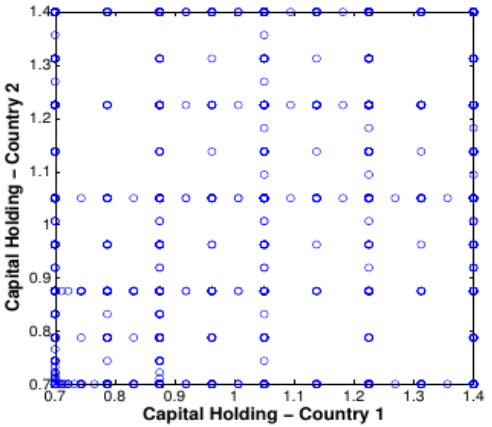
Time Iteration with Sparse Grids

Simon Scheidegger
simon.scheidegger@gmail.com
July 20th, 2017
Open Source Macroeconomics Laboratory – BFI/UChicago
(The Ramsey model & sparse grids – a canonical teaching example by J. Brumm & S. Scheidegger)



Why the Ramsey Model?

We choose the Ramsey model as our first example because:

- it is the most canonical infinite horizon optimization problem.

- it is very simple (in its basic form).

- its simplicity allows us to focus on the solution method.

- however, it can be extended to include many interesting Features.

The Deterministic Ramsey Model

$$\max U(\{c_t\}_{t=0}^{\infty}) \quad \text{s.t. } c_t + k_{t+1} \leq \underbrace{f(k_t) + (1-\delta)k_t}_{\equiv \bar{f}(k_t)}$$

$$c_t \geq 0,\ k_{t+1} \geq 0\ \forall t \in \mathbb{N}_0,\ k_0 \text{ given}$$

Where:

c_t is consumption at time t

$U(\{c_t\}_{t=0}^{\infty})$ is utility of the consumption stream $\{c_t\}_{t=0}^{\infty}$

k_t is the capital stock at time t, and k_0 the initial capital stock

$f(\cdot)$ is the production function

$\bar{f}(\cdot)$ is production including non-depreciated capital

δ is depreciation

Standard Assumptions on Preferences and Production

Production

Neoclassical Production:

$f(0) = 0,\ f \in C^2(\mathbb{R}),$
$f'(k) > 0,\ f''(k) < 0,$
$\lim_{k\to 0} f'(k) = \infty,$
$\lim_{k\to\infty} f'(k) = 0$

Special Case:

$$f(k) = k^{\alpha}$$

Cobb-Douglas with capital share α and fixed labor supply (normalized or intensive form)

Preferences

Time-separable utility:

$$U(\{t\}_{t=0}^{\infty}) = \sum_{t=0}^{\infty} \beta^t u(c_t)$$

with discount factor $0 < \beta < 1$ and $\lim_{c\to 0} U'(c) = \infty$.

Special Case:

$$u(c_t) = \begin{cases} \ln(c_t), & \gamma = 1 \\ \frac{c_t^{1-\gamma}}{1-\gamma}, & \gamma \in \mathbb{R}_+ \setminus \{1\} \end{cases}$$

CRRA utility

The Euler Equation

Due to the above assumptions:

$c_t \geq 0,\ k_{t+1} \geq 0$ are never binding

the budget constraint is always binding: $c_t = \bar{f}(k_t) - k_{t+1}$

Therefore, the Lagrangian of the maximization problem simplifies to:

$$\mathcal{L} = \sum_{t=0}^{\infty} \beta^t [u(c_t) + \lambda_t(\bar{f}(k_t) - c_t - k_{t+1})]$$

$$\frac{\partial \mathcal{L}}{\partial c_t} = 0 \Leftrightarrow u'(c_t) = \lambda_t;\ \frac{\partial \mathcal{L}}{\partial k_{t+1}} = 0 \Leftrightarrow \lambda_t = \beta \lambda_{t+1} \bar{f}'(k_{t+1})$$

Combining, we get the Euler equation(s):

$$u'(\bar{f}(k_t) - k_{t+1}) = \beta \bar{f}'(k_{t+1}) u'(\bar{f}(k_{t+1}) - k_{t+2}) \quad \forall t \in \mathbb{N}_0$$

Recursive Equilibrium

Hard to solve for an infinite sequence directly!

