Merging Simulation and Projection Approaches to Solve High-Dimensional Problems

Kenneth L. Judd, Lilia Maliar and Serguei Maliar

MFM at University of Chicago

Epsilon-distinguishable set (EDS) algorithm

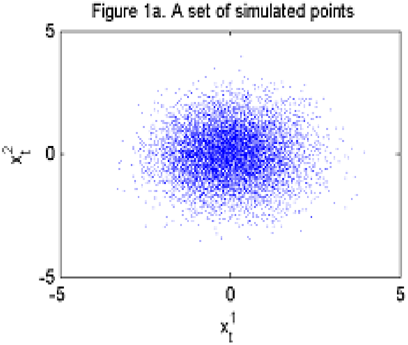
- **A novel accurate method for solving dynamic economic models:** works for problems with high dimensionality, intractable for earlier solution methods:
  - *we accurately solve models 20-50 state variables using a laptop.*
- **EDS algorithm is a global method:** can handle strong non-linearities and inequality constraints.
  - *we solve a new Keynesian model with the zero lower bound.*
- **Examples of potential applications of the EDS algorithm:**
  - macroeconomics (many heterogeneous agents);
  - international economics (many countries);
  - industrial organization (many firms);
  - finance (many assets);
  - climate change (many sectors and countries); etc.

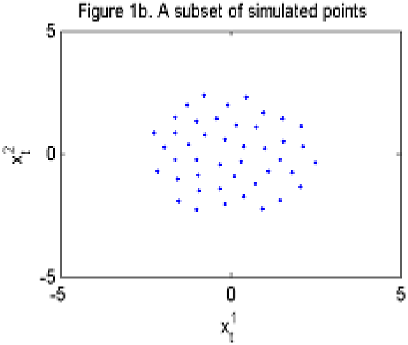
The idea of the EDS algorithm

- **EDS algorithm merges stochastic simulation and projection approaches:**
  - we use simulation to approximate the ergodic measure of the solution;
  - we construct a fixed grid covering the support of the constructed ergodic measure;
  - we use projection techniques to accurately solve the model on that grid.
- **The key novel piece of our analysis**: the EDS grid construction:
  - "*ε-distinguishable set (EDS)*" = a set of points situated at the distance at least ε from one another, where $\varepsilon > 0$ is a parameter.
- **In addition**, we use non-product monomial integration and low-cost derivative free solvers suitable for high dimensional problems.

A grid of points covering the ergodic set

An illustration of an ε-*distinguishable set*.





Remarks

- **Codes**
  - Not yet available for the EDS method;
  - But a simple and well documented MATLAB code is available for generalized stochastic simulation method (GSSA);
  - GSSA is less efficient but can also solve models with 20-50 state variables.

- **Our class of problems differs from Krusell and Smith (1998)**
  - we can have any agents (government, monetary authority, consumers firms) but not too many like 20-50, and we work with a true state space;
  - Krusell and Smith (1998) have a continuum of similar agents, and they describe aggregate behavior with a reduced state space (moments of the aggregate variables).

Mathematical foundations

We provide mathematical foundations for the EDS grid

- We establish computational complexity, dispersion, the cardinality and degree of uniformity of the EDS grid constructed on simulated series.
- We perform the typical and the worst-case analysis for the discrepancy of the EDS grid.
- We relate our results to recent mathematical literature on
  - covering problems (e.g., measuring entropy); see, Temlyakov (2011).
  - random sequential packing problems; (e.g. germ contagion); see, Baryshnikov et al. (2008).

Random sequential packing problems

- Rényi's (1958) car parking model: Cars that park at random occupy 74% of the curb.
- Constructing ε-*distinguishable sets* is like parking cars in multidimensional space.

The representative-agent neoclassical growth model:

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

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

$$\ln \theta_{t+1} = \rho \ln \theta_t + \epsilon_{t+1}, \quad \epsilon_{t+1} \sim \mathcal{N}(0, \sigma^2)$$

where initial condition (k_0, θ_0) is given;
$u(\cdot)$ = utility function; $f(\cdot)$ = production function;
c_t = consumption; k_{t+1} = capital; θ_t = productivity;
β = discount factor; δ = depreciation rate of capital;
ρ = autocorrelation coefficient of the productivity level;
σ = standard deviation of the productivity shock.

Conventional projection methods, Judd (1992)

Characteristic features

- Solve a model on a prespecified grid of points.
- Use quadrature integration for approximating conditional expectations.
- Compute polynomial coefficients of decision functions using Newton's type solver.

Projection-style method

Projection-style method.

Step 1. Choose functional forms $\widehat{K}(\cdot, b)$ for parameterizing K.
Choose a grid $\{k_m, \theta_m\}_{m=1,\ldots,M}$ on which $\widehat{K}$ is constructed.
Step 2. Choose nodes, ϵ_j, and weights, ω_j, $j = 1, ..., J$, for integration.
Compute next-period productivity $\theta'_{m,j} = \theta_m^{\rho} \exp(\epsilon_j)$ for all j, m.
Step 3. Solve for b that approximately satisfy the model's equations:

$$u'(c_m) = \beta \sum_{j=1}^{J} \omega_j \cdot \left[u'\left(c'_{m,j}\right) \left(1 - \delta + \theta'_{m,j} f'(k'_m)\right)\right],$$

$$c_m = (1-\delta) k_m + \theta_m f(k_m) - k'_m.$$

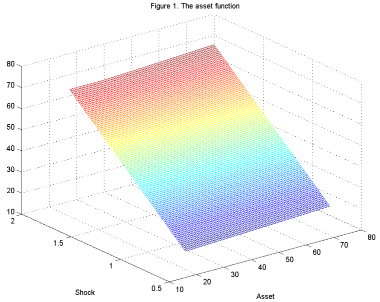
$$c'_{m,j} = (1-\delta) k'_m + \theta'_{m,j} f(k'_m) - k''_{m,j}$$

where $k'_m = \widehat{K}(k_m, \theta_m; b)$ and $k''_{m,j} = \widehat{K}\left(k'_m, \theta'_{m,j}; b\right)$.

