O' Curse of Dimensionality, Where is Thy Sting?

Kenneth L. Judd
Hoover Institution and NBER
April 11, 2008

Curse of Dimensionality

- Many economic models are high dimensional
  - Dynamic Optimization: Multiple kinds of capital stocks
  - DSGE: Multiple consumers/firms/countries
  - Games: Multiple players and states
  - Bayesian analyses compute high-dimensional integrals
- Claim: "You can't solve your model because of the curse of dimensionality."
  - Response I: Analyze silly models
    - Reduce heterogeneity in tastes, abilities, age, etc.
    - Assume no risk
    - Assume common information, common beliefs, etc.
  - Response II: Do bad math
  - Response III: Apply bad math to silly models

- The message today: "The curse is not so bad"
  - "Theorems" about the curse are irrelevant for economics
  - There are many underutilized tools from math that can help
  - Sensible modelling choices can avoid curse
  - Mathematicians are currently developing tools to tackle the curse
  - Physicists are working to build computers that will avoid the curse
  - If the Boston Red Sox can beat the "Curse of the Bambino" then economists can beat the "Curse of Dimensionality"
- Economics presents unique computational challenges due to the desire to look at high dimensional models.
- This continues in the tradition of Hoover Institution Senior Fellows working on computational methods for high-dimensional problems; see Metropolis-Rosenbluth-Rosenbluth-Teller-Teller (1953) for an earlier example.

Specific Responses to the Challenges of Dimensionality

- Math Tools
  - Compute derivatives efficiently
  - Approximate functions efficiently
  - Choose an efficient domain
  - Approximate integrals efficiently
  - Use functional analysis
- Modelling Suggestions
  - Use continuous time
  - Get rid of kinks
  - Use finite-dimensional models

- Look to the future
  - Use "experimental mathematics" - Monte Carlo people do it all the time, why not the rest of us?
  - Use high-power and high-throughput computing tools - supercomputers, distributed computing, and grid computing
  - Study Griebel-Wozniakowski Theorem
  - Be prepared for quantum computing

Dynamic Example - Dynamic Programming

- Basic Bellman equation:

$$V(x) = \max_{u \in D(x)} \pi(u, x) + \beta \, E\{V(x^+)|x, u)\} \equiv (TV)(x). \tag{12.7.1}$$

- Computational task:
  - Choose a finite-dimensional parameterization (e.g., polynomials, splines, etc.)

$$V(x) \doteq \hat{V}(x; a), \; a \in R^m \tag{12.7.2}$$

    and a finite number of states

$$X = \{x_1, x_2, \cdots, x_n\}, \tag{12.7.3}$$

  - Objective: find coefficients $a \in R^m$ such that $\hat{V}(x; a)$ "approximately" satisfies the Bellman equation.

- Value function iteration: For each x_j, $(TV)(x_j)$ is defined by

$$v_j = (TV)(x_j) = \max_{u \in D(x_j)} \pi(u, x_j) + \beta \int \hat{V}(x^+; a) dF(x^+ | x_j, u) \qquad (12.7.5)$$

- In practice, we compute an approximation to T
  - Integration step: use some numerical quadrature formula

$$E\{V(x^+; a) | x_j, u)\} = \int \hat{V}(g(x_j, u, \varepsilon); a) dF(\varepsilon) \doteq \sum_{\ell} \omega_\ell \hat{V}(g(x_j, u, \varepsilon_\ell); a)$$

  - Maximization step: for $x_i \in X$, evaluate

$$v_i = (T\hat{V})(x_i)$$

  - After finding the new v_j, execute a fitting step:
    - Data: $(v_i, x_i),\ i = 1, \cdots, n$
    - Objective: find an $a \in R^m$ such that $\hat{V}(x; a)$ best fits the data
  - Value function iteration iterates on the coefficient vector a.

- Dimension is important at many stages
  - Optimizing over $u \in R^m$ implies m^2 elements in Hessian
  - In simple methods, the cost of approximating the n - D value function $V(x)$ and choosing points x_j is $\sim k^n$ - *curse of dimensionality*!
  - In simple methods, integrating conditional expectation with q - D shocks has cost $\sim k^q$ - *curse of dimensionality*!

AVAILABLE MATH TOOLS

FOR MULTIDIMENSIONAL PROBLEMS

- Most economists use methods motivated by one-dimensional methods when they solve multidimensional models; the result is inefficient
  - Product rule for integration
  - Cartesian grids for discretizing multidimensional state spaces
- There are many powerful tools that are designed for multidimensional problems

Math Tool I: Evaluate Derivatives Efficiently

- Derivatives are important in perturbation methods and in any method that uses nonlinear equation solvers, but common belief is that these methods, such as Newton's method, are not practical due tothe cost of computing derivatives.
- The facts:
  - Analytic derivatives are slow; for example

Analytic Derivatives		+,-	*,÷	Power	Total flops	Total time
function	$u = (x^\sigma + y^\sigma + z^\sigma)^\rho$	2	0	4	6	22
gradient	$u_x = \sigma\rho x^{\sigma-1}(x^\sigma + y^\sigma + z^\sigma)^{\rho-1}$	4	3	5		32
	$u_y = \sigma\rho y^{\sigma-1}(x^\sigma + y^\sigma + z^\sigma)^{\rho-1}$	4	3	5		32
	$u_z = \sigma\rho z^{\sigma-1}(x^\sigma + y^\sigma + z^\sigma)^{\rho-1}$	4	3	5		32
total:		12	9	15	36	96
Hessian						$\geq$400

– Finite differences are slow; for example

	Finite Difference Derivatives	+,-	*,÷	Power	Total flops	Total time
function	$u = (x^\sigma + y^\sigma + z^\sigma)^\rho$	2	0	4	6	22
gradient	$u_x = (u(x+\Delta, y, z) - u)/\Delta$	3	1	4		24
	$u_y = (u(x, y+\Delta, z) - u)/\Delta$	3	1	4		24
	$u_z = (u(x, y, z+\Delta) - u)/\Delta$	3	1	4		24
grad total:		9	3	12	24	72
Hessian						$\geq$150

- Efficient Differentiation: Derivatives can be computed cheaply

		+,-	*,÷	a^b	Total flops	Clock time
function	$x1 = x^\sigma,\ y1 = y^\sigma,\ z1 = z^\sigma$		0	3	3	15
	$A = x1 + y1 + z1$	2			2	2
	$u = A^\rho$			1	1	5
		2	0	4	6	22
gradient	$x2 = x1/x,\ y2 = y1/y,\ z2 = z1/z,$		3		3	3
	$A1 = \rho\ \sigma\ u/A$		3		3	3
	$(u_x, u_y, u_z) = A1\,(x2, y2, z2)\ \ A1$		3		3	3
grad. cost					9	9
					15	31

- Automatic Differentiation
  - Take a computer program, and write another computer program that computes gradients, Jacobians, Hessians, 3-tensors of third derivatives, etc.
  - Implementation: ADIC, ADIFOR, and others
- Two kinds of gains
  - Fewer operations: Theorem: (Griewank) For an n-dimensional function f:
    - Cost (Jacobian) < 5 Cost (f) (usually less)
    - Cost (Hessian) $< 5\ n$ Cost (f) (usually less)
  - Less use of expensive operations: power (~10 adds), exponential (~5 adds), log (~10 adds), etc.
- Insights are old
  - Many, including Leigh Tesfatsion, recognized these ideas by mid 1980's.
  - Software development was slow. Tesfatsion was an early contributor.

