Numerically Stable and Accurate Stochastic Simulation Approaches for Solving Dynamic Economic Models

Kenneth L. Judd, Lilia Maliar and Serguei Maliar

July 1, 2011

Three broad classes of numerical methods

1. Projection methods, Judd (1992), Christiano and Fisher (2000), etc.
   - solution domain = prespecified grid of points;
   - accurate and fast with few state variables but cost grows exponentially with number of state variables (curse of dimensionality!).

2. Perturbation methods, Judd and Guu (1993), Gaspar and Judd (1997), Juillard (2003), etc.
   - solution domain = one point (steady state);
   - practical in large-scale models but the accuracy can deteriorate dramatically away from the steady state.

3. Stochastic simulation methods, Marcet (1988), Smith (2001), etc.
   - solution domain = simulated series;
   - simple to program but often numerically unstable, and the accuracy is lower than that of the projection methods.

In the paper, we show how to enhance the performance of stochastic simulation methods.

Stochastic simulation methods

- A stochastic simulation method is roughly as follows:

  Step 1. Guess a policy function.

  Step 2. Simulate time series.

  Step 3. Use simulation results to recompute the guess.

  Iterate on *Steps* $2 - 3$ until convergence.

- Step 3 requires
  - to fit a polynomial function to the simulated data (regression);
  - to evaluate conditional expectations (integration).

- We show that both regression and integration have problems:
  - In regression, polynomial terms are highly correlated (multicollinearity), and the standard LS technique fails $\Rightarrow$ numerical instability.
  - Monte Carlo integration is very inaccurate $\Rightarrow$ the overall accuracy of solutions is low.

Our results

- We stabilize the stochastic simulation procedure:
  - we build the regression step on approximation methods designed for dealing with multicollinearity
- We attain high accuracy of solutions:
  - we generalize the stochastic simulation algorithm to include accurate Gauss Hermite quadrature and monomial integration methods
- The generalized stochastic simulation algorithm (GSSA) is
  - numerically stable
  - comparable in accuracy to most accurate methods in the literature
  - tractable in problems with high dimensionality (hundreds of state variables)
  - very simple to program

Representative-agent neoclassical growth model

A planner solves

$$\max_{\{k_{t+1}, c_t\}_{t=0}^{\infty}} E_0 \sum_{t=0}^{\infty} \beta^t u\left(c_t\right)$$

$$\text{s.t.} \quad c_t + k_{t+1} = (1-\delta)\, k_t + a_t f\left(k_t\right),$$

$$\ln a_{t+1} = \rho \ln a_t + \epsilon_{t+1}, \quad \epsilon_{t+1} \sim \mathcal{N}\left(0, \sigma^2\right)$$

where initial condition (k_0, a_0) is given;
$f\left(\cdot\right)$ = production function;
c_t = consumption; k_{t+1} = capital; a_t = productivity;
β = discount factor; δ = depreciation rate of capital;
ρ = autocorrelation coefficient of the productivity level;
σ = standard deviation of the productivity shock.

Key advantage of stochastic simulation methods

- Search a solution only in the areas of the state space that are visited in simulation (ergodic set).
- Recall that projection methods compute solutions in a rectangular domain (and perturbation methods – in one (steady-state) point).

Figure 1. The ergodic set in the model with a closed-form solution.
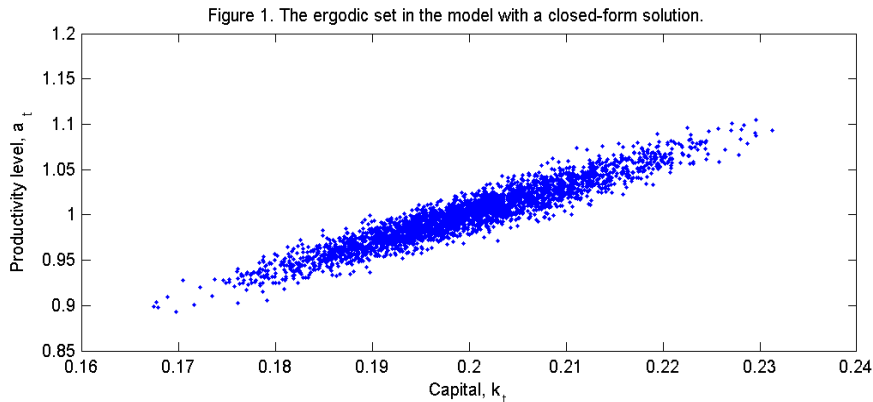
Reduction in cost in a 2-dimensional case

- How much can we save on cost using the ergodic-set domain comparatively to the hypercube domain?
- Suppose the ergodic set is a circle.
- In the 2-dimensional case, a circle inscribed within a square occupies about 79% of the area of the square.
- The reduction in cost is proportional to the shaded area in the figure.
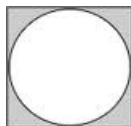
- It does not seem to be a large gain.

