Endogenous Grid Point Methods: Solving discrete-continuous choice models

Fedor Iskhakov[1] John Rust[2] Bertel Schjerning[3]

[1]University of New South Wales

[2]Georgetown University

[3]University of Copenhagen

ZICE 2014
Zurich, February 3rd, 2014

What is EGM?

The Method of Endogenous Gridpoints — fast method for solving dynamic consumption/savings problems

1. finite and infinite horizon
2. A1. Strictly concave monotone and differentiable utility function
3. one continuous state variable (*wealth*) and one continuous choice (*consumption*)
4. particular structure of the motion rule (*intertemporal budget constraint*)
5. credit constraint

Simple problem for the methods you learned at ZICE2014, yet important model which allows for an elegant, fast and accurate solution which is easy to implement

Generalized EGM

Expand the class of problems to be solved:

1. A1. Strictly concave monotone and differentiable utility function
2. Continuous state M_t with a particular motion rule
3. Additional (discrete) state variables st_t
   A2. Transition probabilities of st_t are independent of M_t
4. One continuous (c_t) and one* discrete choice variable d_t

Examples:

- Retirement decisions along with consumption savings
- Discrete labour supply, occupation choice

Much harder problem: non-concave and non-smooth value functions in general case

EGDST software preview

Model definition + sovler, simulator and visualizer
Easier (less flexibility) and harder (mode to specify) than AMPL

1. Create and populate the model object model1
   - states model1.s=<...>,
   - decisions model1.d=<...>,
   - transition probabilities model1.trpr=<...>,
   - preferences model1.u=<...>, etc.
2. Solve, simulate and plot with model1.solve, model1.sim(<...>), model1.plot(<...>)
3. Build a calibration/estimation program around this object for example, using ktrlink(<...>)
4. Solve, calibrate/estimate the model

Summary of results

1. EGM is applicable in the class of models with one continuous (consumption) choice and additional discrete choices
2. If utility is separable in continuous and discrete choices, *extended* EGM remains very powerful in dealing with credit constraints
3. In regular cases our generalization avoids all root-finding operations

Overall:

1. Faster and more accurate than traditional approaches
2. Easy to use software package

Plan for the lecture

Theoretical topics

1. Carroll's EGM: Consumption/savings
2. Generalization of EGM to models with discrete choice: retirement model
3. Discontinuous consumption rules and kinked value functions
4. Smoothing kinks with shocks
5. Credit constraints

Practical exercises

1. (Matlab) EGDST class
2. Implementation of retirement model
   - Discrete choices
   - Absorption
   - Shocks
3. Simulations with EGDST
4. Plotting with EGDST

Simple consumption/savings model

$$V_t(M_t) = \max_{0 \leq c \leq M_t} \left[u(c) + \beta E V_{t+1}\left(\tilde{R}(M_t - c) \right) \right]$$

M_t	cash-in-hand, all resources available at period t
$A_t = M_t - c_t$	assets at the end of period t (savings)
$\tilde{R}$	*deterministic* or *stochastic* return on savings
$u(c)$	utility of current consumption

$$u(c) = \frac{c^\rho - 1}{\rho} \underset{\rho \to 0}{\longrightarrow} log(c)$$

Analytic solution (Hakansson, 1970, Phelps, 1962)

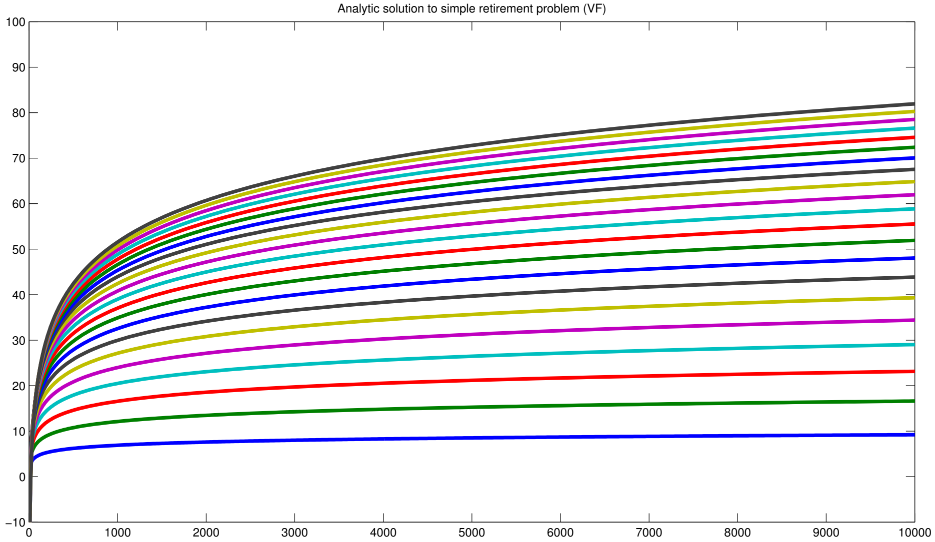
$$V_{T-t}(M) = \left[\frac{M^\rho}{\rho}\right]\left(\sum_{i=0}^{t} K^i\right)^{(1-\rho)} - \frac{1}{\rho}\left(\sum_{i=0}^{t}\beta^i\right)$$

$$V_{T-t}(M) \underset{\rho\to 0}{\to} \log(M)\left(\sum_{i=0}^{t}\beta^i\right) + K_t$$

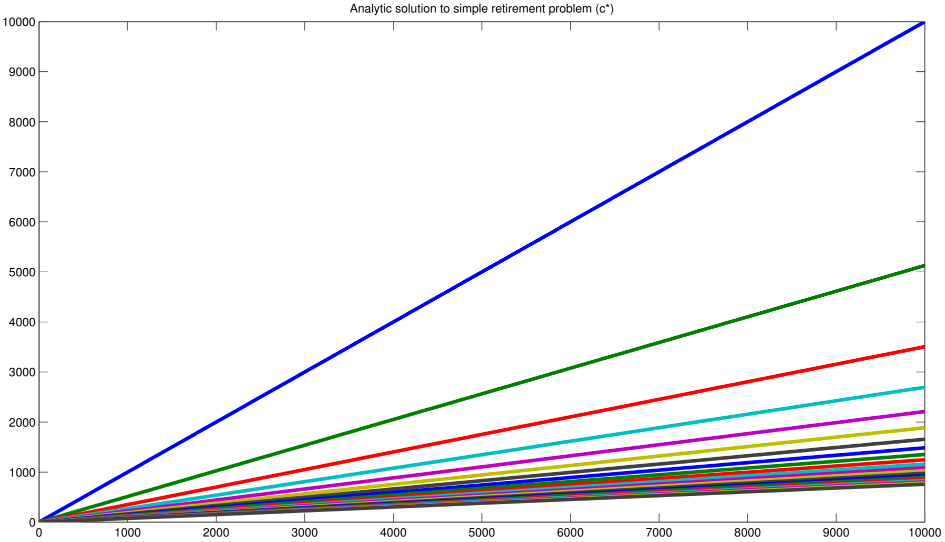
$$c_{T-t}(M) = M\left(\sum_{i=0}^{t} K^i\right)^{-1}$$

K and K_t are functions of primitives

Analytic solution : value functions



Analytic solution : consumption rule



Traditional approach : value function iterations

1. *Fix grid over M_t.* For every point on this grid:
2. In the terminal period calculate $V_T\left(M_T\right) = max_{0 \leq c_T \leq M_T}\left\{u\left(c_T\right)\right\}$ and $c_T^* = \text{argmax}_{0 \leq c_T \leq M_T}\left\{u\left(c_T\right)\right\}$
3. With $t+1$ value function at hand, proceed backward to period t and calculate $V_t\left(M_t\right) = max_{0 \leq c_t \leq M_t}\left\{u(c_t) + \beta E V_{t+1}\left(\tilde{R}(M_t - c_t)\right)\right\}$ and $c_t^* = \text{argmax}_{0 \leq c_t \leq M_t}\left\{u(c_t) + \beta E V_{t+1}\left(\tilde{R}(M_t - c_t)\right)\right\}$ using Bellman equation

Euler equation

Bellman equation:

$$V_t(M_t) = \max_{0 \le c_t \le M_t} \left[u(c_t) + \beta E V_{t+1} \left(\tilde{R}(M_t - c_t) \right) \right]$$

F.O.C. for Bellman equation: $u'(c_t) = \beta E \left[\frac{\partial V_{t+1}(M_{t+1})}{\partial M_{t+1}} \tilde{R} \right]$

Envelope theorem:

$$\frac{\partial V_t(M_t)}{\partial M_t} = \beta E \left[\frac{\partial V_{t+1}(M_{t+1})}{\partial M_{t+1}} \tilde{R} \right] \Rightarrow \frac{\partial V_t(M_t)}{\partial M_t} = u'(c_t) \Rightarrow$$

$$\Rightarrow \frac{\partial V_{t+1}(M_{t+1})}{\partial M_{t+1}} = u'(c_{t+1})$$

Euler equation to characterize the interior solutions:

$$u'(c_t) = \beta E \left[u'(c_{t+1}) \tilde{R} \right]$$

Traditional approach : solving Euler equation

1. *Fix grid over M_t.* For every point on this grid:
2. In the terminal period calculate
   $c_T^* = \text{argmax}_{0 \le c_T \le M_T} \{u(c_T)\}$
3. With $t+1$ optimal consumption rule $c_{t+1}^*(M_{t+1})$ at hand, proceed backward to period t and calculate
   c_t from equation
   $u'(c_t) = \beta E\left[u'\left(c_{t+1}^*\left(\tilde{R}(M_t - c_t)\right)\right)\tilde{R}\right]$
   to recover $c_t^*(M_t)$
4. When M_t is small enough so credit constraint binds, the Euler equation does not hold, and special provisions are necessary

What if no root-finding is necessary?