- Current applications and implications
  - Use Newton method (and discard DFP, BFGS, and BHHH)
    - AD is in software; AMPL and GAMS, Gauss
    - L-B-J: already exploits sparseness, could exploit AD
    - Example of application: Solve stochastic dynamic games
      - Pakes-Maguire and others use Gauss–Seidel methods - sssslooooowwwww
      - Ferris-Judd-Schmedders using GAMS/PATH: 40,000 states, 240,000 unknowns; done in 5 minutes on a laptop
  - Perturbation methods
    - Judd, Guu, Gaspar, Anderson, Juillard, Collard, Kim-Kim, Jesus Fernandez-Villaverde, Juan Rubio.
    - Some use AD ideas into their code: Anderson-Levin-Swanson
- Future applications
  - Automatic comparative statics
  - Implement Implicit Function Theorem: implicitly differentiate $F(x, y(x)) = 0$ to get power series for $y(x)$
  - Implement asymptotic methods: Weierstrass preparation, Lyapunov-Schmidt reduction, Laplace expansions etc.

Math Tool II: Efficient Function Approximation

- Linear polynomial methods:

$$f(x, y, z, ...) = \sum_{i=1}^{m} a_i \phi_i (x, y, z, ...), \ \phi_i \text{ multivariate polynomials}$$

  - Simple tensor product approach produces approximations like

$$\sum_{i=0}^{m} \sum_{j=0}^{m} \sum_{k=0}^{m} a_{ij} x^i y^j z^k$$

  - Proper notion of "degree" in multivariate context is sum of powers

$$\text{degree}\left(x^i y^j z^k\right) = i + j + k$$

  - Complete polynomials like

$$\sum_{i+j+k \leq m} a_{ijk} x^i y^j z^k$$

    have far fewer terms by a ratio of nearly $d!$, but are almost as good

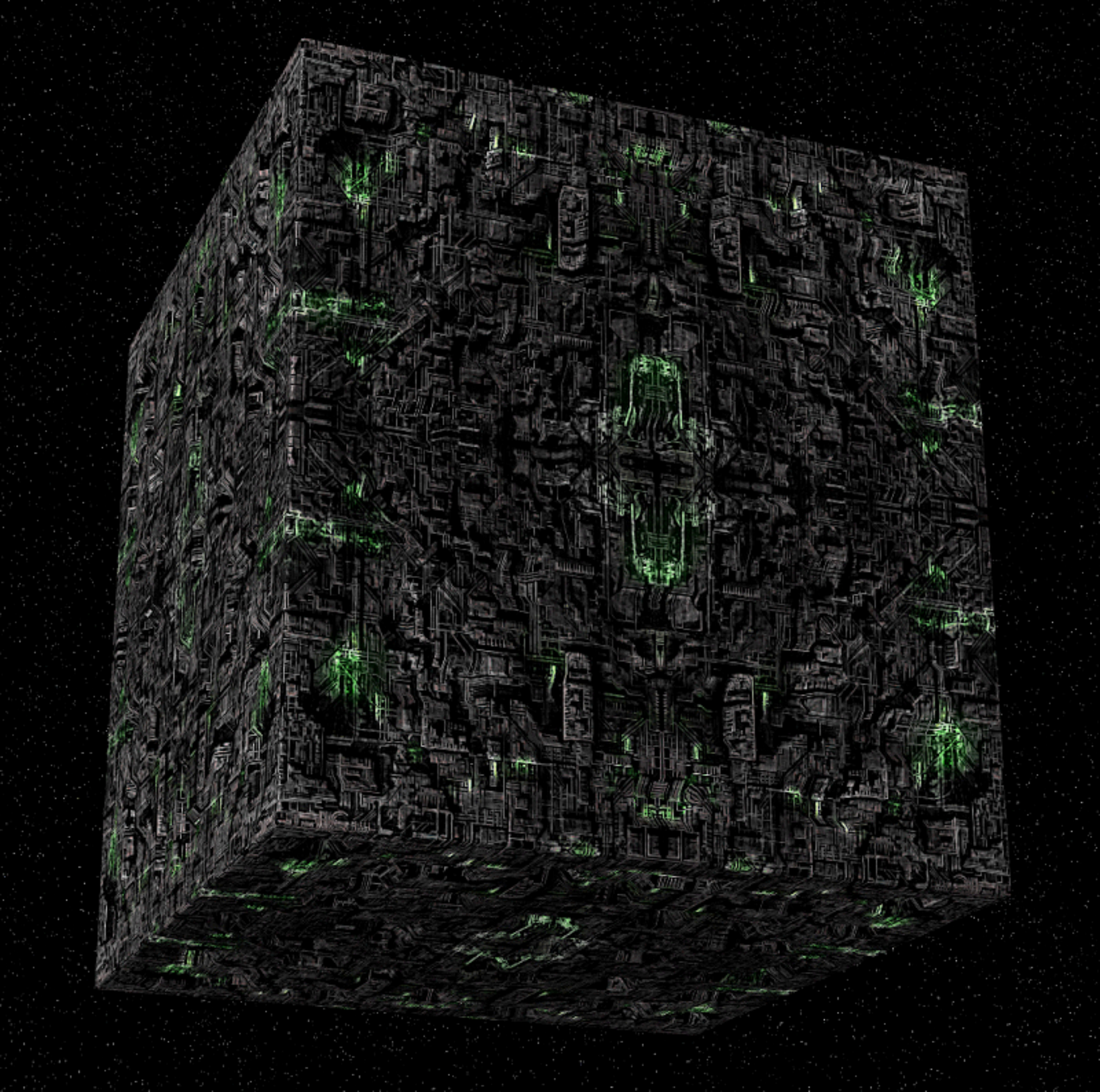
  - See Gaspar-Judd (1997)