Reduction in cost in a p-dimensional case

- In a 3-dimensional case, the gain is larger (a volume of a sphere of diameter 1 is 52% of the volume of a cube of width 1)
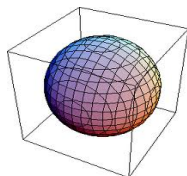
- In a *p*-dimensional case, the ratio of a hypersphere's volume to a hypercube's volume

$$\mathcal{V}^p = \begin{cases} \frac{(\pi/2)^{\frac{p-1}{2}}}{1 \cdot 3 \cdot ... \cdot p} & \text{for } p = 1, 3, 5... \\ \frac{(\pi/2)^{\frac{p}{2}}}{2 \cdot 4 \cdot ... \cdot p} & \text{for } p = 2, 4, 6... \end{cases}.$$

- $\mathcal{V}^p$ declines very rapidly with dimensionality of state space. When $p = 10 \Rightarrow \mathcal{V}^p = 3 \cdot 10^{-3}$ (0.3%). When $p = 30 \Rightarrow \mathcal{V}^p = 2 \cdot 10^{-14}$.
- We face a tiny fraction of cost we would have faced on the hypercube.

Another advantage of focusing on the ergodic set

- Stochastic simulation methods operating on the ergodic set fit a polynomial on the relevant domain.
- Projection methods operating on larger hypercube domains fit a polynomial both inside and outside the relevant domain. Such methods face a trade-off between accuracy inside and outside the ergodic set.
- Hence, for the same degree of approximating polynomial, ergodic set methods are likely to get more accurate solutions in the relevant domain than methods operating on larger prespecified domains.

- The existing conventional stochastic simulation methods did not benefit from their potential advantages. We next explain why...

Starting point: simulation-based PEA of Marcet (1988)

Parameterize the policy function of marginal utility,

$$u'(c_t) = E_t \left\{ \beta u'(c_{t+1}) \left[1 - \delta + a_{t+1} f'(k_{t+1}) \right] \right\} \approx \Psi(k_t, a_t; b),$$

where $\Psi(k_t, a_t; b) = \exp(b_0 + b_1 \ln k_t + b_2 \ln a_t + ...)$ is an exponentiated polynomial. Write the budget constraint as

$$k_{t+1} = (1 - \delta) k_t + a_t f(k_t) - u'^{-1} \left[\Psi(k_t, a_t; b) \right].$$

- Fix $b = (b_0, ..., b_n)$. Given productivity levels $\{a_t\}_{t=0}^{T}$, simulate $\{c_t, k_{t+1}\}_{t=0}^{T}$ and construct

$$y_t \equiv \beta u'(c_{t+1}) \left[1 - \delta + a_{t+1} f'(k_{t+1}) \right],$$

- Run a non-linear LS (NLLS) regression $y_t = \Psi(k_t, a_t; b) + \varepsilon \Rightarrow$ get $\widehat{b}$.
- Compute the next-iteration input $b^{(j+1)}$ using fixed-point iteration

$$b^{(j+1)} = (1 - \xi) b^{(j)} + \xi \widehat{b},$$

where $\xi \in (0, 1] =$ damping parameter.

Numerical problems of the simulation-based PEA

Works well for 1st-degree polynomials but is often unstable for higher degree polynomials.

1. When running a NLLS regression, the computer may fail to deliver an estimator $\widehat{b}$ at all.
2. The NLLS coefficients may change drastically from one iteration to another which causes cycling and leads to non-convergence.
3. Even if convergence is achieved, the resulting approximation is often not accurate.

In practice, numerical problems arise even under 2nd-degree polynomial approximation. For example, Den Haan and Marcet (1990) removed the collinear cross term $\ln k_t \ln a_t$ in the second-degree polynomial,

$$\exp\left(b_0 + b_1 \ln k_t + b_2 \ln a_t + b_3 \ln k_t^2 + b_3 \ln a_t^2\right).$$

What causes numerical problems?

1. *Ill-conditioning of the least-squares (LS) problem* solved in the approximation step,

$$\min_{b} \ \left[y - \Psi\left(k, a; b\right)\right]^{\top} \left[y - \Psi\left(k, a; b\right)\right].$$

It arises due to multicollinearity and poor scaling of explanatory variables.

2. In addition, *exponentiated polynomial approximation* $\Psi\left(k, a; b\right) = \exp\left(b_0 + b_1 \ln k_t + b_2 \ln a_t + ...\right)$, used in Marcet (1988), should be estimated with NLLS methods which

   a) require to supply an initial guess;
   b) involve computing costly Jacobian and Hessian matrices;
   c) often fail to converge.

Ill-conditioned LS problem

- Under *the linear regression model*, $y = Xb + \varepsilon$, we have the OLS estimator

$$\widehat{b} = \left(X^\top X\right)^{-1} X^\top y,$$

where $X \equiv [1_T, x_1, ..., x_n] \in \mathbb{R}^{T \times (n+1)}$.

- The matrix $X^\top X$ is often *ill-conditioned*. The degree of ill-conditioning of $X^\top X$ can be measured in terms of a condition number

$$\mathcal{K}\left(X^\top X\right) \equiv \lambda_1 / \lambda_n$$

$\lambda_1 =$ the largest eigenvalue of $X^\top X$; $\lambda_n =$ its smallest eigenvalue.

- The eigenvalues of $X^\top X$ are defined by

$$X^\top X = V \Lambda V^\top$$

$\Lambda =$ an $n \times n$ diagonal matrix with ordered eigenvalues of $X^\top X$ on its diagonal; $V =$ an $n \times n$ matrix of its eigenvectors.

- $\mathcal{K} \uparrow \implies$ the closer is $X^\top X$ to being singular (not invertible).

Multicollinearity problem

High-degree polynomial terms forming X are significantly correlated.

