Fast upper-envelope scan for discrete-continuous dynamic programming

Isabella Dobrescu[1] and Akshay Shanker[2]

31 August 2022

[1]University of New South Wales
[2]University of Sydney, University of New South Wales and CEPAR

Main contribution

Endogenous grid method (EGM) $\cap$ discrete-continuous problems

$\Rightarrow$ FOCs not sufficient

$\Rightarrow$ Value correspondence contains sub-optimal points on a non-uniform grid

Main contribution

Scan method (FUES) to compute upper-envelope of EGM value correspondence

Structure of talk

1. Introduction, motivation and literature ‣ Introduction
2. Illustrative application ‣ Application
3. Theoretical foundations ‣ Theory
4. Extensions to theory ‣ Extensions
5. Concluding remarks ‣ Conclusion

Introduction

Dynamic programming

Dynamic programming (now a.k.a applied variational analysis) key for:

- Economic dynamics: macro and micro structural
- Artificial intelligence
- Operations research (guidance systems, flight scheduling)
- Finance and dynamic portfolio allocation
- Angst, free-will, non-being, the abyss etc.

Value function iteration

Generic method to solve (primitive form) DP problem

$$V_t(x) = \max_{c \in \Omega} \underbrace{u_t(c) + V_{t+1}(f(x, c))}_{\text{Solve numerically}}$$

Where:

- V_t is time t value function
- u_t is time t pay-off
- f is a transition function
- The term c, $c \in A \subset \mathbb{R}^K$, is a control and Ω is a constraint
- The term x, $x \in S \subset \mathbb{R}^n$, is the endogenous state (assume no shocks for now)

Curse of dimensionality

What happens when $x \in \mathbb{R}^n$, where n is ‘big’?

- Curse of dimensionality

Curse of dimensionality

The curse relevant for 'real world' quantitative applications of economic models

Simple life-cycle model capturing pension and housing

- Liquid assets
- Housing assets*
- Pension balance
- Mortgage balance
- Rental decision*

Endogenous states

- Portfolio* and pension plan* choice
- Contribution choice*

Additional controls

- Wage, preference and price shocks

* – discrete choice controls

Value function iteration

VFI too expensive for applied models with some dimensionality

Obvious solution to use first order information

- Endogenous grid method

Endogenous grid method

Assume differentiability, let $\partial u_t(c)$ be the Gateaux differential at c

Let σ_t be the time t policy

Given σ_{t+1} and x, interior solution c will satisfy:

$$\partial u_t(c) = \partial u_{t+1}(\sigma_{t+1}(f(x, c)))$$

Double curse of dimensionality and the great watershed

Let $\bar{f}$ denote the inverse of f in the first argument and assume $c \mapsto \partial u_t(c)$ is analytically invertible (see Iskhakov 2015):

$$x = \bar{f}(x', \partial u_t^{-1}(\partial u_{t+1}(\sigma_{t+1}(x'))))$$

1. If we have σ_{t+1}, then make a uniform grid of $\hat{x}'$ values
2. Analytically compute endogenous grid of $\hat{x}$ along with $\hat{c}$ values
3. Approximate time t policy function

Double curse of dimensionality and the great watershed

Introduce a discrete choice

Each choice $\mathbf{d}$ yields a future-choice specific value function $V^{\mathbf{d}}_{t+1}$, where

$$V_{t+1}(x) = \max_{\mathbf{d}} V^{\mathbf{d}}_{t+1}(x), \qquad \forall x \in S$$

Define

$$Q(c, x) := u(c) + \underbrace{\max_{\mathbf{d}}}_{\neg \text{ preserve `convexity'}} V^{\mathbf{d}}_{t+1}(f(c, x))$$

$Q(c, x)$ not concave in c!

Double curse of dimensionality and the great watershed

FOC not sufficient!

- Some values $\hat{x}$, $\hat{x}'$ will not be optimal
- recall $\hat{x}$ is not uniform

Our contribution

- recover the upper-envelope of optimal EGM ppints using a scan method

Related work

Upper-envelope construction not new

Iskhakov et al. 2017 construct upper-envelope by identifying monotone segments of the policy function and interpolating the value function on each segment

- extends earlier work by Fella 2014
- monotonicity assumption (?)