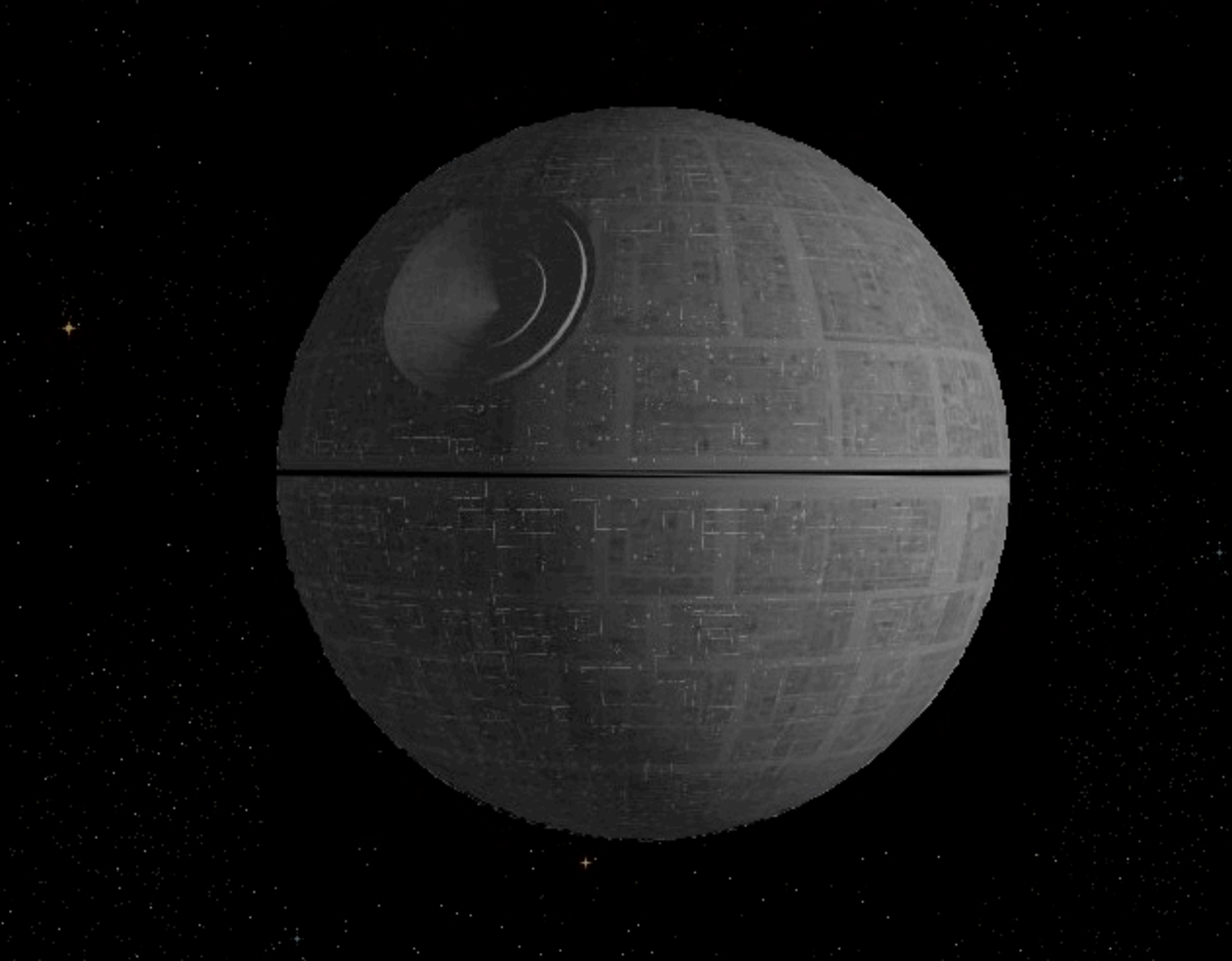
- Splines: build approximations from functions with support
  - One dimension is easy
$$\sum_{i=0}^{m} a_i B_i(x)$$
  - Tensor approach is bad;
$$\sum_{i=0}^{m}\sum_{j=0}^{m} a_{ij} B_i(x) B_j(y)$$
  - Radial basis functions to the rescue:
    * Functional form uses scattered centers p_i in
$$\sum_{i=1}^{N} a_i \phi(\|x - p_i\|)$$
    * ϕ choices include
$$e^{-r^2}, \frac{1}{\sqrt{1+r^2}}, \frac{1}{1+r^2}, \sqrt{1+r^2}, ...$$
    * RBFs can be excellent approximations. Need to figure out best choices for p_i points.
    * RBFs can be very effective on PDEs similar to ones from economics.

- Smolyak points and sparse grids
  - Efficient way to approximate smooth high dimensional functions
  - Krueger-Kubler found them to be very effective in stochastic OLG
  - Mertens used them to solve five-D option pricing problem
  - Judd and Mertens are applying them to Bayesian econometrics

Math Tool III: Define the Domain Efficiently

- Choosing the domain of our problem (e.g., states in a DP or dynamic GE model) is important
  - Want to include values for state that are part of the solution
  - Choosing too large a domain will create unnecessary computational burdens.
- More choices with higher dimensions
  - One dimension: Domain is interval; just need to know max and min
  - Two dimensions: More choices - square/rectangle, sphere/ellipse, simplex, etc.
  - Three dimensions: More choices - cube, sphere, ellipsoid, cylinder, simplex, etc.
- Judd (1992), Gaspar-Judd (1997) made mechanical choice of hypercubes.