⇒ Reduce problem to two periods: '**today**' and '**tomorrow**'

⇒ Suppose optimal choice does not depend on ***t*** directly, just on k_t

⇒ Look for recursive equilibrium with capital k as endogenous state

⇒ A recursive equilibrium policy function p(k) must satisfy:

$$u'\left(\bar{f}(k) - p(k)\right) = \beta \cdot \bar{f}'(p(k))) \cdot u'\left(\bar{f}(p(k)) - p(p(k))\right)$$

Time Iteration (TI): The Idea

(see Judd (1998))

- Start with a guess for the policy function ‘tomorrow’

- Find policy ‘today’ that is optimal given that policy function ‘tomorrow’

- Use this policy as new guess and iterate.

- Hope that this procedure converges, i.e. that the policy does (almost) not change any more.

- The final policy (almost) satisfies the Euler equation when used ‘today’ and ‘tomorrow’.

-Then we have found an (approximate) recursive equilibrium.

Time Iteration Algorithm for the deterministic Ramsey model

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set capital grid $K = [K_1 \; K_2 \; \ldots \; K_n] \in \mathbb{R}^n_+, \; K_j < K_{j+1} \; \forall \; j$
   b) Set guess for policy function $p : [K_1, K_n] \to [K_1, K_n]$
   c) Set error tolerance for time iteration $\bar{\epsilon} > 0$
1. Main Step *(Update policy function)*
   a) For all $1 \leq j \leq n$:
   Solve Euler equation

$$u'(\bar{f}(k) - k^+) - \beta \cdot \bar{f}'(k^+) \cdot u'(\bar{f}(k^+) - k^{++}) = 0$$

   for optimal k^+ given $k = K_j$ and $k^{++} = p(k^+)$. Then, set $K_j^+ = k^+$.
   b) **Approximate** new policy $\tilde{p}$ using the data points $\left\{K_j, K_j^+\right\}_{j=1}^n$.
2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \|\tilde{p} - p\|_\infty / \|p\|_\infty$
   b) Set $p = \tilde{p}$.
   c) If $\epsilon < \bar{\epsilon}$, then stop and report results; otherwise go to step 1.

Measuring Accuracy: Recall Euler Errors I

We want a policy function that satisfies the Euler equation

$$u'(C(k)) = \beta \cdot \bar{f}'(\bar{f}(k) - C(k)) \cdot u'(C(\bar{f}(k) - C(k)))$$

at all $k \in [k_{min}, k_{max}]$, not only at k^*. We proceed as follows:

Create many points $\{\tilde{k}_i\}_{i=1}^{I} : \tilde{k}_i \in [k_{min}, k_{max}]$

Compute consumption implied by approximate policy: $\hat{c}_i = \hat{C}(\tilde{k}_i)$.

Compute consumption implied by Euler equation and approximate policy 'tomorrow': $c_i^* = u_c^{-1}\left[\beta \bar{f}'(\bar{f}(\tilde{k}_i) - \hat{c}_i) \cdot u_c\left(\hat{C}(\bar{f}(\tilde{k}_i) - \hat{c}_i)\right)\right]$

The (relative) error that the agent makes 'today' given his choice 'tomorrow' is the Euler error:

$$E_i = |\frac{\hat{c}_i}{c_i^*} - 1|$$

Measuring Accuracy: Recall Euler Errors II

Choose points $\{\tilde{k}_i\}_{i=1}^{I}$ either

- randomly (uniformly distributed) in $[k_{min}, k_{max}]$, or
- as a very fine (equidistant) grid on $[k_{min}, k_{max}]$

Later we will also look at Euler errors along a simulation path

'Bounded rationality' interpretation: The Euler error

$$E_i = |\frac{\hat{c}_i}{c_i^*} - 1|$$

is the fraction by which the approximate consumption choice today differs from the optimal one (given the approximate consumption choice tomorrow). For instance, $E_i = 0.05$ means that consumption is 5% too high or too low relative to the optimum