Our contribution:

- FUES does not rely on monotonicity (easy to implement under non-monotonicity. Relevant for applications where hard to check monotonicity with K different discrete choices)
- We give a proof that FUES can recover the optimal points if grid-size is large enough
- ....towards theoretical and geometric foundations for identifying the upper-envelope

Related work

FUES inspired by Graham scan (Graham 1972) to compute convex hulls

Intimate relationship of discrete-continuous problems to difference of convex functions (DC) optimization problems and mixed integer non-linear programming (MINLP)

- See Jeyakumar and Srisatkunarajah 2009
- So far, necessary and sufficient SOCs and FOCs seem elusive

‘Attempts’ to solve the general case

Illustrative application

Retirement choice model

Model considered by Iskhakov et al. 2017

- Time starts at $t = 0$
- Agents live, work (if they so choose) and consume until time $t = T$
- Each period, the agent starts as a worker or retiree, denoted by d_t
- If the agent works, they earn at wage y
- Agents can continue to work during the next period by setting $d_{t+1} = 1$, or they permanently exit the workforce by setting $d_{t+1} = 0$
- If the agent chooses to work the next period, they will incur a utility cost δ
- Agents consume c_t and save in capital a_t, with $a_t \in \mathbb{A}$ and $\mathbb{A} := [0, \bar{a}] \subset \mathbb{R}_+$

Retirement choice model

The intertemporal budget constraint:

$$a_{t+1} = (1+r)a_t + d_t y - c_t$$

Utility in each period is given by:

$$\log(c_t) - \delta d_{t+1}$$

Let the function u be defined by:

$$u(c) = \log(c)$$

Maximisation problem

The agent's sequential maximisation problem becomes:

$$V_0^{d_0}(a_0) = \max_{(c_t, d_t)_{t=0}^{T}} \sum_{t=0}^{T} \beta^t \left(u(c_t) - \delta d_{t+1} \right)$$

subject to:

- budget constraint
- $a_t \in \mathbb{A}$ for each t
- agent cannot return to work after retiring, $d_{t+1} = 0$ if $d_t = 0$

Bellman equation

Worker recursive value function can be characterised by the Bellman Equation:

$$V_t^1(a) = \max_{c, d' \in \{0,1\}} u(c) - d'\delta + \beta V_{t+1}^{d'}(a')$$

where $a' = (1+r)a + y - c$ and such that $a' \in \mathbb{A}$

Retiree value function:

$$V_t^0(a) = \max_c u(c) + \beta V_{t+1}^0(a')$$

with $a' = (1+r)a - c$

Non-convexity

Worker optimisation problem not concave since the worker optimizes jointly a discrete choice and a continuous choice

Even conditioned on $d' = 1$, the the next period value function, V^1_{t+1}, will not be concave

The value function represents the supremum over all future feasible combinations of discrete choices

- ‘secondary kinks’ described by Iskhakov et al. 2017

Non-convexity

Write the time t worker's value function as:

$$V_t^1(a) = \max_{c} \max_{\mathbf{d} \in \mathbb{D}} u(c) - d'\delta + \beta Q_{t+1}^{\mathbf{d}}(a')$$

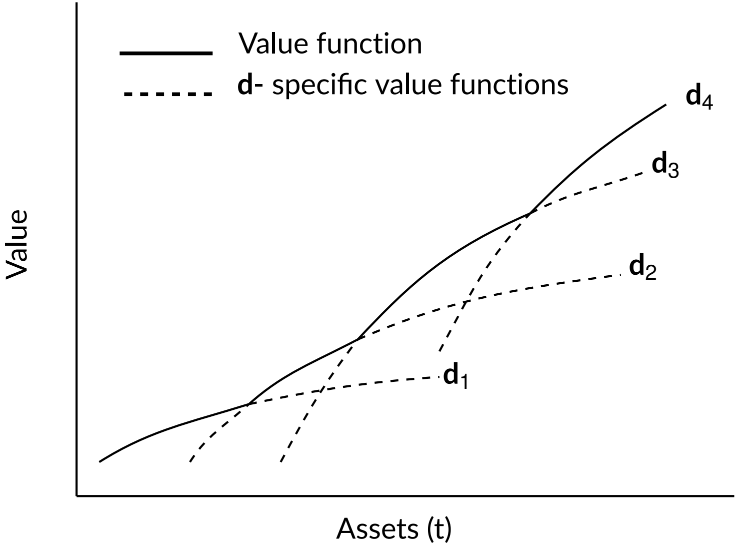
where $Q_{t+1}^{\mathbf{d}}$ is the $t+1$ value function conditioned on a given sequence of future discrete choices $\mathbf{d}$

- We have $\mathbf{d} = \{d', d'', \ldots\}$
- The set $\mathbb{D}$ contains all feasible sequences of discrete choices from t to T