3 potential curses of dimensionality: 1) grid construction;
2) approximation of integrals; 3) solvers.

Curse of dimensionality 1: Tensor product grid

Tensor product rules $\Rightarrow$ number of grid points grows exponentially with number of state variables.



- 2 state variables with 4 grid points $\Rightarrow 4 \times 4 = 4^2 = 16$
- 3 state variables with 4 grid points $\Rightarrow 4^3 = 64$
- 10 state variables with 4 grid points $\Rightarrow 4^{10} = 1,048,576$

$\Rightarrow$ **Not tractable even for moderately large models.**

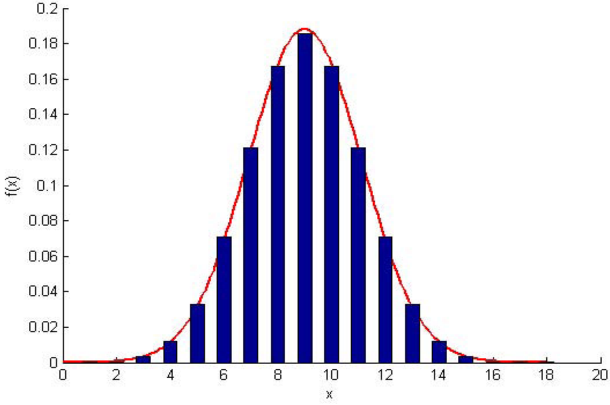
Approximation of integrals

Integral is approximated by weighted sum of integrand evaluated in a set of integration nodes

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

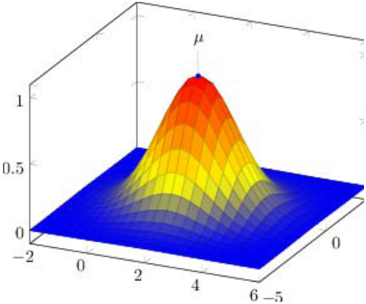
where $\{\varepsilon_j\}_{j=1}^{J}$ = integration nodes, $\{\omega_j\}_{j=1}^{J}$ = integration weights.

Nodes and weights for one dimensional distribution



Curse of dimensionality 2: tensor-product integration

Tensor product integration rules $\Rightarrow$ number of integration nodes grows exponentially with number of state variables.



- 2 shocks with 4 nodes $\Rightarrow 4 \times 4 = 4^2 = 16$
- 3 shocks with 4 nodes $\Rightarrow 4^3 = 64$
- 10 shocks with 4 nodes $\Rightarrow 4^{10} = 1,048,576$

$\Rightarrow$ **Not tractable even for moderately large number of shocks**.

Curse of dimensionality 3: Newton solvers

How do we find b for capital function $\widehat{K}(k, \theta; b)$?

Projection-style method.

Step 1. ...

Step 2. ...

Step 3. Solve for b that approximately satisfy the model's equations:

$$u'(c_m) = \beta \sum_{j=1}^{J} \omega_j \cdot \left[u'\left(c'_{m,j}\right)\left(1 - \delta + \theta'_{m,j} f'(k'_m)\right)\right],$$

$$c_m = (1-\delta) k_m + \theta_m f(k_m) - k'_m.$$

$$c_{m,j} = (1-\delta) k'_m + \theta'_{m,j} f(k'_m) - k''_{m,j}$$

where $k'_m = \widehat{K}(k_m, \theta_m; b)$ and $k''_{m,j} = \widehat{K}\left(k'_m, \theta'_{m,j}; b\right)$.

$\Rightarrow$ The larger is the number of state variables and the more complicated are the equations, the higher is the cost of solving for b using conventional Newton-style methods.

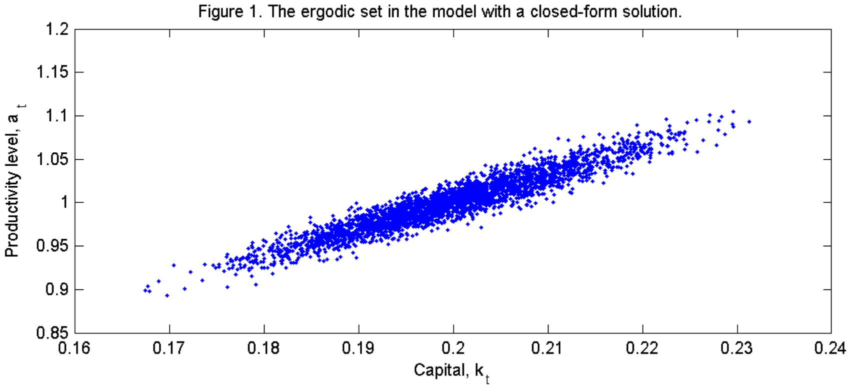
Stochastic simulation methods, e.g., Marcet (1988)

Characteristic features

- Compute a solution on simulated series.
  - Fix shocks $\{\theta_t\}_{t=1}^{T}$.
  - Guess a decision function $\widehat{K}(k, \theta; b)$;
  - Simulate time series $\{k_t, c_t\}_{t=1}^{T}$
  - Check equilibrium conditions and recompute $\widehat{b}$;
  - Iterate on b until convergence.
- Use Monte Carlo integration for approximating conditional expectations.
- Use learning techniques for solving for parameters of decision functions.