> ## Example
>
> Let $X = \begin{bmatrix} 1+\phi & 1 \\ 1 & 1+\phi \end{bmatrix}$ with $\phi \neq 0$. Then, $\mathcal{K}\left(X^\top X\right) = \left(1 + \frac{2}{\phi}\right)^2$.
>
> Let $y = (0,0)^\top$. Thus, the OLS solution is $\left(\widehat{b}_1, \widehat{b}_2\right) = (0,0)$. Suppose y is perturbed by a small amount, i.e. $y = (\varepsilon_1, \varepsilon_2)^\top$. Then, the OLS solution is
>
> $$\widehat{b}_1 = \frac{1}{\phi}\left[\frac{\varepsilon_1\left(1+\phi\right) - \varepsilon_2}{2+\phi}\right] \text{ and } \widehat{b}_2 = \frac{1}{\phi}\left[\frac{\varepsilon_2\left(1+\phi\right) - \varepsilon_1}{2+\phi}\right].$$
>
> Sensitivity of $\widehat{b}_1$ and $\widehat{b}_2$ to perturbation in y is proportional to $1/\phi$ (increases with $\mathcal{K}\left(X^\top X\right)$).

Poor scaling problem

Polynomial terms forming X have very different means and variances (due to different scales among either the state variables, k_t and a_t, or the polynomial terms of different orders, like k_t and k_t^5).

Example

Let $X = \begin{bmatrix} 1 & 0 \\ 0 & \phi \end{bmatrix}$ with $\phi \neq 0$. Then, $\mathcal{K}\left(X^\top X\right) = 1/\phi$. Let $y = (0,0)^\top$.

Thus, the OLS solution is $\left(\widehat{b}_1, \widehat{b}_2\right) = (0,0)$. Suppose y is perturbed by a small amount, i.e. $y = (\varepsilon_1, \varepsilon_2)^\top$. The OLS solution is

$$\widehat{b}_1 = \varepsilon_1 \text{ and } \widehat{b}_2 = \frac{\varepsilon_2}{\phi}.$$

Sensitivity of $\widehat{b}_2$ to perturbation in y is proportional to $1/\phi$ (and $\mathcal{K}\left(X^\top X\right)$).

Our first goal is to attain numerical stability

1. We replace the exponentiated polynomial $\Psi(k, a; b) = \exp(b_0 + b_1 \ln k_t + b_2 \ln a_t + ...)$ used in Marcet (1988) with a simple polynomial $\Psi(k, a; b) = b_0 + b_1 \ln k_t + b_2 \ln a_t + ...$ This allows us to replace NLLS methods with linear methods.

2. We use approximation methods that can handling collinear data and dampen movements in b.

   - LS using SVD, Tikhonov regularization;
   - Least absolute deviations (LAD) methods (primal and dual linear programming problems);
   - Principal components (truncated SVD) method.

3. Other factors that can affect numerical stability of GSSA:

   - Data normalization.
   - The choice of a family of basis functions.
   - The choice of policy functions to parameterize.

Normalizing the variables

- *Center* - subtract the sample mean from each observation.
- *Scale* - divide each observation by the sample standard deviation.
- By construction, a centered variable has a zero mean, and a scaled variable has a unit standard deviation.
- After a regression model is estimated, the coefficients in the original (unnormalized) regression model are restored.

LS approaches to the linear regression model

Two LS approaches that are more numerically stable and more suitable for dealing with ill-conditioning than the standard OLS approach.

1. *LS using SVD (LS-SVD)*: uses a singular-value decomposition of X.
2. *Regularized LS using Tikhonov regularization (RLS-Tikhonov)*: relies on a specific (Tikhonov) regularization of the ill-conditioned LS problem that imposes penalties based on the size of the regression coefficients.

The LS-SVD approach finds a solution to the original ill-conditioned LS problem, while the RLS-Tikhonov approach modifies (regularizes) the original ill-conditioned LS problem into a less ill-conditioned problem.

LS-SVD

- SVD of the matrix $X \in \mathbb{R}^{T \times n}$

$$X = USV^{\top}$$

  where $U \in \mathbb{R}^{T \times n}$ and $V \in \mathbb{R}^{n \times n}$ = orthogonal matrices; $S \in \mathbb{R}^{n \times n}$ = diagonal matrix with diagonal entries $s_1 \geq s_2 \geq ... \geq s_n \geq 0$, known as *singular values* of X.

- The OLS estimator $\widehat{b} = \left(X^{\top}X\right)^{-1} X^{\top}y$ in terms of the SVD:

$$\widehat{b} = \left(VS^{\top}SV^{\top}\right)^{-1} VS^{\top}U^{\top}y = VS^{-1}U^{\top}y$$

- If $X^{\top}X$ is well-conditioned $\Longrightarrow$ the OLS formula and the LS-SVD formula give identical estimates of b.

- However, if $X^{\top}X$ is ill-conditioned and the standard OLS estimator cannot be computed $\Longrightarrow$ it is still possible that matrices X and S are sufficiently well-conditioned, $\mathcal{K}\left(S\right) = \sqrt{\mathcal{K}\left(X^{\top}X\right)} \Longrightarrow$ can compute the LS-SVD estimator.