V_t^1 will be the upper envelope of overlapping concave functions

- each concave function corresponding to a different sequence of future discrete choices

Non-convexity



Neccesary Euler equation

Let $\sigma_t^d \colon \mathbb{A} \times \{0,1\} \to \mathbb{R}_+$ be the conditional asset policy function for the worker at time t

- Worker if $d = 1$ and retiree if $d = 0$
- Policy depends, through its second argument, on the discrete choice (to work or not to work in $t+1$)

Functional recursive Euler equation:

$$u'((1+r)a + dy - \sigma_t^d(a, d')) \geq \beta(1+r)u'((1+r)\sigma_t^d(a, d') + d'y - \sigma_{t+1}^{d'}(a', d''))$$

where $a' = \sigma_t^d(a, d')$

Work choice

The time t worker will chose $d_{t+1} = 1$ if and only if:

$$u((1+r)a + y - \sigma_t^1(a,1)) - \delta + \beta V_{t+1}^1(\sigma_t^1(a,1)) \\ > u((1+r)a - \sigma_t^1(a,0)) + \beta V_{t+1}^0(\sigma_t^1(a,0))$$

Define a discrete choice policy function $\mathcal{I}_t \colon \mathbb{A} \times \{0,1\} \to \{0,1\}$

We will have $d' = \mathcal{I}_t(a,d)$ and $d'' = \mathcal{I}_{t+1}(a', d')$

Euler equation not sufficient

All work choices need to be selected at the same time

Sequence satisfying Euler equation sufficient given work choices, but recursively chosen sequence of work choices may not be optimal

FUES-EGM

Fix a time t and suppose we know:

- The value function V^d_{t+1}
- Optimal policy function σ^d_{t+1}

Set an exogenous grid $\hat{\mathbb{X}}'_t$, we will say $\hat{x}'_i \in \hat{\mathbb{X}}'_t$ (note the i subscript) such that:

$$\hat{\mathbb{X}}'_t = \{\hat{x}'_0, \hat{x}'_1, \ldots, \hat{x}'_i, \ldots, \hat{x}'_N\}$$

FUES-EGM

Let $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ be sequences of points (1D grids) satisfying the Euler equation for workers:

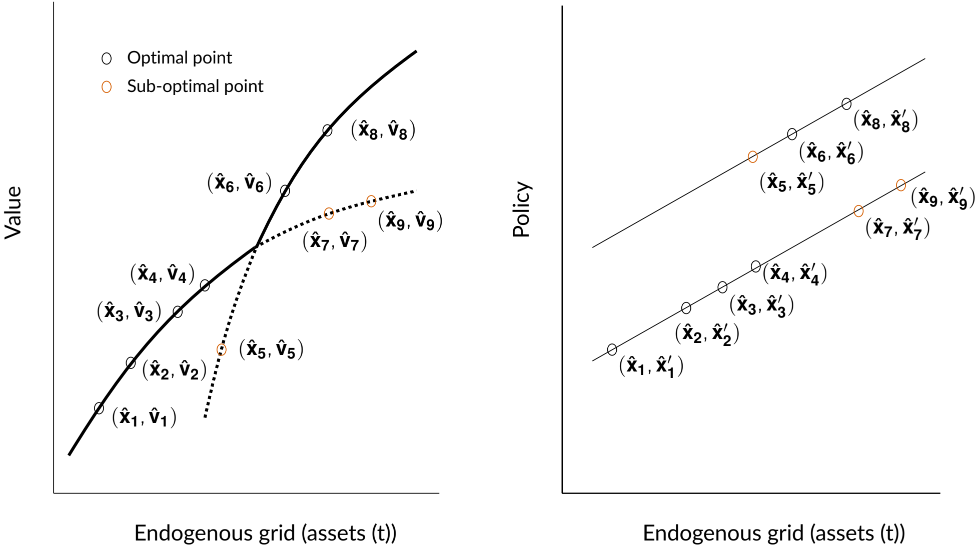
$$u'((1+r)\hat{x}_i + dy - \hat{x}'_i) = \beta(1+r)u'((1+r)\hat{x}'_i + yd' - \sigma^{d'}_{t+1}(\hat{x}'_i, d''))$$

$$\hat{v}_i = u(\hat{c}_i) - d\delta + V^{d}_{t+1}(\hat{x}_i)$$

- Generate using EGM
- Order the sequence of points according to the endogenous grid of points $\hat{\mathbb{X}}_t$

(*Interior solution only, follow Iskhakov et al. 2017 occasional binding constrained policy*)