With numerical optimization

- Relatively slow: iterative numerical optimization in each point of state space!
- Hard to find global optimum in non-convex problems
- Loss of accuracy due to the absence of the point where credit constraint starts to bind on the fixed grid

Even when using state-of-the-art solvers!

Without numerical optimiation

- Much faster: no iterative methods in each point of the state space
- More accurate: using analytical structure of the problem

Endogenous gridpoint method (EGM)

Carroll 2006 *Economics Letters*
The method of endogenous gridpoints for solving dynamic stochastic optimization problems.

Idea

- Instead of searching for optimal decision in each point of the state space (traditional approaches)
- Look for the state variable (level of assets) where arbitrary chosen decision (consumption $\rightarrow$ savings) would be optimal (EGM)

Focus on end of period asset $A_t = M_t - c_t$ as a choice variable instead of consumption c_t

EGM algorithm

Start with $c_T^\star = M_T$. In each period $t = T, T-1, .., 1$:

EGM algorithm

Start with $c^{\star}_T = M_T$. In each period $t = T, T-1, .., 1$:

EGM step

1. Take a guess A = current period savings $(= M_t - c_t)$ (from fixed or adaptive list/grid)

EGM algorithm

Start with $c_T^\star = M_T$. In each period $t = T, T-1, .., 1$:

EGM step

1. Take a guess A = current period savings $(= M_t - c_t)$ (from fixed or adaptive list/grid)
2. Intertemporal budget constraint: $A \rightarrow M_{t+1}$
   $M_{t+1} = \tilde{R}(M_t - c_t) = \tilde{R} \cdot A$

EGM algorithm

Start with $c_T^\star = M_T$. In each period $t = T, T-1, .., 1$:

EGM step

1. Take a guess A = current period savings $(= M_t - c_t)$ (from fixed or adaptive list/grid)
2. Intertemporal budget constraint: $A \to M_{t+1}$
   $M_{t+1} = \tilde{R}(M_t - c_t) = \tilde{R} \cdot A$
3. Policy function at period $t+1$: $M_{t+1} \to c_{t+1}$
   $c_{t+1} = c_{t+1}^\star\left(M_{t+1}\right)$

EGM algorithm

Start with $c_T^\star = M_T$. In each period $t = T, T-1, .., 1$:

EGM step

1. Take a guess A = current period savings $(= M_t - c_t)$ (from fixed or adaptive list/grid)
2. Intertemporal budget constraint: $A \to M_{t+1}$
   $M_{t+1} = \tilde{R}(M_t - c_t) = \tilde{R} \cdot A$
3. Policy function at period $t+1$: $M_{t+1} \to c_{t+1}$
   $c_{t+1} = c_{t+1}^\star\left(M_{t+1}\right)$
4. Inverted Euler equation: $c_{t+1} \to c_t$
   $c_t = (u')^{-1}\left(\beta E\left[\tilde{R} \cdot u'\left(c_{t+1}^\star\left(M_{t+1}\right)\right)|A\right]\right)$

EGM algorithm

Start with $c_T^\star = M_T$. In each period $t = T, T-1, .., 1$:

EGM step

1. Take a guess A = current period savings $(= M_t - c_t)$ (from fixed or adaptive list/grid)
2. Intertemporal budget constraint: $A \to M_{t+1}$
$M_{t+1} = \tilde{R}(M_t - c_t) = \tilde{R} \cdot A$
3. Policy function at period $t+1$: $M_{t+1} \to c_{t+1}$
$c_{t+1} = c_{t+1}^\star (M_{t+1})$
4. Inverted Euler equation: $c_{t+1} \to c_t$
$c_t = (u')^{-1} \left(\beta E \left[\tilde{R} \cdot u' \left(c_{t+1}^\star (M_{t+1}) \right) | A \right] \right)$
5. Intratemporal budget constraint: $c_t + A = M_t \to c_t (M_t)$
$M_t = c_t + A \to c_t^\star (M_t)$

Accuracy and speed of EGM

	Traditional Euler	EGM
Running time	37 sec.	0.11 sec.
Max abs error, $c_t^\star$	5e-9	4e-14
Mean abs error, $c_t^\star$	1.4e-12	1.5e-14
Max abs error, $V_t(M, \mathbb{R})$	39.466	15.163
Mean abs error, $V_t(M, \mathbb{R})$	2.5e-02	3.2e-02

5000 grid points, 50 time periods

Comparison to the analytic solution

EGM vs. MPEC

Jørgensen 2012 *Economics Letters*
Structural Estimation of Continuous Choice Models: Evaluating EGM and MPEC.

Deaton consumption/savings model in infinite horizon, MC experiment with ML on synthetic data, 1 structural parameter

β		RMSE	Time
.70	EGM	0.002	0.1 sec.
	MPEC	0.049	112.4 sec.
.95	EGM	0.006	1.9 sec.
	MPEC	0.009	93.7 sec.
.99	EGM	0.000	5.0 sec.
	MPEC	0.000	30.9 sec.

EGM and credit constraint

Theorem: Monotonicity of savings

Monotone and concave utility function $\Rightarrow$
end-of-period assets $A_t = M_t - c_t$ are non-decreasing in M_t

- With $A = 0$ the EGM loop recovers the value of cash-in-hand M_t^{cc} that bounds the credit constrained region
- For all $M_t < M_t^{cc}$ credit constrained binds $\Rightarrow$ $c_t = M_t$
- Consumption rule in the credit constrained region is 45° line between $(0, 0)$ and (M_t^{cc}, M_t^{cc})

Where the code is

github.com/fediskhakov/egdst

Download now: tiny.cc/EGDST

How to run this code

System requirements:

1. MATLAB
2. C compiler installed on the system
   (Google for the list of supported compilers)
3. mex function in Matlab is correctly configured

Installation:

1. Download the code
   Better yet: clone the GIT repository from github
2. Copy the code in a work directory of your choice
3. Run start.m or explore the code manually
   (folder init_setup contains some help on mac setup)

Matlab implementation

```
[quadp quadw]=quadpoints(EXPN,0,1);
quadstnorm=norminv(quadp,0,1);
savingsgrid=linspace(0,MMAX,NM);
policy{TBAR}.w=[0 MMAX];
policy{TBAR}.c=[0 MMAX];
for it=TBAR-1:-1:1
 w1=exp(quadstnorm*SIGMA)*savingsgrid*(1+R);
 c1=interp1(policy{it+1}.w,policy{it+1}.c,w1,
 rhs=quadw'*(1./c1);
 policy{it}.c=1./(DF*(1+R)*rhs);
 policy{it}.w=savingsgrid+policy{it}.c;
end
```

EGDST overview

Endogenous Grid + Discrete choices + additional STates = EGDST software:

- Matlab class EGDSTmodel
- Simple pseudo-language to define the model
- Collection of methods to solve, simulate and make plots
- Intended to be inbuilt into the estimation procedure

System requirements:

1. Matlab
2. C compiler and configured *mex*

EGDST implementation (dimensions)

```
model=egdstmodel('retired','tmp_model1');

model.t0=1;                 %initial period
model.T=TBAR;               %terminal period
model.mmax=MMAX;            %max cash-on-hand
model.ngridm=NGRIDM;        %number of grid points
model.ny=NQUAD;             %number of points in qua
```

EGDST implementation (variables)

```
model.s={'Singleton state',
         {0,'dummy state'}};
model.trpr={'true',[1]};
model.feasible={'defaultfeasible',true};

model.d={'Dummy decision',
         {0,'dummy decision'}};
model.choiceset={'defaultallow',true};
```

EGDST implementation (preferences)

```
model.u={'utility',
         '(fabs(rho)<1e-10)?
          log(consumption):
          (pow(consumption,rho)-1)/rho'};
model.u={'marginal','pow(consumption,rho-1)'}
model.u={'marginalinverse',
         'pow(mutility,1/(rho-1))'};
model.param={'rho','1-crra parameter',RHO};
model.u={'extrap','pow(x,rho)'};
model.discount='df';
model.param={'df','discount factor',DF};
```