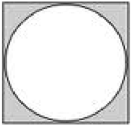
Advantage of stochastic simulation method

Advantage of stochastic simulation method: "Grid" is adaptive: we solve the model only in the area of the state space that is visited in equilibrium.



Reduction in cost in a 2-dimensional case

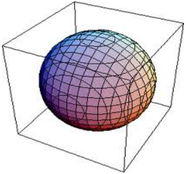
- How much can we save on cost using the ergodic-set domain comparatively to the hypercube domain?
- Suppose the ergodic set is a circle (it was an ellipse in the figure).
- In the 2-dimensional case, a circle inscribed within a square occupies about 79% of the area of the square.
- The reduction in 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 d-dimensional case, the ratio of a hypersphere's volume to a hypercube's volume

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

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

Shortcomings of stochastic simulation approach

1. Simulated points is not an efficient choice for constructing a grid:
   - there are many closely situated and hence, redundant points;
   - there are points outside the high probability set.
2. Simulated points is not an efficient choice for the purpose of integration – accuracy of Monte Carlo integration is low.

$$E_t\left[y_{t+1}\right] \approx \overline{y}_{t+1} \equiv \sum_{\tau=1}^{n} y_{\tau+1}$$

Suppose $var\left(y_{\tau+1}\right) = 1\%$ (like in RBC models)

$$n = 100 \text{ draws } \Rightarrow var\left(\overline{y}_{t+1}\right) = 0.1\%$$

$$n = 10,000 \text{ draws } \Rightarrow var\left(\overline{y}_{t+1}\right) = 0.01\%$$

Monte Carlo method has slow $\sqrt{n}$ rate of convergence.

$\Rightarrow$ *The overall accuracy of solution is restricted by low accuracy of Monte Carlo integration*, e.g., PEA by Marcet's (1988) has low accuracy.

Inefficiency of Monte Carlo integration

Why is Monte Carlo integration inefficient?

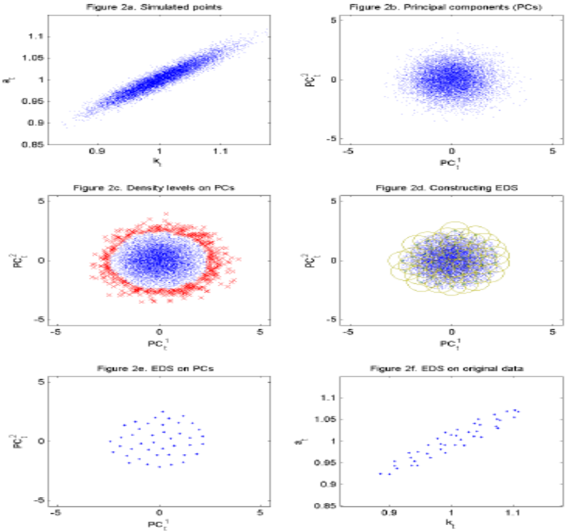
- Because we compute expectations from noisy simulated data as do econometricians who do not know true density of DGP.
- But we do know the true density of DGP (we define the shocks ourselves, e.g., $\ln \theta_{t+1} = \rho \ln \theta_t + \epsilon_{t+1}$).
- We can compute integrals far more accurately using quadrature methods based on true density of DGP!

Merging projection and stochastic simulation

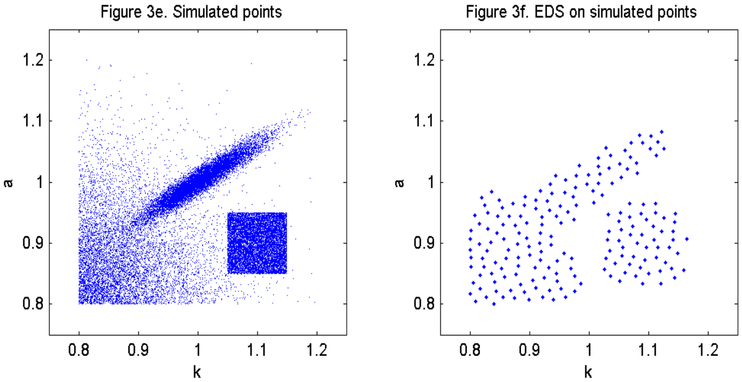
What do we do?

- **Similar to stochastic simulation approach:** use simulation to identify and approximate the ergodic set.
- **Similar to projection approach:** construct a fixed EDS grid and use the quadrature-style integration to accurately solve the model on that grid.
- We use integration and optimization methods that are tractable in high dimensional problems: *non-product monomial integration formulas and derivative-free solvers*.