RLS-Tikhonov

- Regularization - process of re-formulating an ill-conditioned problem by imposing additional restrictions on the solution.
- Tikhonov regularization - the most commonly used regularization method in approximation theory.
- Impose an L_2 penalty on the size of the regression coefficients:

$$\min_b \ \|y - Xb\|_2^2 + \eta \|b\|_2^2 = \min_b \ (y - Xb)^\top (y - Xb) + \eta b^\top b$$

  where $\eta \geq 0$ = regularization parameter.
- Find the FOC with respect to b

$$\widehat{b}(\eta) = \left(X^\top X + \eta I_n\right)^{-1} X^\top y$$

  where I_n = an identity matrix of order n.
- *Note:* add a positive constant to $X^\top X$ prior to inverting this matrix.
  $\implies$ Even if $X^\top X$ is singular, the matrix $X^\top X + \eta I_n$ is non-singular.
  $\implies$ Can compute its inverse.

LAD approaches to the linear regression model

- Replace the ill-conditioned LS problem with a least-absolute deviations (LAD) problem

$$\min_b \|y - Xb\|_1 = \min_b 1_T^\top |y - Xb|$$

where $\|\cdot\|_1$ denotes L_1 vector norm.

- The LAD problem does not require computing $\left(X^\top X\right)^{-1}$.
- No explicit solution. However, we can re-formulate the LAD problem to consist of a linear objective function and linear constraints $\Longrightarrow$ Solve with standard linear programming techniques.
- Substitute $|y - X\beta|$ with a vector $w \in \mathbb{R}^T$ to obtain

$$\min_{b,\, w} 1_T^\top w$$

$$\text{s.t.} \ -w \leq y - X\beta \leq w$$

- This problem has $n + T$ unknowns. We argue that it is not the most suitable for a numerical analysis.

LAD: primal problem (LAD-PP)

- Charnes et al. (1955): express the deviation for each observation as a difference between two non-negative variables v_t^+ and v_t^-,

$$y_t - \sum_{i=0}^{n} b_i x_{ti} = v_t^+ - v_t^-, \tag{1}$$

- v_t^+ and v_t^- can be interpreted as non-negative vertical deviations above and below the fitted line, $\widehat{y}_t = X_t \widehat{b}$, respectively; $v_t^+ + v_t^- =$ absolute deviation between the fit $\widehat{y}_t$ and the observation y_t.

- *Primal problem:* minimize the total sum of absolute deviations subject to (1),

$$\min_{v^+, v^-, b} 1_T^\top v^+ + 1_T^\top v^-$$
$$\text{s.t. } v^+ - v^- + Xb = y,$$
$$v^+ \geq 0, \quad v^- \geq 0,$$

where $v_t^+, v_t^- \in \mathbb{R}^T$.

- This formulation is more simple to solve than the direct formulation.

LAD: dual problem (LAD-DP)

- Every primal problem can be converted into a dual problem.
- *Dual problem* corresponding to the primal problem:

$$\max_q y^\top q$$
$$\text{s.t. } X^\top q = 0$$
$$-1_T \leq q \leq 1_T$$

  where $q \in \mathbb{R}^T$ is a vector of unknowns.

- If the number of observations, T, is sizable (i.e. $T \gg n$), the dual problem is less computationally cumbersome than the primal problem.

Regularized LAD (RLAD)

- Modify the original LAD problem to incorporate an L_1 penalty on b.
- *The RLAD problem:*

$$\min_b \ \|y - Xb\|_1 + \eta \, \|b\|_1 = \ \min_b \ 1_T^\top \, |y - Xb| + \eta 1_n^\top \, |b| \,,$$

  where $\eta \geq 0$ = regularization parameter.
- We develop a linear programming formulation of the RLAD problem parallel to the LAD-PP: replace $|b_i|$ with two variables.
- Wang, Gordon and Zhu (2006): represent $|b_i|$ as $sign\,(b_i)\,b_i$.

RLAD: primal problem (RLAD-PP)

- To cast the RLAD problem into a linear programming form, we represent b as $b_i = \varphi_i^+ - \varphi_i^-$, with $\varphi_i^+ \geq 0$, $\varphi_i^- \geq 0$ for $i = 1, ..., n$.
- We then impose a linear penalty on each φ_i^+ and φ_i^-.
- The resulting regularized version of the primal problem:

$$\min_{v^+, v^-, \varphi^+, \varphi^-} 1_T^\top v^+ + 1_T^\top v^- + \eta 1_n^\top \varphi^+ + \eta 1_n^\top \varphi^-$$

$$\text{s.t.} \quad v^+ - v^- + X\varphi^+ - X\varphi^- = y,$$

$$v^+ \geq 0, \quad v^- \geq 0,$$

$$\varphi^+ \geq 0, \quad \varphi^- \geq 0,$$

where φ^+, $\varphi^- \in \mathbb{R}^n$ are vectors that define $b(\eta)$.

- This problem has $2T + 2n$ unknowns, as well as T equality restrictions and $2T + 2n$ lower bounds.

RLAD: dual problem (RLAD-DP)

- The dual problem corresponding to the RLAD-PP:

$$\max_{q} y^\top q$$
$$\text{s.t. } X^\top q \leqslant \eta \cdot 1_n,$$
$$-X^\top q \leqslant \eta \cdot 1_n,$$
$$-1_T \leq q \leq 1_T,$$

where $q \in \mathbb{R}^T$ = vector of unknowns.

- Here, $2n$ linear inequality restrictions and $2T$ lower and upper bounds on T unknown components of q.