EGDST implementation (intertemporal budget)

```
model.budget={'cashinhand',
              'savings*(1+r)*shock'};
model.budget={'marginal','(1+r)*shock'};
model.param={'r','risk free return',R};
model.a0=CC;
```

EGDST implementation (shocks)

```
model.shock='lognormal';
model.shock={'sigma','sig'};
%to keep the expectation = 1 :
model.shock={'mu','-sigma*sigma/2'};
model.param={'sig','sigma param',SIGMA};
```

EGDST implementation (run methods)

```
model.compile;

model.solve;

model.plot1('c');
model.plot1('vf');
```

Generalization of EGM :

Barillas & Fernandez-Villaverde, JEDC 2007
A generalization of the endogenous grid method

1. Run EGM w.r.t. *one* choice keeping other controls fixed
2. Perform a VFI w.r.t. the rest of decision variables

Giulio Fella, working paper 2011
A generalized endogenous grid method for non-concave problems

1. Identify the regions of the problem where Euler equation is not sufficient for optimality
2. Use global optimization methods inside (VFI) and EGM outside of these regions

Simple retirement model

$$V_t(M_t, \mathbb{W}) = \max \left\{ \begin{array}{l} \max\limits_{0 \le c \le M_t} u(c) + \beta E V_{t+1} \left(\tilde{R}(M_t - c), \mathbb{R} \right) \\ \max\limits_{0 \le c \le M_t} u(c) + \beta E V_{t+1} \left(\tilde{R}(M_t + y - c), \mathbb{W} \right) \end{array} \right\}$$

$$V_t(M_t, \mathbb{R}) = \max_{0 \le c \le M_t} \left[u(c) + \beta E V_{t+1} \left(\tilde{R}(M_t - c), \mathbb{R} \right) \right]$$

$\mathbb{R}, \mathbb{W}$	retirement and working states st_t that evolve according to discrete choices $d_t \in \{\mathbb{R}, \mathbb{W}\}$
y	deterministic wage income

$$u(c) = \frac{c^\rho - 1}{\rho} - \mathbf{1}(\mathbb{W}) \underset{\rho \to 0}{\to} log(c) - \mathbf{1}(\mathbb{W})$$

How to approach discrete/continuos choice

The goal:

- Avoid root finding
- Keep efficient treatment of credit constraints

Generalized EGM ver. 1.0

1. EGM step for each discrete choice d and every state st

How to approach discrete/continuos choice

The goal:

- Avoid root finding
- Keep efficient treatment of credit constraints

Generalized EGM ver. 1.0

1. EGM step for each discrete choice d and every state st
2. Compute d-specific value functions and consumption rules

How to approach discrete/continuos choice

The goal:

- Avoid root finding
- Keep efficient treatment of credit constraints

Generalized EGM ver. 1.0

1. EGM step for each discrete choice d and every state st
2. Compute d-specific value functions and consumption rules
3. Compare the d-specific value functions to find optimal switching points (compute upper envelope)

How to approach discrete/continuos choice

The goal:

- Avoid root finding
- Keep efficient treatment of credit constraints

Generalized EGM ver. 1.0

1. EGM step for each discrete choice d and every state st
2. Compute d-specific value functions and consumption rules
3. Compare the d-specific value functions to find optimal switching points (compute upper envelope)
4. Reconstruct overall consumption rule and value function from optimal switching points

How to approach discrete/continuos choice

Generalized EGM ver. 1.0

1. EGM step for each discrete choice d and every state st
2. Compute d-specific value functions and consumption rules
3. Compare the d-specific value functions to find optimal switching points (compute upper envelope)
4. Reconstruct overall consumption rule and value function from optimal switching points

- No root finding!
- Efficient treatment of credit constraints (to be shown)
- Need to compute value functions
- Need to compute upper envelope

Is Euler equation still a necessary condition?

Clausen & Strub, 2011
A General and Intuitive Envelope Theorem.

Show that Euler equation is a necessary condition for the solution
(not sufficient in non-concave problems)

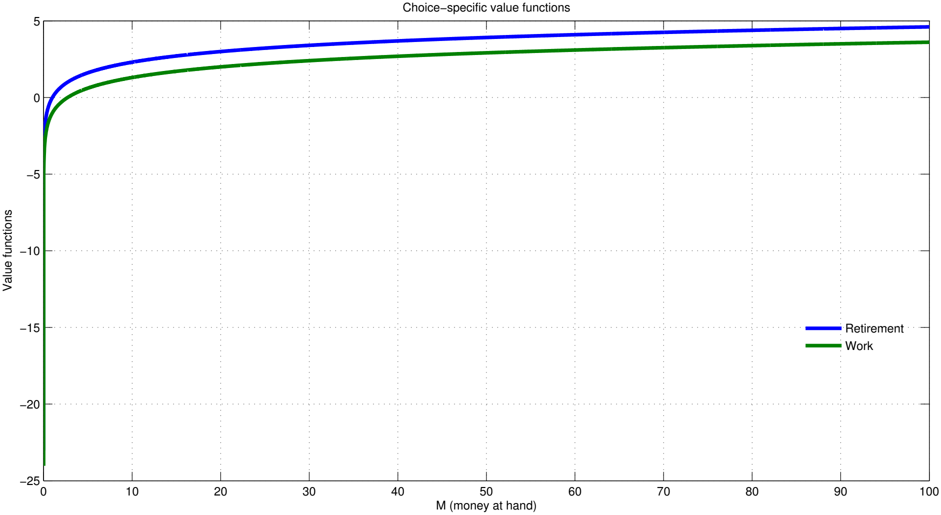
EGDST implementation (diff)

```
%states and decisions
model.s={'Labour market state',
         {0,'retired',1,'working'}};
model.d={'Retirement decision',
         {0,'Retirement',1,'Work'}};
%feasibility of states
model.feasible={'defaultfeasible',true};
%transition probabilities
model.trpr={'dc1==0',[1 0;1 0]};
model.trpr={'dc1==1',[0 1;0 1]};
%choice sets
model.choiceset={'defaultallow',true};
model.choiceset={'ist==0 && id==1',
                 'Retirement is absorbing'};
```

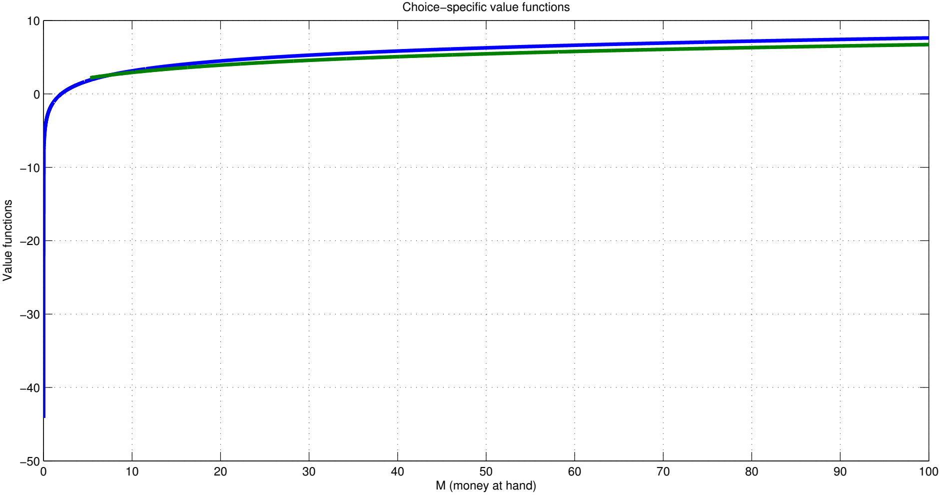
EGDST implementation (diff)

```
%utility
model.u={'utility',
 '(fabs(rho)<1e-10?log(consumption):
  (pow(consumption,rho)-1)/rho)-(id?duw:0.0)'}
model.param={'duw','parameter',duw};
model.budget={'cashinhand',
 'savings*(1+r)*shock+(id?wage:0.0)'};
model.param={'wage','Workers wage',wage};
```