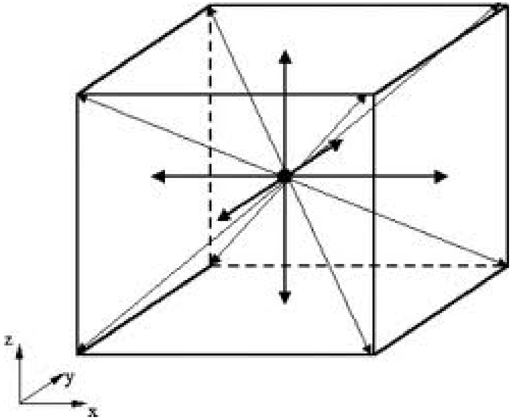
Our ingredient 1: the EDS grid construction



Our ingredient 1 (cont.): the EDS grid construction



Our ingredient 2: non-product integration



Our ingredient 2 (cont): non-product integration

- Monomial formulas are a cheap alternative for approximating high-dimensional integrals.
- There is a variety of monomial formulas differing in accuracy and cost.

Example

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

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
ϵ_j^1	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$	0	0	0	0
ϵ_j^2	0	0	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$	0	0
ϵ_j^3	0	0	0	0	$\sigma\sqrt{3}$	$-\sigma\sqrt{3}$

where weights of all nodes are equal, $\omega_j = 1/6$ 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.

Our ingredient 3: derivative-free solvers

Fixed-point iteration

- The cost of Newton's type method grows quickly with dimensionality because of the growing number of terms in Jacobian and Hessian.
- A simple and efficient alternative is fixed-point iteration

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

  where $\xi \in (0, 1)$ is damping parameter.
- Cost of fixed-point iteration grows little with dimensionality.
- Fixed-point iteration works for very high dimensions, like 400 state variables!

Combining all together: EDS algorithm iterating on Euler equation

Parameterize the RHS of the Euler equation by a polynomial $\widehat{K}(k, \theta; b)$,

$$K(k, \theta) \approx \widehat{K}(k, \theta; b) = b_0 + b_1 k + b_2 \theta + ... + b_n \theta^L$$

Step 1. Simulate $\{k_t, \theta_t\}_{t=1}^{T+1}$. Construct an EDS grid, $\{k_m, \theta_m\}_{m=1}^{M}$.
Step 2. Fix $b \equiv (b_0, ..., b_n)$. Given $\{k_m, \theta_m\}_{m=1}^{M}$ solve for $\{c_m\}_{m=1}^{M}$.
Step 3. Use $\theta'_{m,j} = \theta_m^{\rho} \exp(\epsilon_j)$ to implement monomial integration:

$$\widehat{k}'_m = \sum_{j=1}^{J} \left\{ \beta \frac{u'\left(c'_{m,j}\right)}{u'(c_m)} \left[1 - \delta + \theta'_{m,j} f'\left(k'_m\right)\right] k'_m \right\} \omega_j .$$

Regress $\widehat{k}'_m$ on $\left(1, k_m, \theta_m, k_m^2, k_m \theta_m, \theta_m^2, ..., \theta_m^L\right) \Longrightarrow$ get $\widehat{b}$.
Step 4. Solve for the coefficients using damping,

$$b^{(j+1)} = (1 - \xi) b^{(j)} + \xi \widehat{b}, \quad \xi \in (0, 1).$$

Iterate on b until convergence.

Representative-agent model: parameters choice

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 $\gamma \in \left\{\frac{1}{5}, 1, 5\right\}$.
Process for shocks: $\ln \theta_{t+1} = \rho \ln \theta_t + \epsilon_{t+1}$ with $\rho = 0.95$ and $\sigma = 0.01$.
Discount factor: $\beta = 0.99$.
Depreciation rate: $\delta = 0.025$.
Accuracy is measured by an Euler-equation residual,

$$\mathcal{R}(k_i, \theta_i) \equiv E_i \left[\beta \frac{c_{i+1}^{-\gamma}}{c_i^{-\gamma}} \left(1 - \delta + \alpha \theta_{i+1} k_{i+1}^{\alpha-1}\right) \right] - 1.$$

Table 1. Accuracy and speed of the Euler equation EDS algorithm in the representative-agent model

Polynomial degree	Mean error	Max error	CPU (sec)
1st degree	-4.29	-3.31	24.7
2nd degree	-5.94	-4.87	0.8
3rd degree	-7.26	-6.04	0.9
4th degree	-8.65	-7.32	0.9
5th degree	-9.47	-8.24	5.5

Target number of grid points is $\overline{M} = 25$.
Realized number of grid points is $M(\varepsilon) = 27$.
Mean and Max are unit-free Euler equation errors in log10 units, e.g.,

- -4 means $10^{-4} = 0.0001 \ \ (0.01\%)$;
- -4.5 means $10^{-4.5} = 0.0000316 \ \ (0.00316\%)$.

Benchmark parameters: $\gamma = 1$, $\delta = 0.025$, $\rho = 0.95$, $\sigma = 0.01$.
In the paper, also consider $\gamma = 1/5$ (low risk aversion) and $\gamma = 5$ (high risk aversion). Accuracy and speed are similar.