Principal component method (Truncated SVD, LS-TSVD)

- $Z \equiv XV$, where $X \in \mathbb{R}^{T \times n}$, $Z \in \mathbb{R}^{T \times n}$ and $V \in \mathbb{R}^{n \times n}$.
- $z_1, ..., z_n$ are called *principal components* of X and are orthogonal, $z_i^{\top} z_i = s_i^2$ and $z_j^{\top} z_i = 0$ for any $j \neq i$, where $s_i = i$th singular value of X.
- *Idea:* reduce ill-conditioning of X to a "desired" level by excluding low-variance principal components corresponding to small singular values.
- Let $\kappa = $ largest condition number of X that we are willing to accept.
- Compute $\frac{s_1}{s_2}, ..., \frac{s_1}{s_n}$, where $s_1 = $ largest singular value.
- $\mathcal{K}(X) = \mathcal{K}(S) = \frac{s_1}{s_n} = $ actual condition number of the matrix X.

Principal component method (Truncated SVD, LS-TSVD)

- Let $Z^r \equiv (z_1, ..., z_r) \in \mathbb{R}^{T \times r}$ be the first r principal components for which $\frac{s_1}{s_i} \leq \kappa$.
- Remove the last $n - r$ principal components for which $\frac{s_1}{s_i} > \kappa$.
- By construction, $\mathcal{K}(Z^r) \leq \kappa$.
- Re-write the linear regression model in terms of Z^r,

$$y = Z^r \vartheta^r + \varepsilon,$$

where $\vartheta^r \in \mathbb{R}^r = $ vector of coefficients.
- Estimate ϑ^r using any of the LS and LAD methods described.
- Find $\widehat{b} = V^r \widehat{\vartheta}^r \in \mathbb{R}^n$, where $V^r = (v_1, ..., v_r) \in \mathbb{R}^{n \times r}$ contains the first r right singular vectors of X.

Choosing policy functions to parameterize

- *Marcet (1988):* parameterize marginal-utility policy function

$$u'(c_t) = E_t \left\{ \beta u'(c_{t+1}) \left[1 - \delta + a_{t+1} f'(k_{t+1}) \right] \right\} \approx \Psi(k_t, a_t; b)$$

- *Our benchmark case:* parameterize capital policy function $k_{t+1} = K(k_t, a_t)$,

$$k_{t+1} = E_t \left\{ \beta \frac{u'(c_{t+1})}{u'(c_t)} \left[1 - \delta + a_{t+1} f'(k_{t+1}) \right] k_{t+1} \right\} \approx \Psi(k_t, a_t; b)$$

Choosing a family of basis functions

- Polynomial families of basis functions.
- Ordinary polynomial family - standard.
- A better alternative is orthogonal polynomial families.
- Ordinary polynomials $O_m(x)$ versus Hermite polynomials $H_m(x)$ up to degree 5:

$$
\begin{array}{ll}
O_0(x) = 1 & H_0(x) = 1 \\
O_1(x) = x & H_1(x) = x \\
O_2(x) = x^2 & H_2(x) = x^2 - 1 \\
O_3(x) = x^3 & H_3(x) = x^3 - 3x \\
O_4(x) = x^4 & H_4(x) = x^4 - 6x^2 + 3 \\
O_5(x) = x^5 & H_5(x) = x^5 - 10x^3 + 15x.
\end{array}
$$

- $O_m(x)$, $m = 1, ..., 5$ appear very similar $\Longrightarrow$ the explanatory variables for the regression are likely to be correlated.
- $H_m(x)$, $m = 1, ..., 5$ are different in the shapes $\Longrightarrow$ the multicollinearity problem manifests to a much lesser degree, if at all.

Choosing a family of basis functions

Figure 2a. Ordinary polynomials.
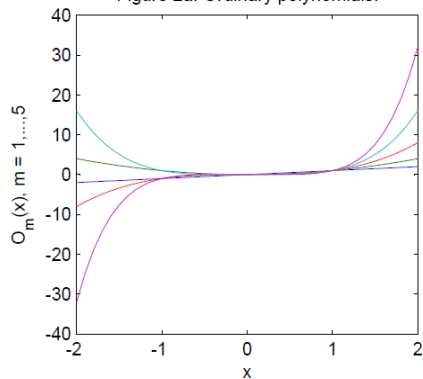
Figure 2b. Hermite polynomials.
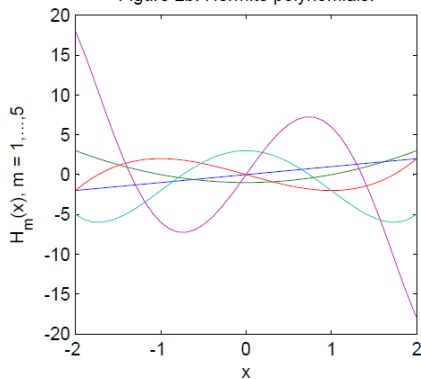
Methodology and parameterization

- Production function: $f(k_t) = k_t^\alpha$ with $\alpha = 0.36$.
- Utility function: $u(c_t) = \frac{c_t^{1-\gamma} - 1}{1-\gamma}$ with $\gamma \in \{0.1, 1, 10\}$.
- Process for shocks: $\rho = 0.95$ and $\sigma = 0.01$.
- Discount factor: $\beta = 0.99$.
- Depreciation rate: $\delta = 1$ and $\delta = 0.02$.
- Under $\gamma = 1$ and $\delta = 1 \implies$ *closed-form solution*.
- Accuracy is measured by an Euler-equation error,

$$\mathcal{E}(k_t, a_t) \equiv E_t \left[\beta \frac{c_{t+1}^{-\gamma}}{c_t^{-\gamma}} \left(1 - \delta + \alpha a_{t+1} k_{t+1}^{\alpha-1} \right) \right] - 1,$$

expressed in $\log 10$ units.