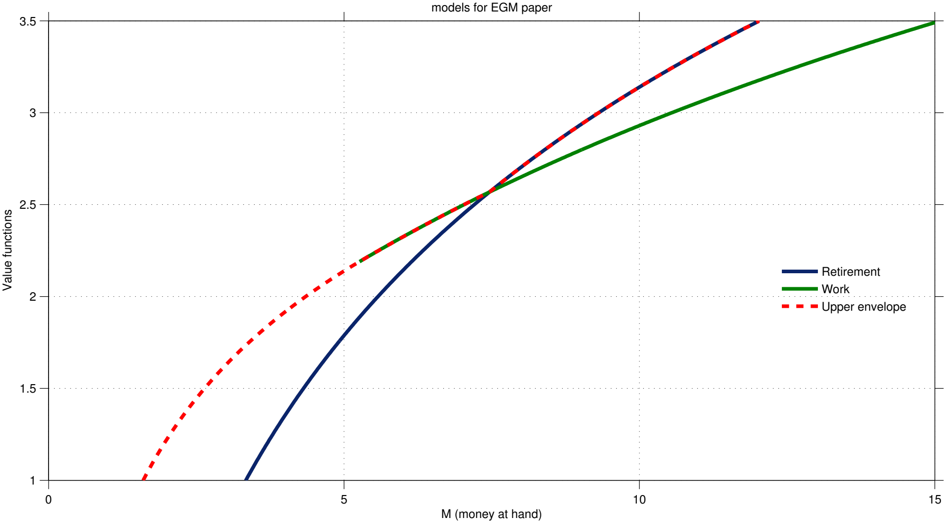
Period T : choice specific value functions



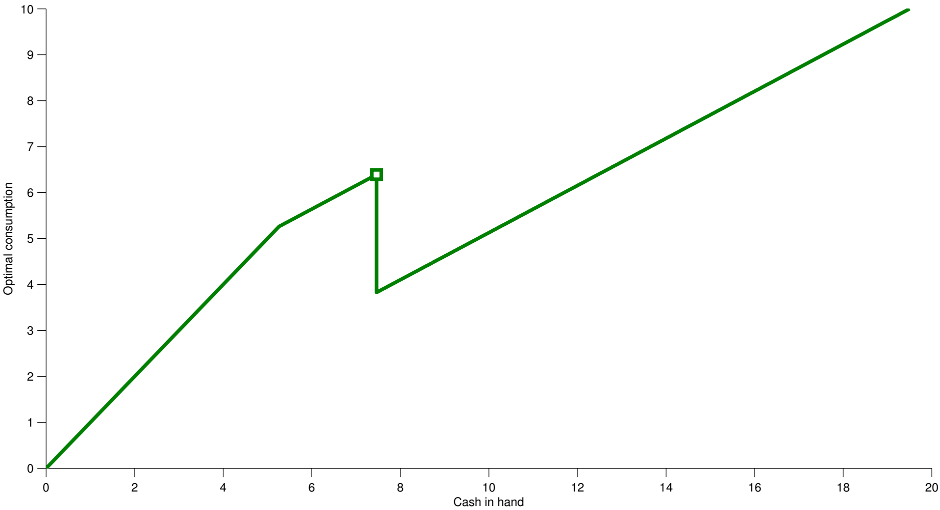
Period $T-1$: Choice specific VF



Period $T-1$: Choice specific VF



Period $T-1$: Optimal consumption



So, what is going on

1. *d*-specific value functions intersect
   (due to trade-off between income and disutility of work)