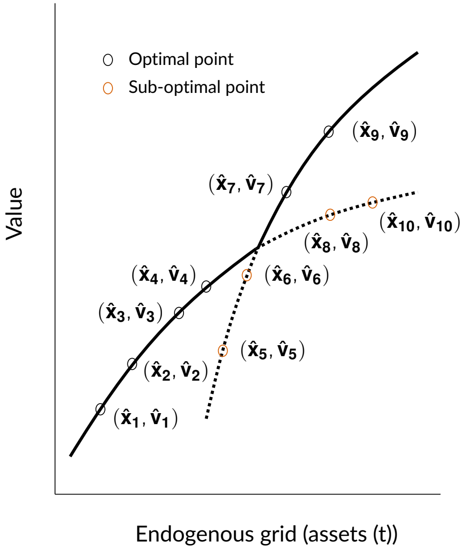
FUES-EGM



Optimal point
Sub-optimal point
$(\hat{x}_8, \hat{v}_8)$
$(\hat{x}_6, \hat{v}_6)$
$(\hat{x}_9, \hat{v}_9)$
$(\hat{x}_7, \hat{v}_7)$
$(\hat{x}_4, \hat{v}_4)$
$(\hat{x}_3, \hat{v}_3)$
$(\hat{x}_5, \hat{v}_5)$
$(\hat{x}_2, \hat{v}_2)$
$(\hat{x}_1, \hat{v}_1)$
Value
Endogenous grid (assets (t))
$(\hat{x}_8, \hat{x}'_8)$
$(\hat{x}_6, \hat{x}'_6)$
$(\hat{x}_5, \hat{x}'_5)$
$(\hat{x}_9, \hat{x}'_9)$
$(\hat{x}_7, \hat{x}'_7)$
$(\hat{x}_4, \hat{x}'_4)$
$(\hat{x}_3, \hat{x}'_3)$
$(\hat{x}_2, \hat{x}'_2)$
$(\hat{x}_1, \hat{x}'_1)$
Policy
Endogenous grid (assets (t))

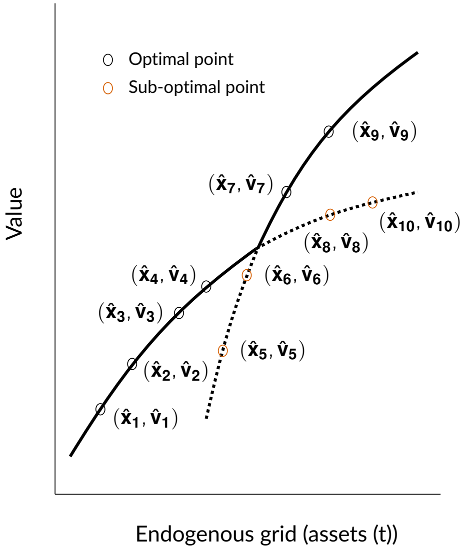
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM



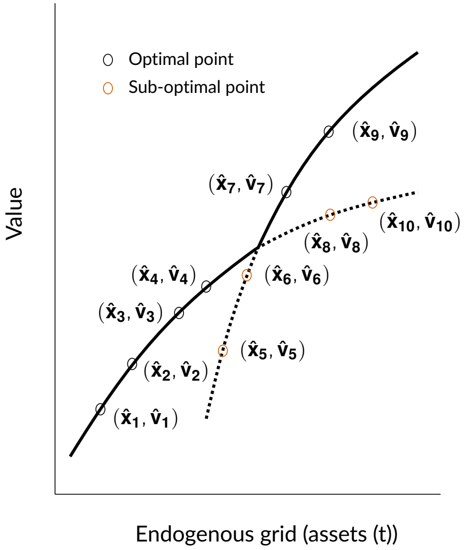
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}_t'$ using EGM
2. Set 'jump detection' threshold $\bar{M}$



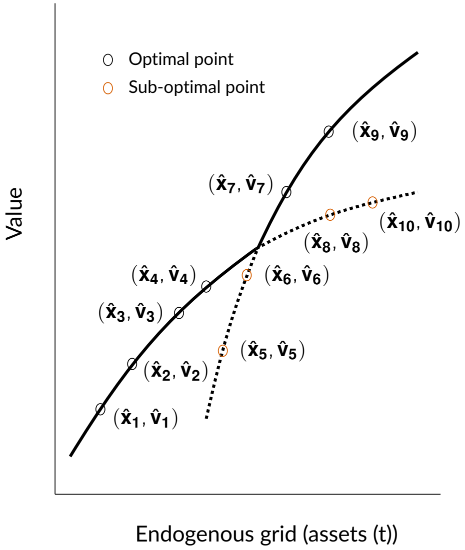
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$