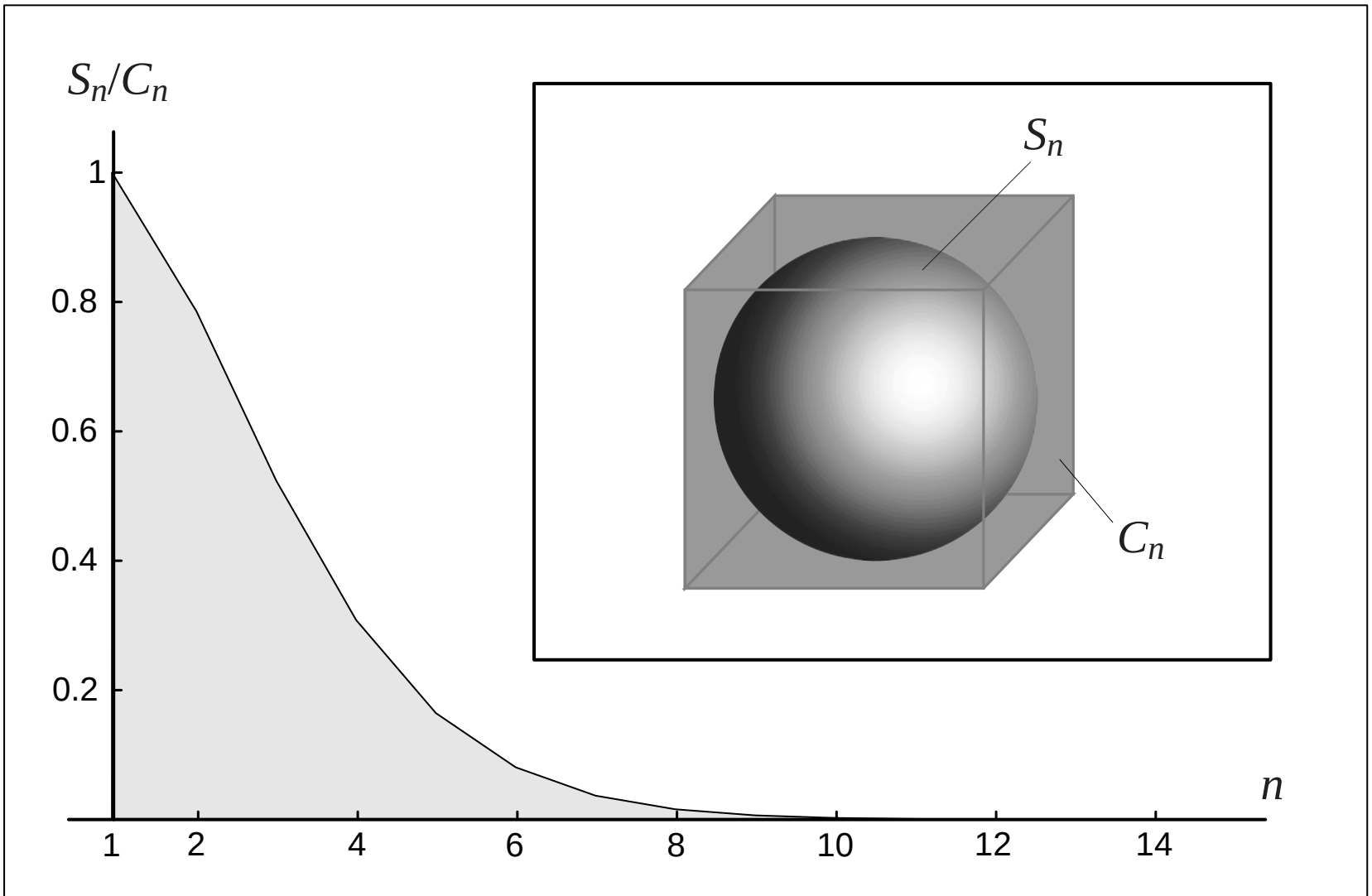


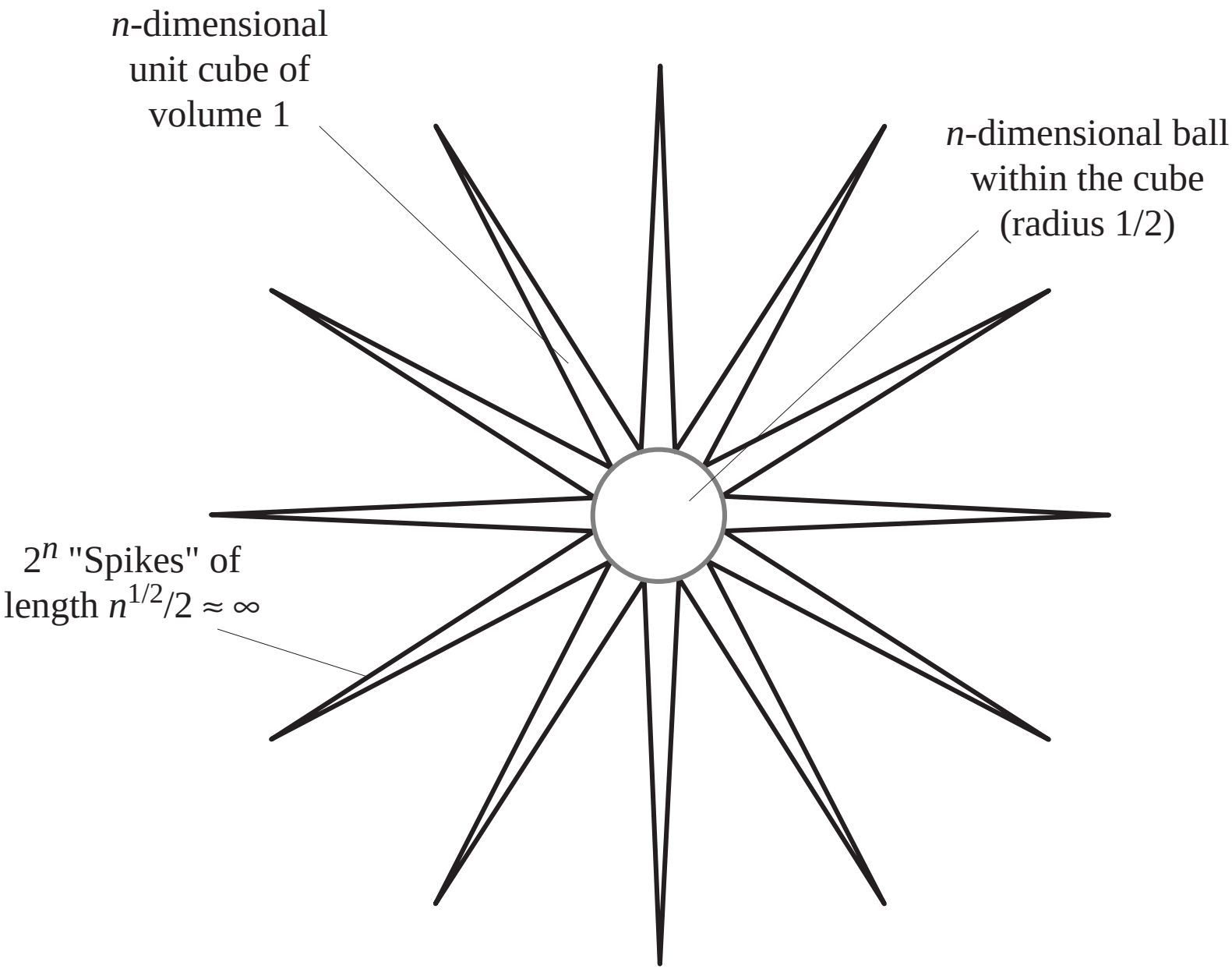


- Cube versus Sphere
  - Spheres are much more compact:
    - In cube of unit length at edge, length of longest diagonal is $n^{1/2}$
    - Ratio of sphere to cube volume is

$$\begin{cases} \frac{\pi^{n/2}}{(n/2)!} & \text{, n even} \\ \frac{2^{n/2+1}\pi^{n/2}}{1 \cdot 3 \cdot 5 \cdot \ldots \cdot n} & \text{, n odd} \end{cases}$$