   $\Downarrow$

2. The upper envelope of the value functions has a kink

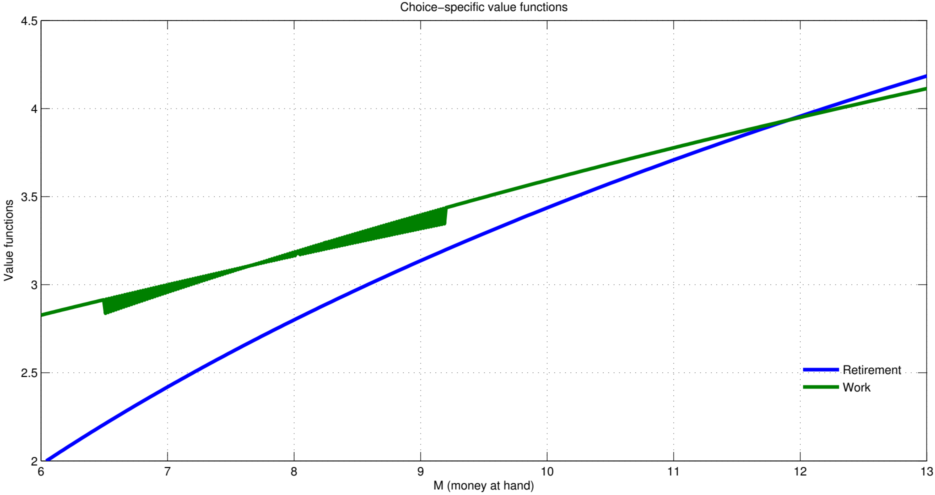
   $\Downarrow$

3. Derivative of the value function has a discontinuity at the kink

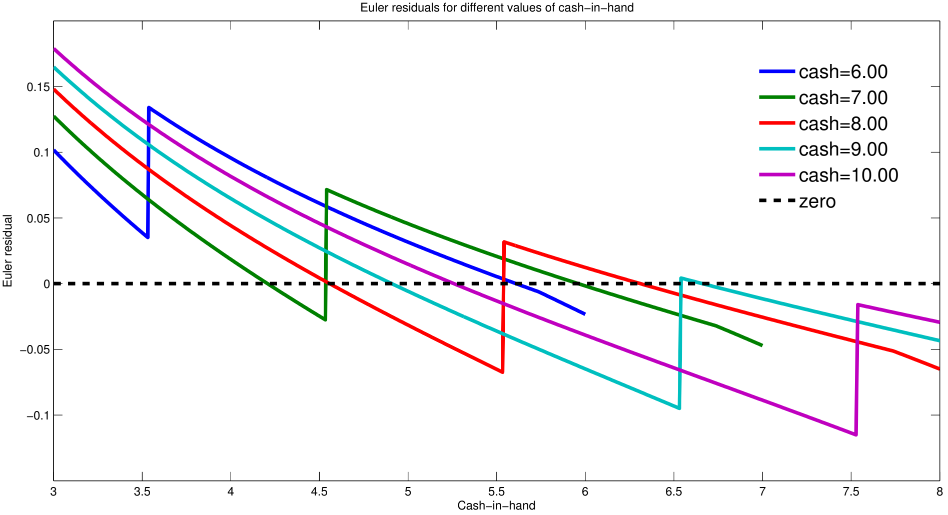
   $\Downarrow$

4. Optimal consumption rule one period prior has a discontinuity
   (which translates through the Euler euqation)

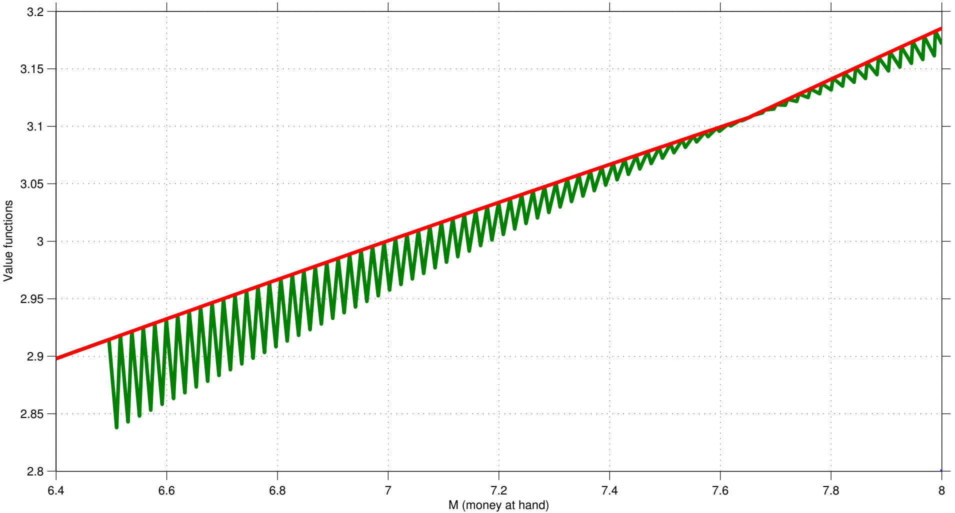
Period $T-2$: Choice specific VF



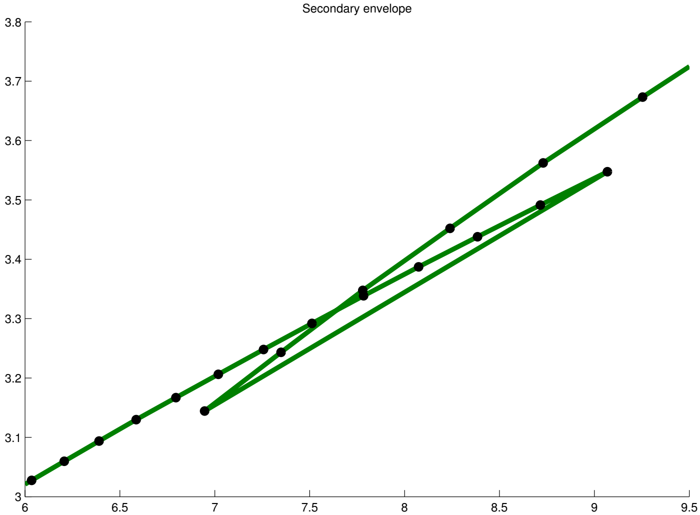
Multiple zeros of Euler residuals



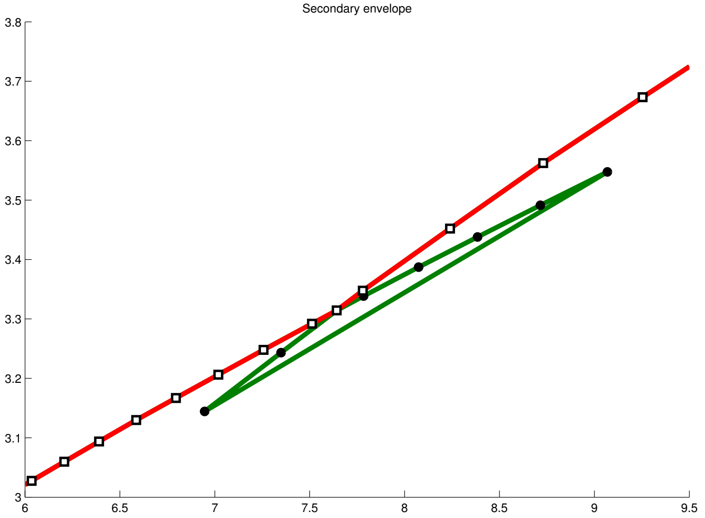
Period $T-2$: Secondary upper envelope



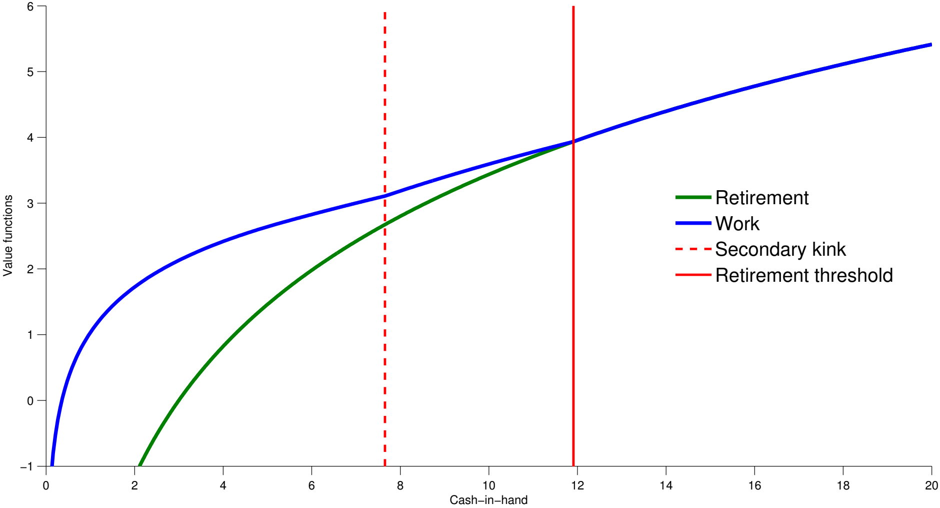
Period $T-2$: Secondary upper envelope: how



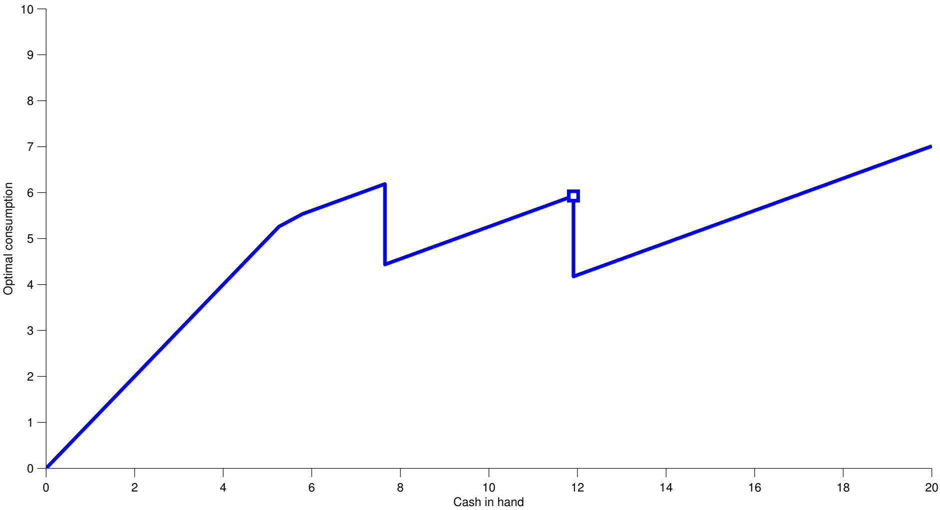
Period $T-2$: Secondary upper envelope: result



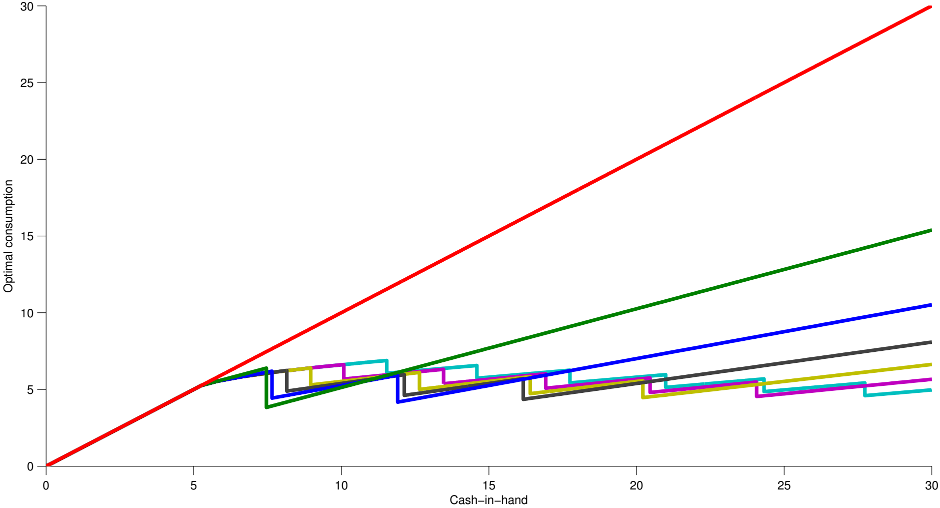
Period $T-2$: VF


Value functions
Cash–in–hand
Retirement
Work
Secondary kink
Retirement threshold
0
2
4
6
8
10
12
14
16
18
20
−1
0
1
2
3
4
5
6

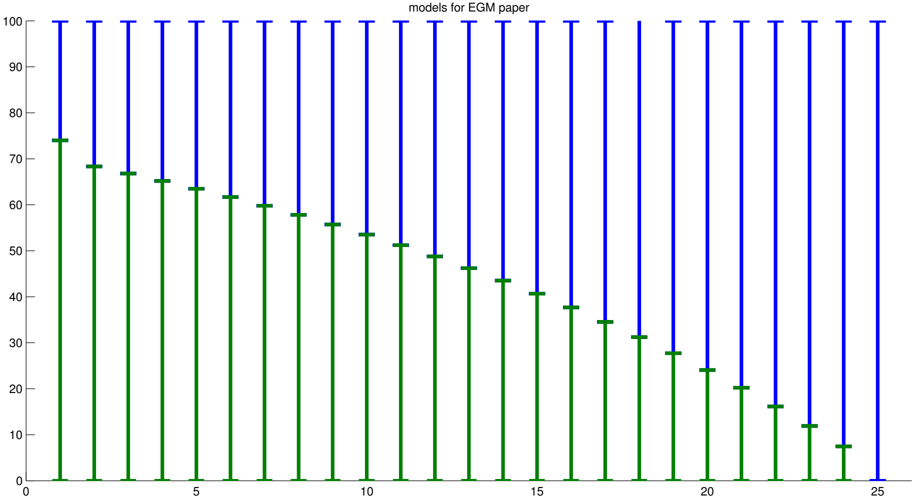
Period $T-2$: Optimal consumption



Optimal consumption (many periods)



Optimal retirement (many periods)



Generalized EGM full algorithm

Generalized EGM ver. 2.0

1. Start from terminal period, compute optimal consumption rule and value function. Loop backwards over time:
2. EGM step for each discrete choice d and every state st
3. Compute d-specific value functions and consumption rules
4. Compute the "secondary" upper envelope over the "zig-zag" regions of the d-specific value functions and update the corresponding consumption rules
5. Compare the d-specific value functions to find optimal switching points (compute upper envelope)
6. Reconstruct overall consumption rule and value function from optimal switching points

Final remarks

1. When discrete choice are "sufficiently different" the computation of the upper envelope may be difficult because the d-specific endogenous grids may not overlap
   $\downarrow$
   Use adaptive sequences of savings to contain the resulring endogenous grid in the desired bounds

2. What if a "zig-zag" region is missed by coarse endogenous grid?
   $\downarrow$
   We are working on assessing the error bounds