The Stochastic Ramsey Model

$$\max \mathbb{E}_0[U(\{c_t\})] \quad \text{s.t. } c_t + k_{t+1} \leq \underbrace{a_t f(k_t) + (1-\delta)k_t}_{\equiv \bar{f}(a_t, k_t)},$$

$$c_t \geq 0, \; k_{t+1} \geq 0 \; \forall t \in \mathbb{N}_0$$

$$\ln a_{t+1} = \rho \ln a_t + \epsilon_{t+1}, \epsilon_{t+1} \sim \mathcal{N}(0, \sigma_a)$$

$$k_0, a_0 \text{ given}$$

where the expectation is over the sequence of stocks $\{a_t\}_{t=1}^{\infty}$ given a_0.

A recursive equilibrium (capital) policy function $p(k, a)$ must satisfy:

$$u'\left(\bar{f}(k,a) - p(k,a)\right) = \beta \mathbb{E}[\bar{f}'(p(k,a)) \cdot u'\left(\bar{f}(p(k,a)) - p(p(k,a), a^+)\right)]$$

Time Iteration Algorithm for Stochastic Ramsey Model

0. Initial Step *(Set grid, initial policy, and error tolerance)*
   a) Set grid $G = [(K_1, A_1) \ \ldots \ (K_n, A_n)] \in \mathbb{R}_+^{n \times 2}$
   b) Set guess for policy function $p : [K_1, K_n] \times [A_1, A_n] \to [K_1, K_n]$
   c) Set error tolerance for time iteration $\bar{\epsilon} > 0$
1. Main Step *(Update policy function)*
   a) For all $1 \leq j \leq n$:
   Solve Euler equation

$$u'(\bar{f}(k, a) - k^+) - \beta \mathbb{E}[\bar{f}'(k^+, a^+) \cdot u'(\bar{f}(k^+, a^+) - k^{++})] = 0$$

   for optimal k^+ given $k = K_j$ and $k^{++} = p(k^+, a^+)$. Then, set $K_j^+ = k^+$.
   b) **Approximate** new policy $\tilde{p}$ using the data points $\left\{K_j, K_j^+\right\}_{j=1}^n$.
2. Final Step *(Check error criterion)*
   a) Calculate error: $\epsilon = \|\tilde{p} - p\|_\infty / \|p\|_\infty$
   b) Set $p = \tilde{p}$.
   c) If $\epsilon < \bar{\epsilon}$, then stop and report results; otherwise go to step 1.

Evaluating the Expectation: Recall Quadrature

- Each time we solve the first order conditions, we have to evaluate:

$$\mathbb{E}[\bar{f}'(k^+, a^+) \cdot u'(\bar{f}(k^+, a^+) - p(k^+, a^+))]$$

- To transform the expectation into a sum we use a quadrature method.

- We choose Gauss-Hermite quadrature (see Judd 1998, p.262).

Choosing Equation Solver and Function Approximation

To implement policy function iteration, we need to choose:

→ A method for solving equations, namely the Euler equation

→ A method for approximating functions, namely the policy function

We choose for equation solving:

→ **fsolve** (from Matlab's Optimization Toolbox)

→ for interpolation: **sparse grids**, in particular spinterp (from Klimke's Sparse Grid Interpolation Toolbox – we want to avoid to introduce more complex optimizers)

Let's look at the code and see how that works ...

Comparing Sparse Grids

We solve the stochastic Ramsey model with different sparse grids:

Grid Type (in **spinterp**)	Clenshaw-Curtis	Chebyshev
Basis Function	Piecewise Linear	Global Polynomial
Points / Avg EE / Time (sec.)	13 / $7 \cdot 10^{-3}$ / 6	13 / $7 \cdot 10^{-4}$ / 7
Points / Avg EE / Time (sec.)	145 / $6 \cdot 10^{-4}$ / 132	29 / $6 \cdot 10^{-5}$ / 17
Points / Avg EE / Time (sec.)	321 / $3 \cdot 10^{-4}$ / 389	65 / $7 \cdot 10^{-6}$ / 45