Results for the model with the closed-form solution

Full depreciation of capital, $\delta = 1$.

	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU
Polyn. degree	OLS, Ordinary Unnormalized		OLS, Ordinary Normalized		OLS, Hermite Unnormalized	
1st	-3.52	0.8 sec	-3.52	1 sec	-3.52	1 sec
2nd	-5.46	3.1 sec	-5.46	3 sec	-5.46	4 sec
3rd	-	-	-6.84	5 sec	-6.84	6 sec
4th	-	-	-	-	-7.94	8 sec
5th	-	-	-	-	-9.09	10 sec
	Ordinary, LS-SVD Normalized		Ordinary, LAD-PP Normalized		Ordinary, RLS-Tikh. $\eta = 10^{-7}$	
1st	-3.52	1 sec	-3.52	16 sec	-3.52	1 sec
2nd	-5.46	3 sec	-5.55	1.5 min	-5.46	3 sec
3rd	-6.84	5 sec	-6.97	4.1 min	-5.85	4 sec
4th	-7.94	6 sec	-8.16	6.4 min	-6.12	7 sec
5th	-9.12	10 sec	-9.10	9.3 min	-6.22	11 sec

Results for the model without a closed-form solution

Partial depreciation of capital, $\delta = 0.02$.

	$\mathcal{E}_{mean}$	CPU
Polyn.	MC(1)	
degree	$T = 10,000$	
1st	-4.26	1 sec
2nd	-4.42	11 sec
3rd	-4.32	25 sec
4th	-4.31	47 sec
5th	-4.23	80 sec

We attain stability but now high-degree polynomials do not lead to more accurate solution. Why?

Conventional one-node Monte Carlo integration

Integral $\approx$ the next-period's realization of the integrand

$$\beta E_t \left[u'\left(c_{t+1}\right) \left[1 - \delta + a_{t+1} f'\left(k_{t+1}\right)\right]\right] \approx \beta u'\left(c_{t+1}\right) \left[1 - \delta + a_{t+1} f'\left(k_{t+1}\right)\right]$$

This integration method is used in Marcet's (1988) PEA.

- An integration error is $\varepsilon_t^I \equiv y_t - E_t\left[y_t\right]$.
- The OLS estimator is $\widehat{b} = b + \left[\left(X\right)^\top X\right]^{-1} \left(X\right)^\top \varepsilon^I$.
- Assuming that $\varepsilon_t^I \sim \mathcal{N}\left(0, \sigma_\varepsilon^2\right)$, we have the central limit theorem: the asymptotic distribution of the OLS estimator is
  $\sqrt{T}\widehat{b} \sim \mathcal{N}\left(b, \left[X^\top X\right]^{-1} \sigma_\varepsilon^2\right)$, i.e., the convergence rate is $\sqrt{T}$.
- In RBC models, variables like y_t fluctuate by several percents.
- Assume error $\left|\frac{y_t - E_t[\cdot]}{E_t[\cdot]}\right|$ is on average 10^{-2} (i.e. 1%). Then a regression with $T = 10,000$ has errors of order $10^{-2}/\sqrt{T} = 10^{-4}$.
- To reduce errors to order 10^{-5}, we need $T = 1,000,000$.

$\Rightarrow$ **High accuracy is theoretically possible but impractical.**

Deterministic integration methods

Our GSSA relies on accurate Gauss Hermite quadrature integration

$$\int_{\mathbb{R}^N} g(\varepsilon) w(\varepsilon) d\varepsilon \approx \sum_{j=1}^{J} \omega_j g(\varepsilon_j),$$

where $\{\varepsilon_j\}_{j=1}^{J}$ = integration nodes, $\{\omega_j\}_{j=1}^{J}$ = integration weights.

Example

a) A two-node Gauss-Hermite quadrature method, $Q(2)$, uses nodes $\epsilon_1 = -\sigma$, $\epsilon_2 = \sigma$ and weights $\omega_1 = \omega_2 = \frac{1}{2}$.

b) A three-node Gauss-Hermite quadrature method, $Q(3)$, uses nodes $\epsilon_1 = 0$, $\epsilon_2 = \sigma\sqrt{\frac{3}{2}}$, $\epsilon_3 = -\sigma\sqrt{\frac{3}{2}}$ and weights $\omega_1 = \frac{2\sqrt{\pi}}{3}$, $\omega_2 = \omega_3 = \frac{\sqrt{\pi}}{6}$.

c) A one-node Gauss-Hermite quadrature method, $Q(1)$, uses a zero node, $\epsilon_1 = 0$, and a unit weight, $\omega_1 = 1$.