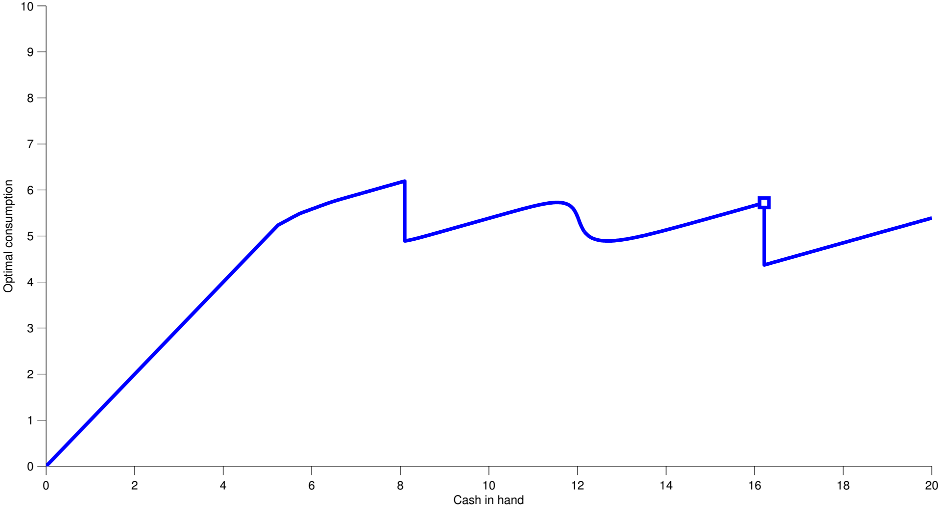
Random returns $\tilde{R}$

- **Shocks do help**: smooth out kinks and discontinuities
- **Size matters**: small shocks may not be enough
  - Sharp continuous declines in optimal consumption may lead to a discontinuity/kink in preceding period
- Expectations have to be taken over discontinuous functions
  - More discontinuities may be introduced by sloppy computation
  - Separate integration over “continuous” intervals works better

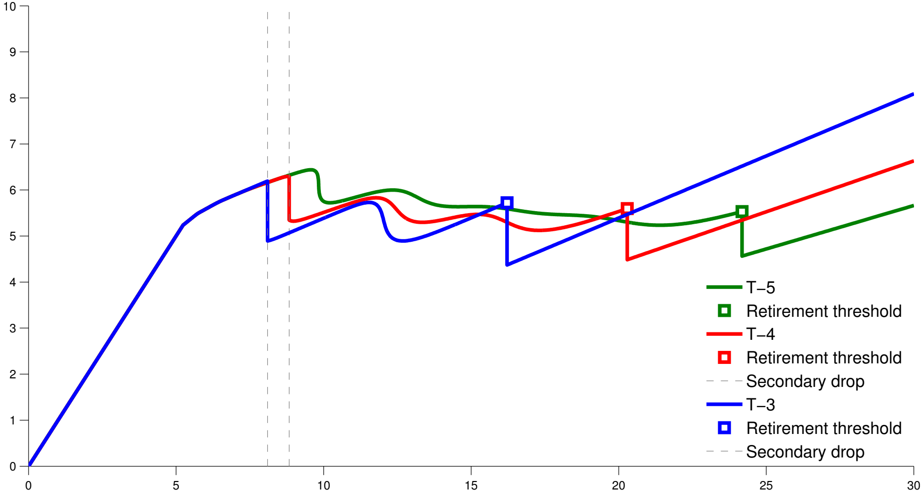
EGDST implementation (diff)

```
model.setparam('sig',0.1);
model.solve
model.plot1('c','it=3','ist=2');
```

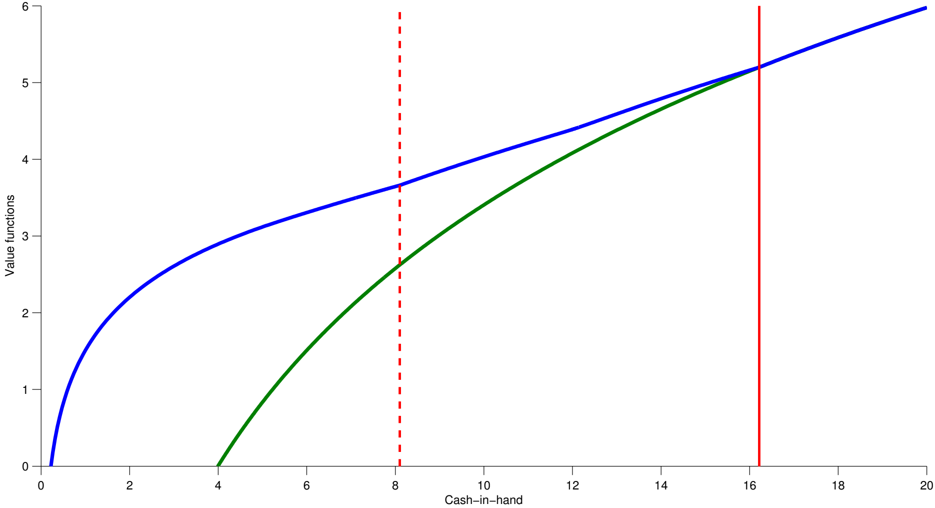
Period $T-3$: Optimal consumption with $\sigma = .1$



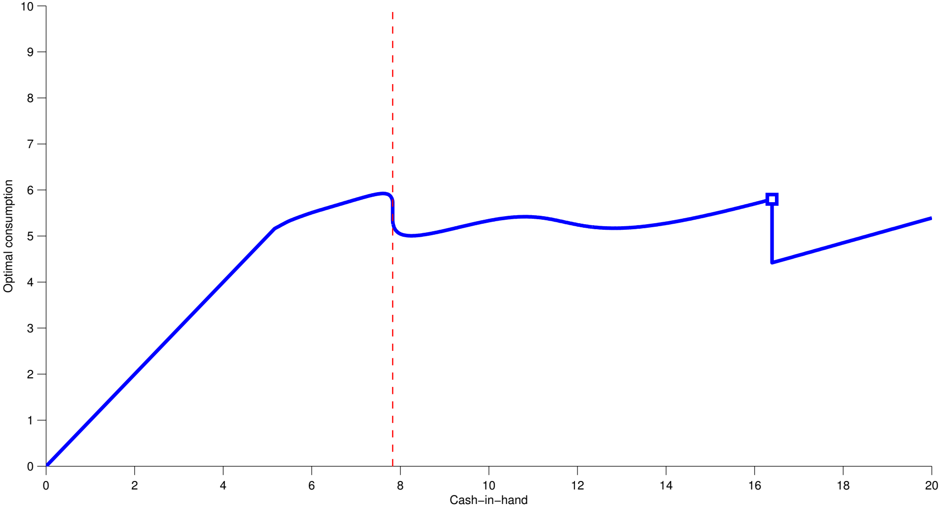
Before $T-3$: Optimal consumption with $\sigma = .1$