    - Smaller volume reduces costs of approximation; allows one to exploit periodicity
    - Smaller volume reduces cost of integration
  - If solution has a central tendency, then it rarely visits vertices
  - Mathematicians are developing methods for spheres: orthogonal polynomials for hyperspherical coordinates, quadrature rules for spheres


S_n/C_n
1
0.8
0.6
0.4
0.2
1
2
4
6
8
10
12
14
n
S_n
C_n


n-dimensional unit cube of volume 1
n-dimensional ball within the cube (radius 1/2)
2^n "Spikes" of length $n^{1/2}/2 \approx \infty$

Math Tool IV: Use Efficient Integration Methods
Engineer Gaussian-Style Formulas

- Integration formulas for one dimension and for weighting function $W(x)$ are

$$\int_{-1}^{1} f(x)\ W(x)\ dx \doteq \sum_{i=1}^{n} \omega_i f(z_i)$$

  - n point formula, $2n$ parameters (points and weights)
  - Uses n evaluations of f
  - Exactly integrates all polynomials of degree $2n-1$.

- Simple approach for higher dimensions:

  - Take product of one-dimensional methods:

$$\int_{-1}^{1} f(z)\ W_1(z_1)..W_d(z_d)\ dx \doteq \sum_{i_1=1}^{m} \cdots \sum_{i_d=1}^{m} \omega_{i_1}^1 \omega_{i_2}^2 \cdots \omega_{i_d}^d\ f(z_{i_1}^1, z_{i_2}^2, \cdots, z_{i_d}^d)$$

  - Curse of dimensionality - number of points used is exponential in dimension d

- There are other approaches
  - Do not have to use simple Cartesian grids
  - Consider $X = \{(x, y, z) \,|x, y, z \in \{-1, 1\}\}$. The rule

    $$\frac{4}{3}\left(f(1,0,0) + f(-1,0,0) + f(0,1,0) + f(0,-1,0) + f(0,0,1) + f(0,0,-1)\right)$$

    uses 6 points and exactly integrates all degree 3 polynomials

    $$\left\{1, x, y, z, x^2, y^2, z^2, xyz, xy^2, x^2y, xz^2, x^2z, yz^2, y^2z\right\}$$

    over $[-1, 1]^3$
  - More generally, in dimension d you can use $2d$ points and exactly integrate all degree 3 polynomials over $[-1, 1]^d$ with

    $$\int_{[-1,1]^d} f \doteq \omega \sum_{i=1}^{d} \left(f(ue^i) + f(-ue^i)\right),$$

    where e^i is ± 1 in dimension i and

    $$u = \left(\frac{d}{3}\right)^{1/2}, \ \omega = \frac{2^{d-1}}{d}$$

  - In general, there are nongrid sets of points that can be used - monomial rules.

- New research direction I: Find rules that are good for many polynomials
  - Choose points z_i and weights ω_i, $i = 1, .., m$, to create a quadrature rule,

$$Q\left(f; z, \omega\right) = \sum_{i=1}^{n} \omega_i f(z_i)$$

    to minimize errors over a large set of f functions, not just low order polynomials
    * The literature is for one-dimensional problems:

$$\min_{z,\omega} \sum_{i=0}^{\infty} \left(Q\left(x^i; z, \omega\right) - \int x^i \, dx \right)^2$$

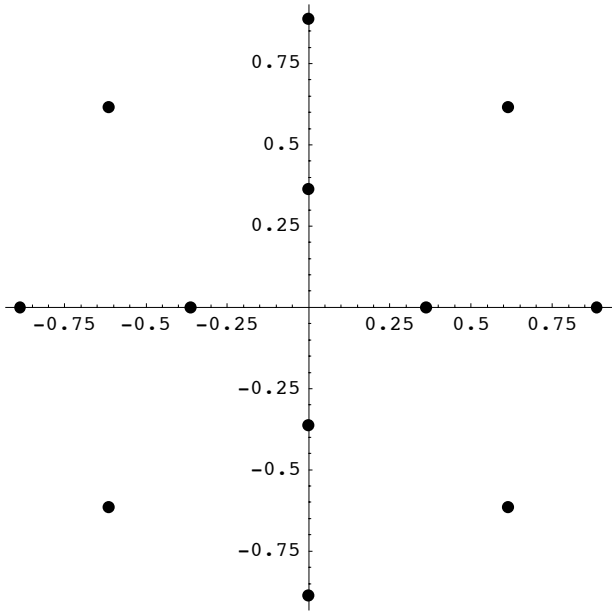
    * A few mathematicians do this - Gismalla, Cohen, Minka
    * This is not done often since "you can't publish the results".

– I created one for a two-D disk:

* We need new formulas if we switch to spheres
* Choose 12 points (24 coordinates and 12 weights) to minimize sum of squared errors of formula applied to $x^i y^j$, $i, j \leq 20$.
* Use unconstrained optimization software; use many restarts to avoid local solution
* Result was

$$\begin{aligned} & 0.2227\ (f[-0.8871, 0] + f[0, -0.8871] + f[0, 0.8871] + f[0.8871, 0]) \\ + & \ 0.2735\ (f[-0.6149, 0.6149] + f[-0.6149, -0.6149] \\ & + f[0.6149, -0.6149] + f[0.6149, 0.61496]) \\ + & \ 1.0744 f[-0.3628, 0] + 1.0744 f[0, 0.3628] \\ & + 1.0744 f[0, -0.3628] + 1.0744 f[0.3628, 0] \end{aligned}$$

  with relativized errors of 10(-5) on average and 10(-4) at worst on degree 20 polynomials
* Result had interesting symmetry - 3 groups of 4 points lying on 3 circles - which gives indication as to what symmetries I should try in higher dimensions.