Autocorrection of the EDS grid

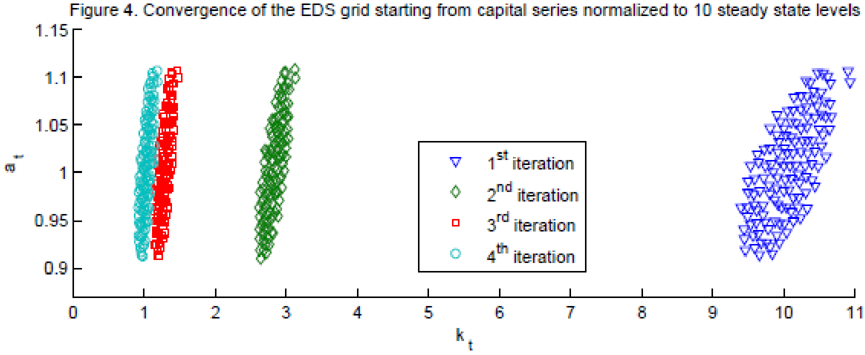


Table 2: Accuracy and speed in the one-agent model: Smolyak grid versus EDS grid

	Test on a simulation				Test on a hypercube			
Polyn.	Smolyak grid		EDS grid		Smolyak grid		EDS grid	
deg.	Mean	Max	Mean	Max	Mean	Max	Mean	Max
1st	-3.31	-2.94	-4.23	-3.31	-3.25	-2.54	-3.26	-2.38
2nd	-4.74	-4.17	-5.89	-4.87	-4.32	-3.80	-4.41	-3.25
3rd	-5.27	-5.13	-7.19	-6.16	-5.39	-4.78	-5.44	-4.11

EDS algorithm iterating on Bellman equation

Parameterize the value function by a polynomial $V \approx \widehat{V}(\cdot; b)$:

$$\widehat{V}(k, \theta; b) = b_0 + b_1 k + b_2 \theta + ... + b_n \theta^L .$$

Step 1. Find $\widehat{K}$ corresponding to $\widehat{V}(\cdot; b)$. Simulate $\{k_t, \theta_t\}_{t=1}^{T+1}$.
Construct an EDS grid, $\{k_m, \theta_m\}_{m=1}^{M}$.
Step 2. Fix $b \equiv (b_0, ..., b_n)$. Given $\{k_m, \theta_m\}_{m=1}^{M}$ solve for $\{c_m\}_{m=1}^{M}$.
Step 3. Use $\theta'_{m,j} = \theta_m^{\rho} \exp(\epsilon_j)$ to implement monomial integration:

$$V_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \widehat{V}\left(k'_m, \theta'_{m,j}; b\right) \omega_j .$$

Regress V_m on $\left(1, k_m, \theta_m, k_m^2, k_m \theta_m, \theta_m^2, ..., \theta_m^L\right) \Longrightarrow$ get $\widehat{b}$.
Step 4. Solve for the coefficients using damping,

$$b^{(j+1)} = (1 - \xi) b^{(j)} + \xi \widehat{b}, \quad \xi \in (0, 1) .$$

Iterate on b until convergence.

Table 3. Accuracy and speed of the Bellman equation EDS algorithm in the representative-agent model

Polynomial degree	Mean error	Max error	CPU (sec)
1st degree	–	–	–
2nd degree	-3.98	-3.11	0.5
3rd degree	-5.15	-4.17	0.4
4th degree	-6.26	-5.12	0.4
5th degree	-7.42	-5.93	0.4

Target number of grid points is $\overline{M} = 25$.
Realized number of grid points is $M(\varepsilon) = 27$.

Multi-country model

The planner maximizes a weighted sum of N countries' utility functions:

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

subject to

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

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

$$\ln \theta_{t+1}^h = \rho \ln \theta_t^h + \epsilon_{t+1}^h,$$

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.

Table 3. Accuracy and speed in the multi-country model

	Polyn. degree	M1			Q(1)		
		Mean	Max	CPU	Mean	Max	CPU
N=2	1st	−4.09	−3.19	44 sec	−4.07	−3.19	45 sec
	2nd	−5.45	−4.51	2 min	−5.06	−4.41	1 min
	3rd	−6.51	−5.29	4 min	−5.17	−4.92	2 min
N=20	1st	−4.21	−3.29	20 min	−4.17	−3.28	3 min
	2nd	−5.08	−4.17	5 hours	−4.83	−4.10	32 min
N=40	1st	−4.23	−3.31	5 hours	−4.19	−3.29	2 hours
	2nd	−	−	-	−4.86	−4.48	24 hours
N=100	1st	−4.09	−3.24	10 hours	−4.06	−3.23	36 min
N=200	1st	−	−	-	−3.97	−3.20	2 hours

M1 means monomial integration with 2N nodes; Q(1) means quadrature integration with one node in each dimension; Mean and Max are mean and maximum unit-free Euler equation errors in log10 units, respectively; CPU is running time.

A new Keynesian (NK) model

A stylized new Keynesian model with Calvo-type price frictions and a Taylor (1993) rule with the ZLB

- *Literature that estimates the models:*
  -Christiano, Eichenbaum and Evans (2005), Smets and Wouters (2003, 2007), Del Negro, Schorfheide, Smets and Wouters (2007).
- *Literature that finds numerical solutions:* mostly relies on local (perturbation) solution methods. Few papers apply global solution methods to low-dimensional problems.
- *Perturbation*:
  -most use linear approximations (Christiano, Eichenbaum&Rebelo, 2009);
  -some use quadratic approx. (Kollmann, 2002, Schmitt-Grohé&Uribe, 2007);
  -very few use cubic approximations (Rudebusch and Swanson, 2008).
- *Global solution methods*: at most 4 state variables and simplifying assumptions.
  -Adam and Billi (2006): all except one FOCs are linearized;
  -Adjemian and Juillard (2011): extended path method of Fair&Taylor (1984) $\Rightarrow$ perfect foresight.