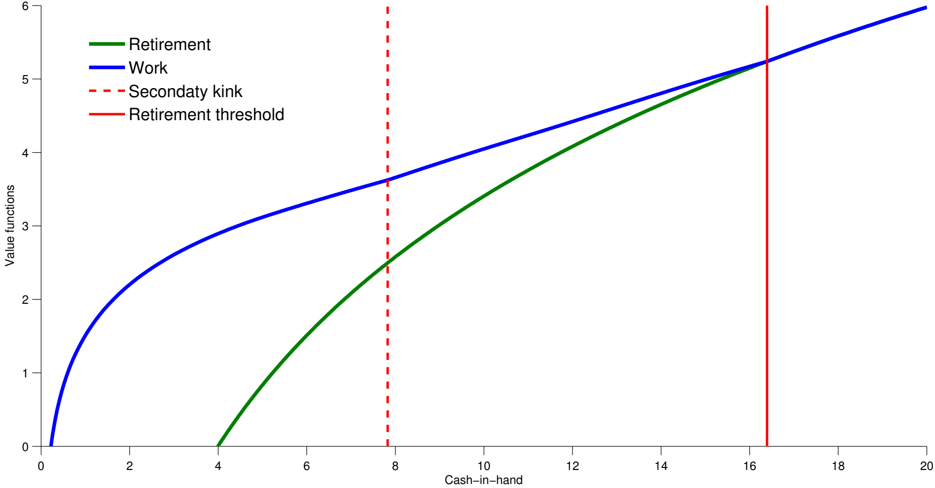
Period $T-3$: VF with $\sigma = .1$



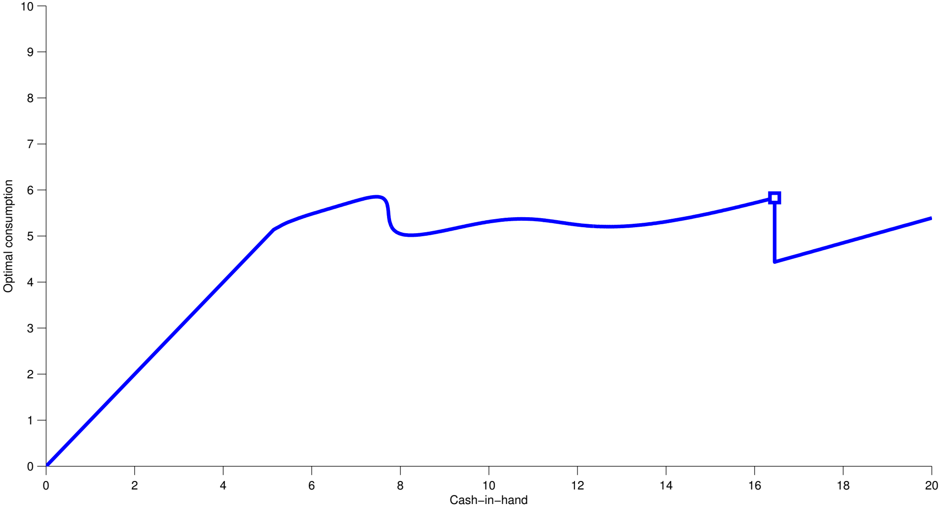
Period $T-3$: Optimal consumption with $\sigma = .2$



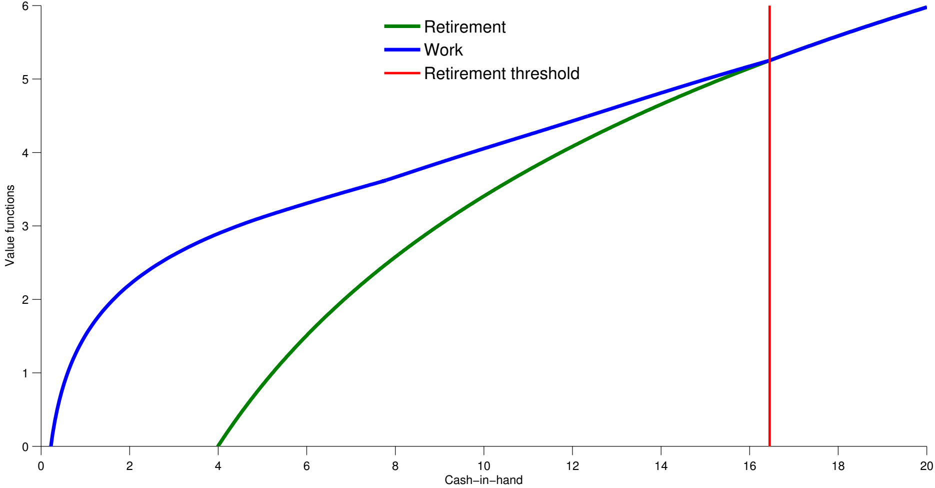
Period $T-3$: VF with $\sigma = .2$



Period $T-3$: Optimal consumption with $\sigma = .22$



Period $T-3$: VF with $\sigma = .22$



Full solution of the simple retirement model

Properties of the full solution

1. When R is deterministic discontinuities/kinks propagate through time and multiply
2. Stochastic $\tilde{R}$ (shocks in general) *may or may not* smooth out the *secondary* kinks

Credit constraints

- Credit constraints are handled so well by EGM because it is never necessary to compute utility of nearly zero consumption
- Instead we "connect the dots" $(0,0)$ and (M_t^{cc}, M_t^{cc})

  M_t^{cc} — level of wealth corresponding to $A_t = 0$
- Inevitable when value functions have to be computed
- If utility is additively separable in consumption and discrete choices (AS), the problem can be avoided almost entirely!

Credit constraints

Dealing with credit constraints

1. For each d_t compute M^{cc}_{t,d_t} correspond to zero savings
   EGM loop can be started from $A = 0$
   $M^{cc}_{t,d_t} : \quad \forall M < M^{cc}_{t,d_t} \quad c^{\star}_t = M$

Credit constraints

Dealing with credit constraints

1. For each d_t compute M^{cc}_{t,d_t} correspond to zero savings
   EGM loop can be started from $A = 0$
   $M^{cc}_{t,d_t} : \ \forall M < M^{cc}_{t,d_t} \ \ c^{\star}_t = M$

2. Value function for $M < M^{cc}_{t,d_t}$ has analytic form
   $V^{d_t}_t(M) = u(M, d_t) + \beta EV^0_{t+1}(d_t)$
   $EV^0_{t+1}(d_t)$ — expected value of ending period t with $A_t = 0$

Credit constraints

Dealing with credit constraints

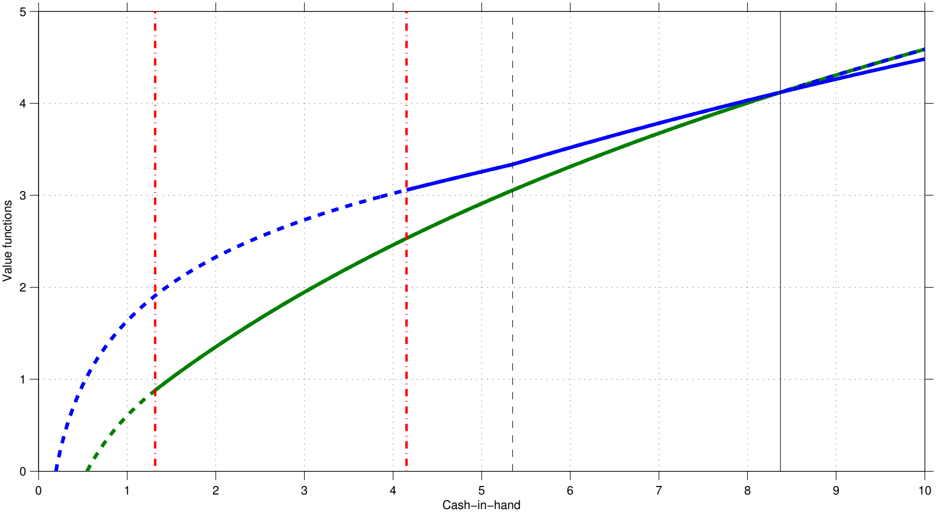
1. For each d_t compute M^{cc}_{t,d_t} correspond to zero savings
   EGM loop can be started from $A = 0$
   M^{cc}_{t,d_t} : $\forall M < M^{cc}_{t,d_t}$ $c^\star_t = M$
2. Value function for $M < M^{cc}_{t,d_t}$ has analytic form
   $V^{d_t}_t(M) = u(M, d_t) + \beta EV^0_{t+1}(d_t)$
   $EV^0_{t+1}(d_t)$ — expected value of ending period t with $A_t = 0$
3. (AS) $\Rightarrow V^{d_t}_t(M) = u(M) + v(d_t) + \beta EV^0_{t+1}(d_t)$

Credit constraints

Dealing with credit constraints

1. For each d_t compute M^{cc}_{t,d_t} correspond to zero savings
   EGM loop can be started from $A = 0$
   M^{cc}_{t,d_t} : $\forall M < M^{cc}_{t,d_t} \quad c^\star_t = M$
2. Value function for $M < M^{cc}_{t,d_t}$ has analytic form
   $V^{d_t}_t(M) = u(M, d_t) + \beta EV^0_{t+1}(d_t)$
   $EV^0_{t+1}(d_t)$ — expected value of ending period t with $A_t = 0$
3. (AS) $\Rightarrow V^{d_t}_t(M) = u(M) + v(d_t) + \beta EV^0_{t+1}(d_t)$
4. $V^{d_t}_t(M)$ do not intersect when $M < \min_{d_t}\left\{M^{cc}_{t,d_t}\right\}$
   $\Rightarrow$ No need to compute utility of nearly zero consumption

Pension benefit $.25y$



What to do with EGM methods

We can solve many problems of this type $\Rightarrow$

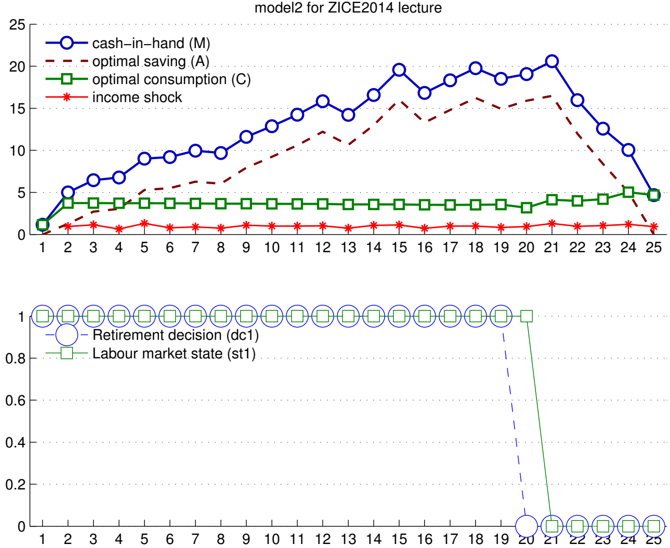
1. Fast solver for important problems with discrete/continuous choice $\rightarrow$
   - calibration
   - structural estimation with your favourite method
2. Use the solver repeatedly in some "outer loop" $\rightarrow$
   - individual heterogeneity : solve the model for each individual in the sample
   - unobserved heterogeneity : random effects
   - flexibility of distributional assumptions