0.75
0.5
0.25
-0.75
-0.5
-0.25
0.25
0.5
0.75
-0.25
-0.5
-0.75

- General strategy: Look for formulas with small numbers of points to find desirable patterns for point sets, then assume those patterns when searching for bigger formulas.
- General principal: Use your time to come up with ideas, and use the computer to do the tedious work.
  * Idea here: use formulas that integrate a set of polynomials.
  * Tedious work here: searching for optimal rule that satisfies some criterion.

Engineer Formulas That Use More Information

- One-Dimensional Gauss-Turan methods use derivatives

$$\int_{-1}^{1} f(x)\,dx \doteq \sum_{i=1}^{n} \omega_{i,0} f(z_i) + \sum_{i=1}^{n} \omega_{i,1} f'(z_i) + \sum_{i=1}^{n} \omega_{i,2} f''(z_i)$$

  - n-point formula has $4n$ parameters, and uses n evaluations of f, f', and f'' to integrate first $4n$ polynomials
  - In one dimension, the cost of f and first two derivatives is about same as three f's, so no gain in one dimension.

- However, Gauss-Turan has potential for high-dimensional integrals
  - The formula

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)\,dx\;dy = \sum_{i=1}^{n} \omega_{i,0} f(x_i,y_i) + \sum_{i=1}^{n} \left(\omega_{i,x} f_x(x_i,y_i) + \omega_{i,y} f_y(x_i,y_i)\right) + \sum_{i=1}^{n} \left(\omega_{i,xx} f_{xx}(x_i,y_i) + \omega_{i,xy} f_{xy}(x_i,y_i) + \omega_{i,yy} f_{yy}(x_i,y_i)\right)$$

    computes f and five derivatives each point - has $7n$ parameters and can integrate first $7n$ polynomials
  - Using automatic differentiation, multidimensional Gauss-Turan will beat regular quadrature rules that use only f values

Ignore Monte Carlo Propaganda

- If X were distributed uniformly on $[0, 1]$, then

$$\int_0^1 f(x)\,dx = E\{f(X)\} = I_f$$

- Monte Carlo idea: Generate N draws from $U[0, 1]$, $\{x_i\}_{i=1}^N$, and approximate

$$\int_0^1 f(x)\,dx \equiv \frac{1}{N}\sum_{i=1}^N f(x_i) \tag{1}$$

- Monte Carlo Propaganda
  - Best deterministic methods converge at rate $N^{-1/d}$
  - MC converges at rate $N^{-1/2}$ for any dimension d (by Central Limit Theorem)
  - So, MC breaks the curse of dimensionality but deterministic methods cannot.

- Observations about Monte Carlo Propaganda
  - Implementations of MC use pseudorandom sequences *which are deterministic* instead of random numbers
  - Therefore, MC propaganda says that MC won't work if you use standard "random number" generators
  - Implementations of MC converge at rate $N^{-1/2}$ for any dimension d
  - Therefore, there do exist deterministic methods which converge at rate $N^{-1/2}$ for any dimension d.
  - Therefore, MC propaganda is logically inconsistent!

- MC propagandists pull a bait-and-switch
  - They use worst-case analysis (Bakhvalov, 1959) when they say "Best deterministic methods for integrating C^1 functions converge at rate $N^{-1/d}$"
  - They use (the weaker) probability-one criterion when they say "MC methods converge at rate $N^{-1/2}$"
- Mathematical Facts:
  - MC worst-case convergence rate is N^{-0} - *no convergence* - since there always is some sequence where MC does not converge
  - LCM methods converge at $N^{-1/2}$ for smooth functions *in worst case*; proofs (see Neiderreiter) are number-theoretic.
  - Economists do not cite the other *often more relevant* results in Bakhvalov (1959), such as if f is C^k and periodic on the hypercube, there are deterministic rules which converge at rate N^{-k} *independent of dimension*

Math Tool V: Functional Analysis

- Economics problems often reduce to finding unknown functions defined by functional equations
  - Dynamic programming: contraction fixed point map on space of bounded functions using L_∞ norm.
  - Recent work in DP has used Holder spaces - other Banach spaces
  - Dynamic games: must solve equations like

$$0 = G\left(x, S\left(x\right), S\left(S\left(x\right)\right), S'\left(S\left(x\right)\right)\right)$$

- Functional analysis tells us how to generalize calculus (e.g., Taylor series, IFT, etc.) to spaces of functions

- Use IFT in C^k Holder spaces to solve dynamic game
  - Hyperbolic discounting
    - Krusell-Kuruscu-Smith (2002) used high-order conjectural variation approach; produced many solutions
    - Judd (2005) used Banach space IFT to prove local existence and uniqueness, and demonstrated nonlocal validity of expansion
  - Stochastic growth model
    - Perturbation method applied to stochastic growth, as done for example in Judd (1998), is *not* justified by linearization around steady state.
    - Start with deterministic model to get, e.g., $C(k)$ for $k \in [k_0, k_1]$
    - Add uncertainty - σ; compute the function $C_\sigma(k), C_{\sigma\sigma}(k), C_{\sigma\sigma\sigma}(k), C_{\sigma\sigma\sigma\sigma}(k)$, etc., functions for $k \in [k_0, k_1]$
    - Construct series expansion $\sum_{i=0}^{n} \frac{\sigma^i}{i!} C_{\sigma^i}(k)$ that is uniformly valid for $k \in [k_0, k_1]$, *not just around steady state.*
  - General idea: Differentiate in Banach spaces to derive equations that fix $f_\varepsilon(k)$, $f_{\varepsilon\varepsilon}(k)$, $f_{\varepsilon\varepsilon\varepsilon}(k)$, etc., for k on some interval that includes the steady state.Use IFT in C^k Holder spaces to solve dynamic game
- More general results are proved by using hard implicit function theorems.