A new Keynesian (NK) model

Assumptions:

- *Households* choose consumption and labor.
- Perfectly competitive *final-good firms* produce goods using intermediate goods.
- Monopolistic *intermediate-good firms* produce goods using labor and are subject to sticky price (á la Calvo, 1983).
- *Monetary authority* obeys a Taylor rule with zero lower bound (ZLB).
- *Government* finances a stochastic stream of public consumption by levying lump-sum taxes and by issuing nominal debt.
- *6 exogenous shocks and 8 state variables* $\Longrightarrow$ The model is large scale (it is expensive to solve or even intractable under conventional global solution methods that rely on product rules).

The representative household

The utility-maximization problem:

$$\max_{\{C_t,L_t,B_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta}-1}{1+\vartheta}\right]$$

$$\text{s.t. } P_t C_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} + T_t = B_{t-1} + W_t L_t + \Pi_t$$

where $\left(B_0, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}\right)$ is given.
– C_t, L_t, and B_t = consumption, labor and nominal bond holdings, resp.;
– P_t, W_t and R_t = the commodity price, nominal wage and (gross) nominal interest rate, respectively;
– T_t = lump-sum taxes;
– Π_t = the profit of intermediate-good firms;
– β = discount factor; $\gamma > 0$ and $\vartheta > 0$.

The representative household

Stochastic processes for shocks

- $\eta_{u,t}$ and $\eta_{L,t}$ = exogenous preference shocks;
- $\eta_{B,t}$ = exogenous premium in the return to bonds;

$$\eta_{u,t+1} = \rho_u \eta_{u,t} + \epsilon_{u,t+1}, \qquad \epsilon_{u,t+1} \sim \mathcal{N}\left(0, \sigma_u^2\right)$$
$$\eta_{L,t+1} = \rho_L \eta_{L,t} + \epsilon_{L,t+1}, \qquad \epsilon_{L,t+1} \sim \mathcal{N}\left(0, \sigma_L^2\right)$$
$$\eta_{B,t+1} = \rho_B \eta_{B,t} + \epsilon_{B,t+1}, \qquad \epsilon_{B,t+1} \sim \mathcal{N}\left(0, \sigma_B^2\right)$$

Final-good producers

The profit-maximization problem:

- Perfectly competitive producers
- Use intermediate goods $i \in [0,1]$ as inputs

$$\max_{Y_t(i)} \quad P_t Y_t - \int_0^1 P_t(i)\, Y_t(i)\, di$$

$$\text{s.t. } Y_t = \left(\int_0^1 Y_t(i)^{\frac{\varepsilon-1}{\varepsilon}} di \right)^{\frac{\varepsilon}{\varepsilon-1}}, \qquad \varepsilon \geq 1 \tag{1}$$

– $Y_t(i)$ and $P_t(i)$ = quantity and price of an intermediate good i, resp.;
– Y_t and P_t = quantity and price of the final good, resp.;
– Eq (1) = production function (Dixit-Stiglitz aggregator function).

Result 1: Demand for the intermediate good i: $Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}$.

Result 2: Aggregate price index $P_t = \left(\int_0^1 P_t(i)^{1-\varepsilon} di \right)^{\frac{1}{1-\varepsilon}}$.

Intermediate-good producers

The cost-minimization problem:

- Monopolisticly competitive
- Use labor as an input
- Are hit by a productiviy shock
- Are subject to sticky prices

$$\min_{L_t(i)} \quad \text{TC}\left(Y_t\left(i\right)\right) = \left(1 - v\right) W_t L_t\left(i\right)$$

$$\text{s.t. } \; Y_t\left(i\right) = \exp\left(\eta_{\theta,t}\right) L_t\left(i\right)$$

$$\eta_{\theta,t+1} = \rho_\theta \eta_{\theta,t} + \epsilon_{\theta,t+1}, \qquad \epsilon_{\theta,t+1} \sim \mathcal{N}\left(0, \sigma_\theta^2\right)$$

– TC $=$ nominal total cost (net of government subsidy v);
– $L_t\left(i\right) =$ labor input;
– $\exp\left(\eta_{\theta,t}\right)$ is the productivity level.

Intermediate-good producers (price decisions)

Calvo-type price setting:
$1-\theta$ of the firms sets prices optimally, $P_t(i)=\widetilde{P}_t$, for $i\in[0,1]$;
θ is not allowed to change the price, $P_t(i)=P_{t-1}(i)$, for $i\in[0,1]$.

The profit-maximization problem of a reoptimizing firm i:

$$\max_{\widetilde{P}_t} \sum_{j=0}^{\infty}\beta^j\theta^j E_t\left\{\Lambda_{t+j}\left[\widetilde{P}_t Y_{t+j}(i)-P_{t+j}\mathrm{mc}_{t+j}Y_{t+j}(i)\right]\right\}$$

$$\text{s.t. } Y_t(i)=Y_t\left(\frac{P_t(i)}{P_t}\right)^{-\varepsilon} \tag{2}$$

– Eq (2) is the demand for an intermediate good i;
– Λ_{t+j} is the Lagrange multiplier on the household's budget constraint;
– mc_{t+j} is the real marginal cost of output at time $t+j$.

