people acting within a social system with procedures and rules: a game
Most social systems have multiple possible outcomes
We are implementing methods that determine all possible outcomes.

Example of Parallelization of Dynamic Games

- Dynamic Cournot Game with Capacity, 3 players, n states per player, 3 outer iterations
- Number of optimization problems in each iteration: N

Num of States (n^3)	N (billion)	Num of Nodes	Wall Clock Time (minutes)	Total Effective CPU Time (days)	Total Time (days)
27	2.9	312	7.8	31	54
		614	4	31	56
		1228	2.1	31	60
		2456	1.1	31	88
64	16	312	47.3	179	328
		614	24	179	333
		1228	10.4	167	291
		2456	5.3	167	301
		4912	2.8	168	389

Conclusions

Economic analysis of policies for facing risky and uncertain futures require the same scale of computational power as is used for other complex systems.

Economic problems are different from physics and engineering problems

- Different math
  - Unknown functions are relatively smooth, leading to global spectral methods
  - Unknown functions have high dimension
- Different combination of tasks
  - Parallelism breaks big problems into smaller, compute-intensive nonlinear problems
  - Economics applications use less communication relative to compute effort
  - Economics applications can use asynchronous parallelization

Economics is the same in that math and computation can help answer questions.