Quadrature integration in the studied model

For $t = 0, ..., T-1$, we approximation the conditional expectation as

$$y_t = \sum_{j=1}^{J} \left\{ \omega_j \cdot \left(\beta u' \left(c_{t+1,j} \right) \left[1 - \delta + a_{t+1,j} \ f' \left(k_{t+1} \right) \right] \right) \right\},$$

where $c_{t+1,j}$, the value of c_{t+1} if the innovation in productivity is ϵ_j, is defined for $j = 1, ..., J$ by

$$
\begin{aligned}
a_{t+1,j} &\equiv a_t^\rho \exp\left(\epsilon_j \right), \\
c_{t+1,j} &\equiv \Psi \left(k_{t+1}, a_t^\rho \exp\left(\epsilon_j \right) ; b^{(p)} \right).
\end{aligned}
$$

where $\{\epsilon_j\}_{j=1,...,J}$ and $\{\omega_j\}_{j=1,...,J}$ are J integration nodes and weights, respectively.

Results for the model with partial depreciation of capital

	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU
Polyn. degree	MC(1) $T = 10,000$		MC(2000) $T = 10,000$		MC(1) $T = 100,000$	
1st	-4.26	1 sec	-4.40	20.6 min	-4.39	4 sec
2nd	-4.42	11 sec	-6.04	28.5 min	-4.87	1.3 min
3rd	-4.32	25 sec	-6.15	36.6 min	-4.86	3.1 min
4th	-4.31	47 sec	-6.08	55.6 min	-4.72	5.7 min
5th	-4.23	80 sec	-6.07	1.27 h	-4.71	10.4 min
	Q(1) $T = 100$		Q(2) $T = 10,000$		Q(10) $T = 10,000$	
1st	-4.36	3 sec	-4.36	16 sec	-4.36	20 sec
2nd	-6.05	4 sec	-6.13	27 sec	-6.13	34 sec
3rd	-6.32	5 sec	-7.48	35 sec	-7.48	44 sec
4th	-6.24	6 sec	-8.72	44 sec	-8.72	54 sec
5th	-6.04	7 sec	-8.91	51 sec	-8.91	63 sec

RLS-TSVD with $\kappa = 10^7$

Model with rare disasters

We now study the performance of the GSSA under large shocks.

- In addition to standard normally distributed shocks, the productivity level is subject to large negative low-probability shocks (rare disasters).

- We modify the process for productivity as:
  $\ln a_{t+1} = \rho \ln a_t + (\epsilon_{t+1} + \zeta_{t+1})$, where $\epsilon_{t+1} \sim \mathcal{N}\left(0, \sigma^2\right)$, ζ_{t+1} takes values $-\zeta\sigma$ and 0 with probabilities $\overline{p}$ and $1 - \overline{p}$, respectively, and $\zeta > 0$.

- We assume that $\zeta = 10$ and $\overline{p} = 0.02$, i.e. a 10% drop in the productivity level occurs with the probability of 2% (these values are in line with the estimates in Barro, 2009).

Results for the model with rare disasters

	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU	$\mathcal{E}_{mean}$	CPU
Polyn. degree	RLS-Tikhonov $\eta = 10^{-6}$		RLS-TSVD $\kappa = 10^{8}$		RLAD-DP $\eta = 10^{-6}$	
1st	-3.97	50 sec	-3.97	40 sec	-3.98	1.4 min
2nd	-5.47	1.3 min	-5.47	1.1 min	-5.61	2.5 min
3rd	-6.63	1.8 min	-6.64	1.6 min	-6.81	4.3 min
4th	-7.67	2.2 min	-7.67	2 min	-7.88	10.7 min
5th	-8.16	2.6 min	-8.66	2.4 min	-8.86	19.9 min

Ordinary polynomials, $T = 10,000$, $Q(10)$.

Gauss Hermite product rules

In multi-dimensional problem, we can use Gauss Hermite product rules.

Example

Let $\epsilon_{t+1}^h \sim \mathcal{N}\left(0, \sigma^2\right)$, $h = 1$, 2, 3 be uncorrelated random variables. A two-node Gauss-Hermite product rule, $Q\left(2\right)$, (obtained from the two-node Gauss-Hermite rule) has 2^3 nodes, which are as follows:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	$j = 7$	$j = 8$
$\epsilon_{t+1,j}^1$	σ	σ	σ	σ	$-\sigma$	$-\sigma$	$-\sigma$	$-\sigma$
$\epsilon_{t+1,j}^2$	σ	σ	$-\sigma$	$-\sigma$	σ	σ	$-\sigma$	$-\sigma$
$\epsilon_{t+1,j}^3$	σ	$-\sigma$	σ	$-\sigma$	σ	$-\sigma$	σ	$-\sigma$

where weights of all nodes are equal, $\omega_{t,j} = 1/8$ for all j.

The cost of product rules increases exponentially, 2^N, with the number of exogenous state variables, N. Such rules are not practical when the dimensionality is high.

Monomial non-product integration formulas

Monomial formulas are a cheap alternative for multi-dimensional problem (there is a variety of such formulas differing in accuracy and cost).

Example

Let $\epsilon_{t+1}^h \sim \mathcal{N}\left(0, \sigma^2\right)$, $h = 1$, 2, 3 be uncorrelated random variables. Consider the following monomial (non-product) integration rule with $2 \cdot 3$ nodes:

	$j = 1$	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$
$\epsilon_{t+1,j}^1$	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$	0	0	0	0
$\epsilon_{t+1,j}^2$	0	0	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$	0	0
$\epsilon_{t+1,j}^3$	0	0	0	0	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$

where weights of all nodes are equal, $\omega_{t,j} = 1/6$ for all j.