Government

The government budget constraint:

$$T_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} = P_t \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} + B_{t-1} + vW_t L_t$$

– $\frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} = G_t$ is government spending;
– $vW_t L_t$ is the subsidy to the intermediate-good firms;
– $\eta_{G,t}$ is a government-spending shock,

$$\eta_{G,t+1} = \rho_G \eta_{G,t} + \epsilon_{G,t+1}, \qquad \epsilon_{G,t+1} \sim \mathcal{N}\left(0, \sigma_G^2\right)$$

Monetary authority

Taylor rule with ZLB on the net nominal interest rate:

$$R_t = \max\left\{1, \quad R_* \left(\frac{R_{t-1}}{R_*}\right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu} \exp\left(\eta_{R,t}\right)\right\}$$

- $R_* =$ long-run gross nominal interest rate;
- $\pi_t =$ gross inflation rate between $t-1$ and t;
- $\pi_* =$ inflation target;
- $Y_{N,t} =$ natural level of output;
- $\eta_{R,t} =$ monetary shock

$$\eta_{R,t+1} = \rho_R \eta_{R,t} + \epsilon_{R,t+1}, \qquad \epsilon_{R,t+1} \sim \mathcal{N}\left(0, \sigma_R^2\right)$$

Natural equilibrium

"Natural equilibrium" - the model in which the potential inefficiencies have been eliminated:

- Natural level of output $Y_{N,t}$ in the Taylor rule is a solution to a planner's problem

$$\max_{\{C_t,L_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right)\left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right)\frac{L_t^{1+\vartheta}-1}{1+\vartheta}\right]$$

$$\text{s.t. } C_t = \exp\left(\eta_{\theta,t}\right) L_t - G_t$$

where G_t is given.

- This implies

$$Y_{N,t} = \left[\frac{\exp\left(\eta_{\theta,t}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,t}\right)\right]^{-\gamma}\exp\left(\eta_{L,t}\right)}\right]^{\frac{1}{\vartheta+\gamma}}$$

Summary of equilibrium conditions

- **Aggregate production**

$$Y_t = \exp\left(\eta_{\theta,t}\right) L_t \Delta_t$$

- **Aggregate resource constraint**

$$C_t + G_t = Y_t$$

- **Taylor rule with ZLB on the net nominal interest rate**

$$R_t = \max\left\{1, \quad R_* \left(\frac{R_{t-1}}{R_*}\right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu} \exp\left(\eta_{R,t}\right)\right\}$$

- **Natural level of output**

$$Y_{N,t} = \left[\frac{\exp\left(\eta_{\theta,t}\right)^{1+\vartheta}}{\left[\exp\left(\eta_{G,t}\right)\right]^{-\gamma} \exp\left(\eta_{L,t}\right)}\right]^{\frac{1}{\vartheta+\gamma}}.$$

Summary of equilibrium conditions

We have

- Stochastic processes for 6 exogenous shocks $\{\eta_{u,t}, \eta_{L,t}, \eta_{B,t}, \eta_{\theta,t}, \eta_{G,t}, \eta_{R,t}\}$.
- 8 endogenous equilibrium equations & 8 unknowns $\{C_t, Y_t, R_t, L_t, \Delta_t, \pi_t, F_t, S_t\}$.
- 2 endogenous state variables $\{\Delta_{t-1}, R_{t-1}\}$.
- Thus, there are 8 (endogenous plus exogenous) state variables.

Computational papers on ZLB

How to impose the ZLB on nominal interest rate?

- Perturbation methods do not allow us to impose the ZLB in the solution procedure.
- The conventional approach in the literature is to disregard the ZLB when computing perturbation solutions and to impose the ZLB in simulations when running simulation (that is, whenever R_t happens to be smaller than 1 in simulation, we set it at 1).
  - Christiano, Eichenbaum&Rebelo (2009)
- **In contrast,** our global EDS method does allow us to impose the ZLB both in the solution and simulation procedures.

Parameter values

We calibrate the model using the results in Smets and Wouters (2003, 2007), and Del Negro, Smets and Wouters (2007).

- Preferences: $\gamma = 1$; $\vartheta = 2.09$; $\beta = 0.99$
- Intermediate-good production: $\varepsilon = 4.45$
- Fraction of firms that cannot change price: $\theta = 0.83$
- Taylor rule: $\phi_y = 0.07$; $\phi_\pi = 2.21$; $\mu = 0.82$
- Inflation target: $\pi_* \in \{1, 1.0598\}$
- Government to output ratio: $\overline{G} = 0.23$
- Stochastic processes for shocks:
  $\rho_u = 0.92$; $\rho_L = 0.25$; $\rho_B = 0.22$; $\rho_\theta = 0.95$; $\rho_R = 0.15$; $\rho_G = 0.95$
  $\sigma_u = 0.54\%$; $\sigma_L \in \{18.21\%, 40.54\%\}$; $\sigma_B = 0.23\%$; $\sigma_\theta = 0.45\%$;
  $\sigma_R = 0.28\%$; $\sigma_G = 0.38\%$

We compute 1st and 2nd perturbation solutions using Dynare, and we compute 2nd and 3rd degree EDS solutions.

Time-series solution and EDS grid

Figure 5. Simulated points and the grid for a new Keynesian model: ZLB is imposed

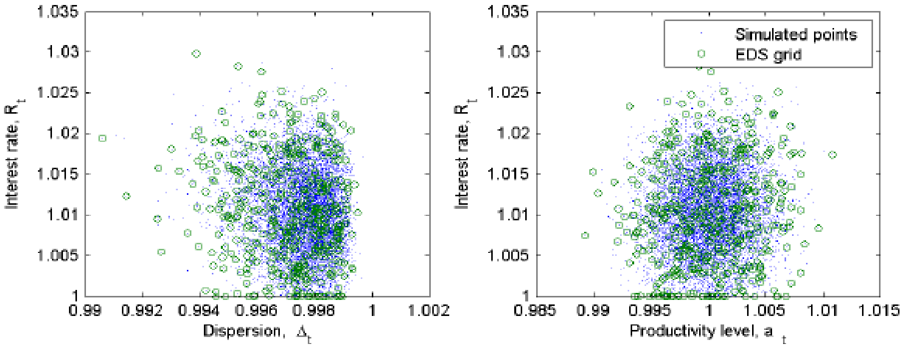


Table 4. Accuracy and speed in the NK model with 0% inflation target and 18.21% volatility of labor shock