MODELLING CHOICES THAT

PRODUCE MANAGEABLE MATHEMATICAL PROBLEMS

- Many modelling choices are not essential for the economics.
- Economists often make choices that to make analysis tractable.
- Economists should make choices to reduce computational problems

Modelling Suggestion I: Use Continuous Time

- We pay a high price when we choose discrete-time formulations
- Dynamic programming: Bellman equation
  - Deterministic case for for consumption function
    - Discrete-time:

$$\begin{aligned} k_{t+1} &= F(k_t) - c_t \\ V(k) &= u(C(k)) + \beta V(F(k) - C(k)) \\ u'(C(k)) &= \beta V'(F(k) - C(k)) \\ V'(k) &= \beta V'(F(k) - C(k)) F'(k) \end{aligned}$$

      compositions involving unknowns - $V(F(k) - C(k)))$ and $V'(F(k) - C(k)))$

    - Continuous-time:

$$\begin{aligned} \frac{dk}{dt} &= F(k) - c \\ \rho V(k) &= u(C(k)) + (F(k) - C(k)) V'(k) \\ u'(C(k)) &= \beta V'(k) \\ \rho V'(k) &= F'(k) V'(k) + (F(k) - C(k)) V''(k) \end{aligned}$$

      NO compositions of unknowns! Just multiplications. Problem is linear in coefficients of $V(k)$ if we approximate it by $\Sigma_i a_i \phi_i(k)$ $V(k)$

– Compositions are far more costly than derivatives

* Discrete-Time Bellman series expansion:

$$-V(k) + u(C(k)) + bV(F(k) - C(k))$$

bytes	864	3920	9320	23000	55296	1.4(5)	3.2(5)	7.7(5)	1.8(6)	4.0(6)
time	0.00	0.02	0.02	0.03	0.06	0.08	0.08	0.08	0.95	3.08

* Continuous-Time Bellman series expansion

$$-rV(k) + u(C(k)) + (F(k) - C(k))V'(k)$$

bytes	528	2624	7824	18600	39032	75104	1.5(5)	3(5)	6.1(5)	1.2(6)
time	0.00	0.00	0.00	0.00	0.02	0.02	0.02	0.02	0.03	0.05

– Stochastic case

  * Discrete time: composition plus multidimensional integral

$$E\left\{V\left(F\left(k_t, C\left(k_t\right)\right), \theta_{t+1}\right) | \theta_t\right\}$$

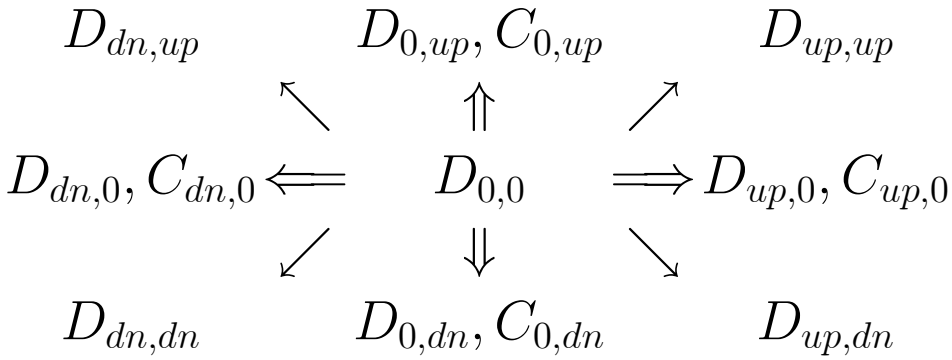
  * Continuous time: $*$, ∇, and Hessian; NO integrals

$$V'\left(k\right)\left(F\left(k\right) - C\left(k\right)\right) + \sigma^2 V''\left(k\right)$$

– Projection methods are much better in continuous-time

  * Equations are purely local
  * Locality increases parallelization possibilities

- Stochastic games:
  - Doraszelski-Judd: continuous-time games are orders of magnitude faster to solve than discrete-time games.
  - If you are at X, your "next" state can be any C state in continuous time, any D state in discrete time.



  - In n dimensions, the difference is 3^n versus $2n$.
    - Pakes-Maguire has curse of dimensionality
    - Doraszelski-Judd does not have curse of dimensionality
    - Why choose discrete-time models? Bad habit? REStud says "Data is in discrete time."

Modelling Suggestion II: Use Finite-But-Large Dimensional States

- Many economists have problems with infinite-dimensional states, such as the distribution of income over a continuum of individuals
- Alternative approach: Assume there is only a finite number of people
- Example: DP problem with N factories
  - k_i - capital stock in factory i
  - $f(k_i)$ - DRTS production function
  - $g^i(k_{i,t+1} - k_{i,t})$ - adjustment costs for investment, $I_{i,t} = k_{i,t+1} - k_{i,t}$.
  - Bellman equation

$$\begin{aligned} V(k) &= \max_I u(c) + \beta V(k + I) \\ c &= \Sigma_i f(k_i) - \Sigma_i g^i\left(I^i(k)\right) \end{aligned}$$

  - Equations defining $V(k)$ and $I(k)$:

$$\begin{aligned} V(k) &= u(c) + \beta V(k + I(k)) \\ 0 &= -u'(c)(1 + \alpha I^i(k)) + V_i(k + I(k)) \end{aligned}$$

  - Idea: Use perturbation method to compute Taylor series for $V(k)$ and the $I^i(k)$

– Problems:

  * V_i is a vector of length N; V_{ij} is a matrix with N^2 elements;
  * $I^i(k)$ is a list of N functions; $I^i_j(k)$ is an $N \times N$ matrix; $I^i_{jm}(k)$ is $(N, N \times N)$ tensor, etc.
  * If $N = 10^9$, that is a lot of unknowns

– Solution: Exploit symmetry

  * $V(x)$ is a polynomial combination of fundamental symmetric polynomials

$$\sum_i x_i, \sum_{i \neq j} x_i x_j, \sum_{i \neq j \neq k \neq i} x_i x_j x_k, \ \ldots$$

  * At steady state,
    · $V_i = V_j, \forall i, j$
    · $V_{ii} = V_{11}, \forall i$; $V_{ij} = V_{12}, \forall i \neq j$
    · $V_{iii} = V_{111}, \forall i$; $V_{iij} = V_{112}$, $V_{ijj} = V_{122}, \forall i \neq j$; $V_{ijm} = V_{123}, \forall i \neq j \neq m \neq i$
  * Similarly for I^i functions

- High-order Taylor series are feasible
  - The number of unknowns when computing q'th derivative is $2q$ independent for any N.
  - Solutions depend on N; take $N \to \infty$ to find infinite population solution
  - Risk - idiosyncratic and aggregate - can be added with little extra computational cost.
- Similar to Gaspar-Judd (1997) use of symmetry, but far more efficient
- Approximate decision rules in terms of moments
  - Like Krusell-Smith (1997), but K-S makes ad hoc choice of moments based on unreliable accuracy tests.
  - This is asymptotically valid and choice of moments is determined by math.

Modelling Suggestion III: Get Rid of Kinks

- General observation: the more smoothness, the easier the computation.
- Economists love to put kinks and discontinuities in their models
  - These formulations make distinctions very clear
  - Kinks and discontinuities create many computational problems particularly when you go to multiple dimensions
  - However, real world is never so clear.
  - Adding real-world fuzziness will make computing easier.