Monomial rules are practical for problems with very high dimensionality, for example, with $N = 100$, this rule has only $2N = 200$ nodes.

The multi-country model

The planner maximizes a weighted sum of N countries' lifetime utilities

$$\max_{\left\{\left\{c_t^h, k_{t+1}^h\right\}_{h=1}^N\right\}_{t=0}^\infty} E_0 \sum_{h=1}^N \lambda^h \left(\sum_{t=0}^\infty \beta^t u^h \left(c_t^h \right) \right)$$

subject to

$$\sum_{h=1}^N c_t^h + \sum_{h=1}^N k_{t+1}^h = \sum_{h=1}^N k_t^h \left(1 - \delta \right) + \sum_{h=1}^N a_t^h f^h \left(k_t^h \right),$$

where λ^h is country h's welfare weight.
Productivity of country h follows the process

$$\ln a_{t+1}^h = \rho \ln a_t^h + \epsilon_{t+1}^h,$$

where $\epsilon_{t+1}^h \equiv \varsigma_{t+1} + \varsigma_{t+1}^h$ with $\varsigma_{t+1} \sim \mathcal{N}\left(0, \sigma^2 \right)$ is identical for all countries and $\varsigma_{t+1}^h \sim \mathcal{N}\left(0, \sigma^2 \right)$ is country-specific.

Results for the multi-country model

Numb. of countr.	Polyn. degree	Numb. of coeff.	$\mathcal{E}_{mean}$ RLS-Tikh.,$\eta=10^{-5}$ MC(1), $T=10,000$	CPU	$\mathcal{E}_{mean}$ RLS-TSVD, $\kappa=10^{7}$ M2, $T=1000$	CPU
	1st	5	-4.70	4.2 min	-4.65	37 sec
	2nd	15	-4.82	19.3 min	-6.01	6.8 min
N=2	3rd	35	-4.59	57 min	-7.09	10.4 min
	4th	70	-4.57	2.6 hours	-7.99	16.3 min
	5th	126	-4.53	6.8 hours	-8.00	34.8 min
			RLS-Tikh.,$\eta=10^{-5}$ MC(1), $T=10,000$		RLS-Tikh., $\eta=10^{-5}$ Q(1), $T=1000$	
N=20	1st	41	-4.55	6.5 min	-4.75	56 sec
	2nd	861	-3.88	2.1 hours	-5.40	18 min
N=200	1st	401	-3.97	37.2 min	-4.59	16.8 min

When N=200, for RLS-Tikh.,Q(1), we use $T=2000$

Conclusion

- Ergodic set methods operate on relevant domain and have potential advantages both in terms of accuracy and cost compared to methods operating on prespecified domains.
- The performance of the existing stochastic simulation algorithms was handicapped by two problems:
  - numerical instability (because of multicollinearity);
  - large integration errors (because of low accuracy of Monte Carlo integration).
- GSSA, we fixed both of these problems:
  - approximation methods that can handle ill-conditioned problems;
  - a generalized notion of integration that accurate deterministic methods.
- GSSA demonstrated a great performance in the studied examples:
  - Numerically stable;
  - Very accurate;
  - Very simple to program;
  - Tractable for problems with high dimensionality.

LS and LAD approaches to the non-linear regression model

- Extensions to the case of the non-linear regression model,

$$y = \Psi(k, a; b) + \varepsilon$$

- NLLS computes a Taylor's expansion of $\Psi(k, a; b)$ around a initial guess, b and makes a step Δb toward a solution, $\widehat{b}$,

$$\widehat{b} \simeq b + \Delta b$$

- The step Δb is a solution to the system of normal equations,

$$J^\top J \Delta b = J^\top \Delta y$$

where $J \equiv \begin{pmatrix} \frac{\partial \Psi(k_1, a_1; b)}{\partial b_0} & \cdots & \frac{\partial \Psi(k_1, a_1; b)}{\partial b_n} \\ \cdots & \cdots & \cdots \\ \frac{\partial \Psi(k_T, a_T; b)}{\partial b_0} & \cdots & \frac{\partial \Psi(k_T, a_T; b)}{\partial b_n} \end{pmatrix}$ is Jacobian and

$$\Delta y \equiv \begin{pmatrix} y_1 - \Psi(k_1, a_1; b) \\ \cdots \\ y_T - \Psi(k_T, a_T; b) \end{pmatrix}$$

LS and LAD approaches to the non-linear regression model

- *Gauss-Newton method*,

$$\Delta b = \left(J^\top J \right)^{-1} J^\top \Delta y \text{ looks like OLS } b = \left(X^\top X \right)^{-1} X^\top y$$

  $J^\top J$ is ill-conditioned $\implies$ Employ the described approaches developed for the linear regression model.

1. Compute an inverse of the ill-conditioned matrix $J^\top J$ by using LS methods based on *SVD or QR factorization of J*.

2. Tikhonov type of regularization leading to the *Levenberg-Marquardt* method,

$$\Delta b = \left(J^\top J + \eta I_{n+1} \right)^{-1} J^\top \Delta y$$

3. Replace the ill-conditioned NLLS problem with *a non-linear LAD* (NLLAD) problem,

$$\min_b 1_T^\top |y - \Psi(k, a; b)| \simeq \min_{\Delta b} 1_T^\top |\Delta y - J \Delta b|$$

  Formulate NLLAD problem as a linear programming problem.