	PER1	PER2	EDS2	EDS3
CPU	0.15		24.3	4.4
Mean	-3.03	-3.77	-3.99	-4.86
Max	-1.21	-1.64	-2.02	-2.73
R_{min}	0.9916	0.9929	0.9931	0.9927
$R_{m\theta x}$	1.0340	1.0364	1.0356	1.0358
$Fr_{(R\leq 1)}$, %	2.07	1.43	1.69	1.68
$\triangle R$, %	0.17	0.09	0.05	0
$\triangle C$, %	1.00	0.19	0.12	0
$\triangle Y$, %	1.00	0.19	0.12	0
$\triangle L$, %	0.65	0.33	0.16	0
$\triangle \pi$, %	0.30	0.16	0.11	0

PER 1 and PER 2 = 1st and 2nd order Dynare solutions; EDS2 and EDS3 = 2nd and 3rd degree EDS algorithm; Mean and Max = average and max absolute errors (in log10 units); R_{min} and R_{max} = min and max R; Fr = frequency of R$\leq$ 1; $\triangle$X = max difference from EDS3.

Table 5. Accuracy and speed in the NK model with 5.98% inflation target and 40.54% volatility of labor shock

	PER1	PER2	EDS2	EDS3
CPU	0.15		22.1	12.0
Mean	−2.52	−2.90	−3.43	−4.00
Max	−0.59	−0.42	−1.31	−1.91
R_{min}	1.0014	1.0065	1.0060	1.0060
$R_{\max}$	1.0615	1.0694	1.0653	1.0660
$Fr_{(R\leq 1)}$, %	0	0	0	0
$\triangle R$, %	0.63	0.39	0.25	0
$\triangle C$, %	6.57	1.49	0.72	0
$\triangle Y$, %	6.57	1.48	0.72	0
$\triangle L$, %	3.16	1.30	0.54	0
$\triangle \pi$, %	1.05	0.79	0.60	0

PER 1 and PER 2 = 1st and 2nd order Dynare solutions; EDS2 and EDS3 = 2nd and 3rd degree EDS; Mean and Max = average and max absolute errors (in log10 units); R_{min} and R_{max} = min and max R; Fr = frequency of $R \leq 1$; $\triangle X$ = max difference from EDS3.

Table 6. Accuracy and speed in the NK model with 0% inflation target, 18.21% volatility of labor shock and ZLB

	PER1	PER2	EDS2	EDS3
CPU	0.15		21.4	3.58
Mean	−3.02	−3.72	−3.57	−3.65
Max	−1.21	−1.34	−1.58	−1.81
R_{min}	1.0000	1.0000	1.0000	1.0000
$R_{\max}$	1.0340	1.0364	1.0348	1.0374
$Fr_{(R\leq 1)}$, %	1.76	1.19	2.46	2.23
$\triangle R$, %	0.33	0.34	0.34	0
$\triangle C$, %	4.31	3.65	2.26	0
$\triangle Y$, %	4.33	3.65	2.26	0
$\triangle L$, %	3.37	3.17	2.45	0
$\triangle \pi$, %	1.17	1.39	0.79	0

PER 1 and PER 2 = 1st and 2nd order Dynare solutions; EDS2 and EDS3 = 2nd and 3rd degree EDS; Mean and Max = average and max absolute errors (in log10 units); $R_{\min}$ and $R_{\max}$ = min and max R; Fr = frequency of $R \leq 1$; $\triangle X$ = max difference from EDS3.

Simulated series: ZLB is not imposed versus ZLB is imposed

Figure 6a. A time-series solution to a new Keynesian model: ZLB is not imposed

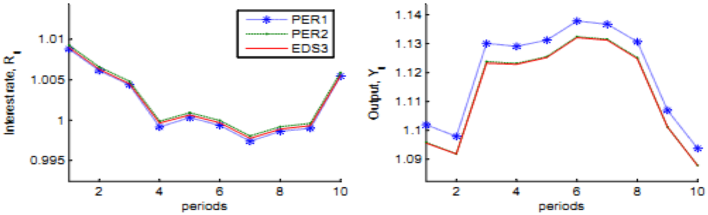
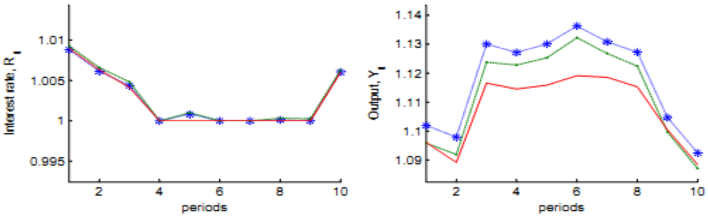


Figure 6b. A time-series solution to a new Keynesian model: ZLB is imposed



Conclusion

- The EDS algorithm accurately solves models that were considered to be unfeasible until now.
- A mix of techniques taken together allows us to address the challenges of high-dimensional problems:
    - EDS grid domain - a tiny fraction of the standard hypercube domain;
    - monomial and one-node integration rules;
    - derivative-free solvers.
- A proper coordination of the above techniques is crucial for accuracy and speed.