- Example: Hubbard and Judd (1986)
  - Wanted to examine tax policy implications of borrowing constraints.
    - Assumed one could not borrow against future wages; equivalent to

$$r(W) = \begin{cases} r, & W > 0 \\ \infty, & W < 0 \end{cases} \quad \text{or} \quad u(c, W) = \begin{cases} u(c), & W > 0 \\ -\infty, & W < 0 \end{cases}$$

    - Results were interesting; for example, positive taxation of interest income is desirable.
    - Any multidimensional extension would be hampered by computational difficulties
  - Is this economically reasonable? Debt is not infinitely painful
    - First, go to parents and other family members, and, second, run up credit card debt.
    - In general, there is a set of sources of credit, with rising interest rates
    - Empirical fact: people do have debt!
  - Smooth $r(W)$ is more realistic and more tractable

FUTURE TOOLS

- Economists should not just think in terms of the hardware, algorithms, and software available today.
- Some new tools will be particularly valuable for solving high dimensional models.

Methodology: Pure Math vs. Experimental Math

- Mathematical proofs are nice but not necessary.
- Example: MC methods as practiced in economics
  - Monte Carlo is very useful and sound but **NOT** supported by probability theorems.
  - Real proof is
    - Suppose $f(x) = \sum_{i=0}^{\infty} a_i x^i$ on $[0,1]$ and $\sum_{i=K}^{\infty} a_i x^i$ is negligible for some K
    - Suppose computations show that a proposed pMC sequence X_i properly computes $\int x^i dx$ at rate $N^{-1/2}$ for each $i < K$.
    - Then, this pMC sequence will compute $\int f(x)$ accurately at rate $N^{-1/2}$
- This is experimental mathematics, **NOT** probability theory!
- Experimental math:
  - Test out conjecture on many cases to explore validity
  - Combine computational results with pure math to arrive at conclusions with known range of validity
  - Computational results may inspire theorems, such as Neiderreiter analysis of LCM.

- Problem is not with using MC, but with understanding logical underpinnings.
  - MC in practice is not based on probability theory
  - It is *inspired* by probability theory, but theorems do not apply
  - This inspiration led to searches for pseudo-MC sequences which, by *testing*, were found to do a good job on *some* problems
- Why are these logical points important?
  - All agree that Monte Carlo is a very important and useful tool.
  - Recognition of the true foundation for MC will encourage us to develop other methods based on a similarly disciplined combination of analysis and computational experimentation.

- Potential applications of experimental math
  - Search for best integration methods for economics problems
    - Minimize error on concave and/or monotone integrands
    - Minimize error for expectations of marginal utility over range of CRRA utility functions
  - Search for best approximation methods for economics problems.
    - Choose interpolation points according to general criterion
    - Choose interpolation methods best for concave/monotone functions
    - Find best centers for radial basis functions

Hardware: Computing Speed

- We need more speed to do the necessary heavy lifting - searches for good methods, symbolic manipulation, experimental mathematics - implicit in the ideas mentioned above.
- Economists do not use existing computing power
  - Massively parallel architectures - 10^4 processors in BlueGene
  - Network computing - Condor, BOINC, Globus, ...
  - Available resources - modern solvers (Filter, CPlex, Knitro, Snopt, Ipopt, NEOS,...), MPI
- Current projects
  - Dynamic programming on MW Condor and supercomputers
  - Dynamic games on DAGMAN Condor

- More speed is coming in the next couple of years
  - Massively parallel architectures - 10^5 processors in next BlueGene
  - Multicore processors, on desktops and inside supercomputer processors.
  - Economics problems are better able to exploit supercomputer power than standard physics problems.
  - Even more opportunity for economists to fall further behind other fields.

Numerical Analysis: Griebel-Wozniakowski Theorem

- Question: Are there good rules out there to defeat the curse of dimensionality?
- Answer: Yes, if we formulate problem in reasonable spaces.
- "On the Optimal Convergence Rate of Universal and Non-Universal Algorithms for Multivariate Integration and Approximation" by Griebel and Wozniakowski
  - Consider functions that belong to reproducing kernel Hilbert spaces (RKHS).
  - Bad news
    - Any algorithm for integration that works for all RKHS (a so-called "universal algorithm") displays a curse of dimensionality.
      - Economists want universal algorithms
      - Economic models often have simple structures, implying particular RKHS

– Good news

  * For any specific RKHS, there is an algorithm with at least $N^{-1/2}$ convergence for multivariate integration (like MC) - no curse of dimensionality!
  * If the kernel is a product of univariate kernels (as is often the case in economics), i.e.,

$$\int_{[0,1]^n} g(x)\, dF(x) = \int_{[0,1]} ... \int_{[0,1]} g(x)\, dF_1(x_1) ... dF_n(x_n)$$

    · Then there is an algorithm that converges at the same rate as the slowest univariate algorithm

    · Many univariate algorithms have exponential convergence

    · Hence, there exist some excellent rules!

– Proof is nonconstructive, but tells us that computer searches will likely succeed.

Quantum Computing

- New technology may break curse of dimensionality
- Quantum computer example
  - Load quantum computer with a function f and a number n.
  - ZAP it and it becomes n computers (more precisely, the quantum state of the computer will be a superposition of the n possible states)
  - ZAP it so that computer i computes $f(i)$, $i = 1, .., n$
  - Take a random draw among the $f(i)$ on the n computers *biased* in favor the larger $f(i)$
    - Get $\max f(i)$ with probability $1 - n^{-1}$
    - Time is $\sim N^{1/2}$!

- Quantum complexity theory
  - Examines possible efficiency of quantum computer algorithms.
  - There are examples of where quantum computing breaks curse of dimensionlity.
  - "Path Integration on a Quantum Computer," Traub and Wozniakowski (2001).
    - Path integration on a quantum computer is tractable - i.e., no curse of dimensionality.
    - Path integration on a quantum computer can be solved
      - roughly $\varepsilon^{-1/2}$ times faster than on a classical computer using randomization, and
      - exponentially faster than on a classical computer with a worst case assurance.
    - The number of quantum queries is the square root of the number of function values needed on a classical computer using randomization.
  - In general, integration is faster on a quantum computer than a classical computer - Brassard-Hoya-Mosca-Tapp.

CONCLUSION

- The curse of dimensionality can often be avoided in economic analysis
  - If you formulate models in the right way
  - If you use best available math
  - If you use modern hardware and software
- New developments are making this easier to do.
- The path is clear, but there is a lot of work to do to build the road.