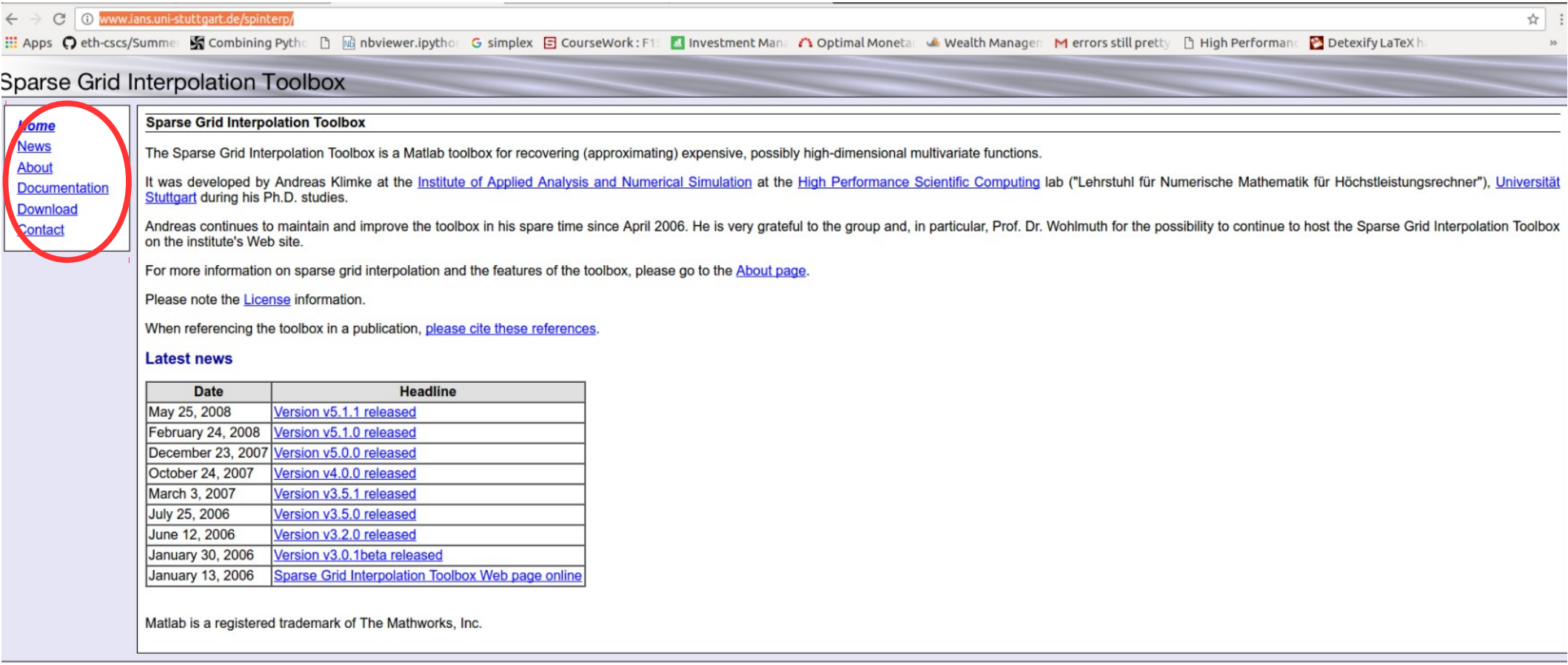
- Increasing the number of gridpoints substantially reduces the Euler errors for both types of grids.

- The global polynomial approximation performs much better in our (smooth) Application.

- For models with kinks (e.g. from occasionally binding constraints) local basis functions are preferable.

SPINTERP

http://www.ians.uni-stuttgart.de/spinterp/



Run Example Code on MIDWAY

→ **Log on to MIDWAY**

> ssh USERNAME@midway1.rcc.uchicago.edu

→ **Load matlab**

> module load matlab

→ **Start MATLAB without graphical interface**

> matlab -nojvm

→ **Go to example and run it.**

> cd OSM_Lab/SparseGrid/SparseGridCode/Econ_example_ramsey
> TimeIterationWithSparseGrids

→ Play with settings (basis functions, accuracy, etc...)