Simulations : code

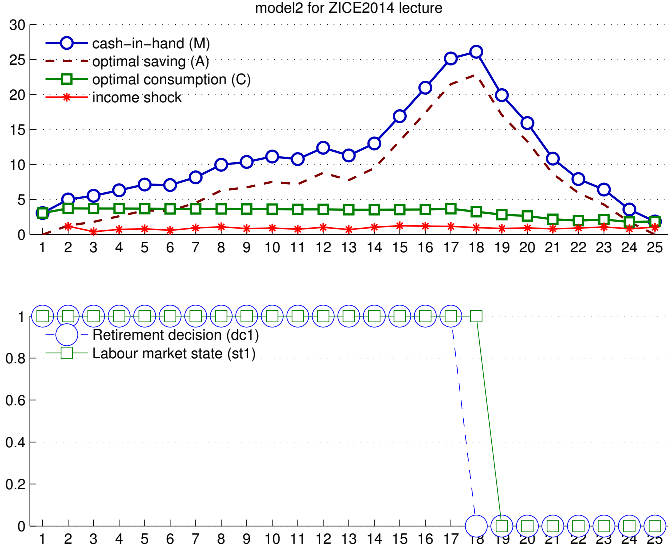
```
%randstream.mat is read if available
%individual or common shocks
model.sim(INITCOND,'own_shocks');

%plots
model.plot2('mack-ds','sims=1:2');
model.plot2('m','sims=1:100','combine');
model.plot2('c','sims=1:100','combine');
model.plot2('d1','sims=1:100','combine');
```

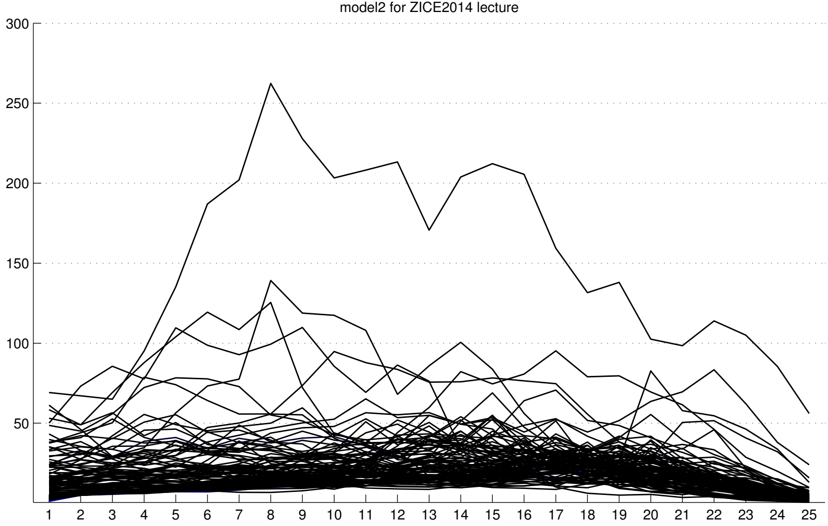
Simulations : individual 1



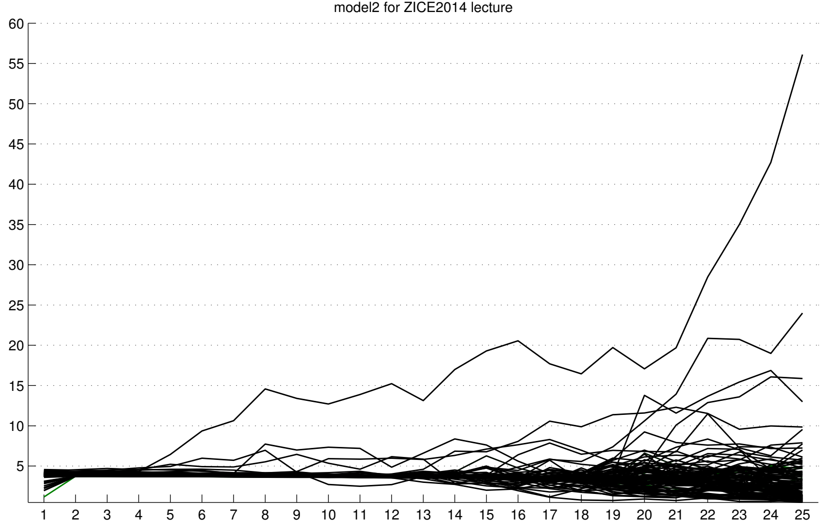
Simulations : individual 2



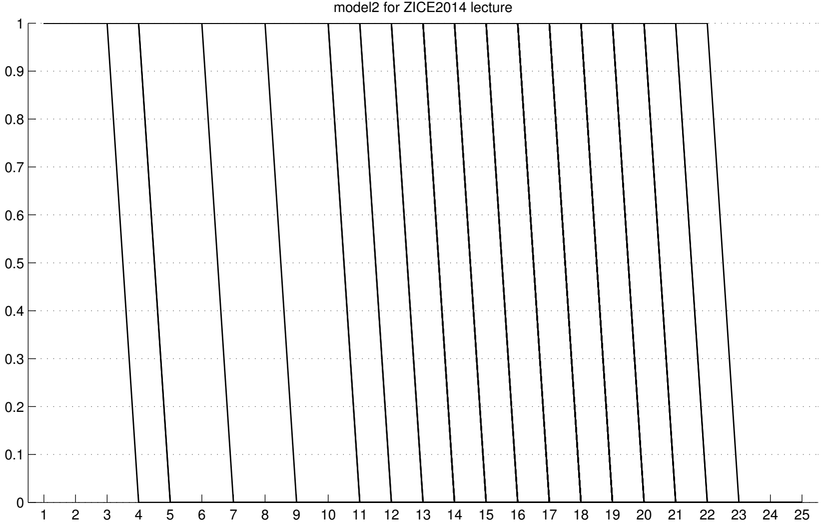
Simulations : wealth



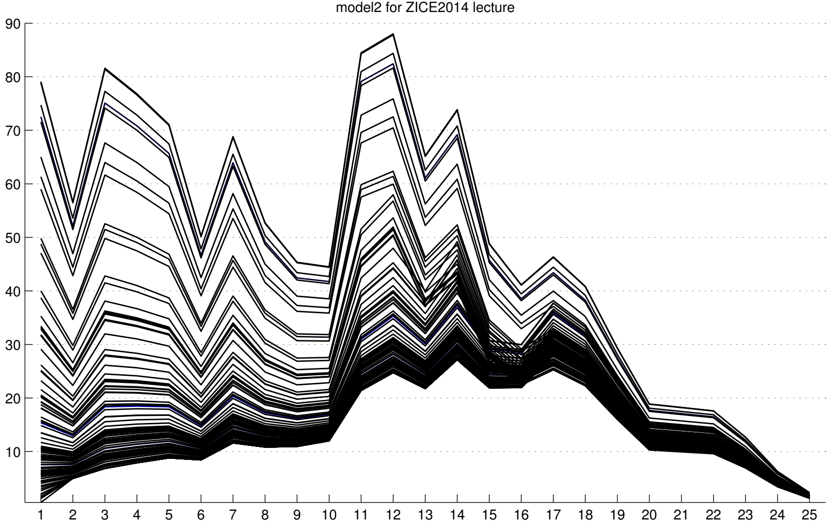
Simulations : consumption



Simulations : retirement choices



Simulations : common shocks



Conclusions

1. EGM is applicable to discrete-continuous problems
2. Care has to be taken in non-concave regions
3. With additively separable utility generalized EGM is also very efficient with credit constraint
4. EGM is faster and more accurate than traditional methods for solving lifecycle models
5. Easy to use open-source software package

github.com/fediskhakov/egdst

Future research

1. Still working on assessing the error bounds of the solution produced by generalized EGM
2. Generalized EGM is not hard to parallelalize, OpenMP and MPI parallelized versions are on the way
3. Error reporting and diagnostics will be in better shape in future versions

Work in progress, updates at

github.com/fediskhakov/egdst

Exercises

1. Download the code from and reproduce the graphs in these slides
2. Solve the consumption/savings model in infinite horizon
3. Compare the EGM solution in (3) with value function iterations and policy iterations
4. Add education to the retirement model so that wage incomes varies by education, discuss the differences in labour supply decisions in this model
5. Add part time work decision in the retirement model and simulate the case of phased retirement
6. Use EGDST software in your research