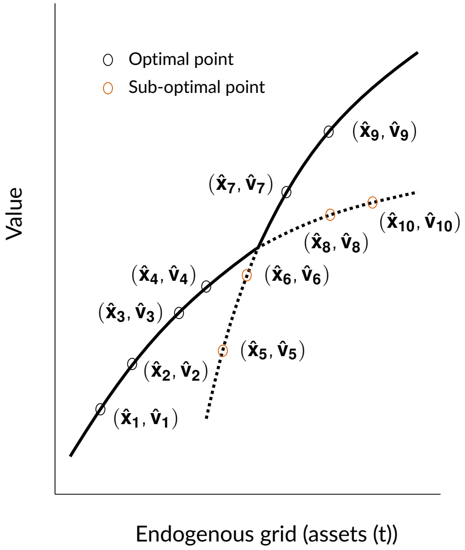
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$



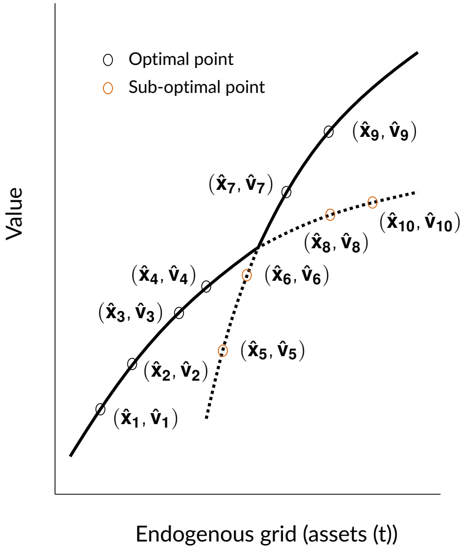
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$



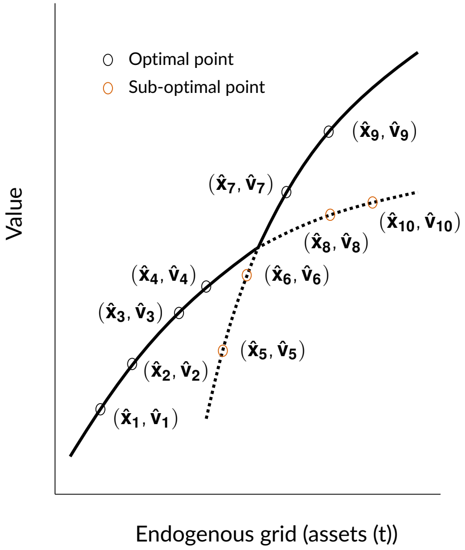
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $\left|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}\right| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$



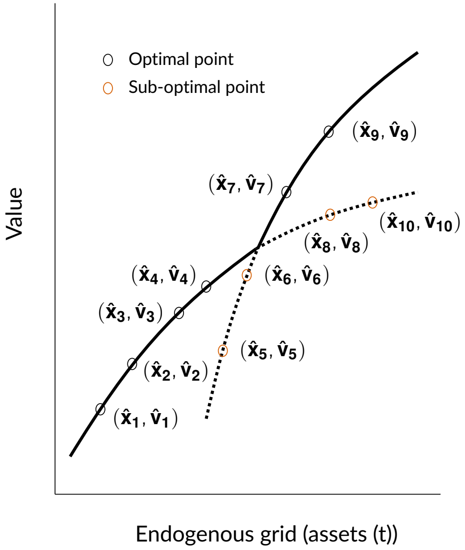
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



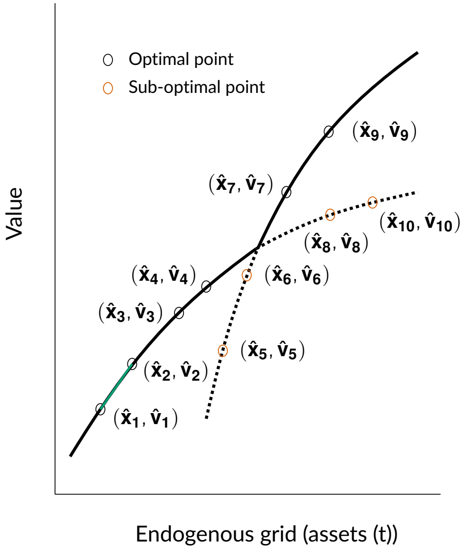
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i + 1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



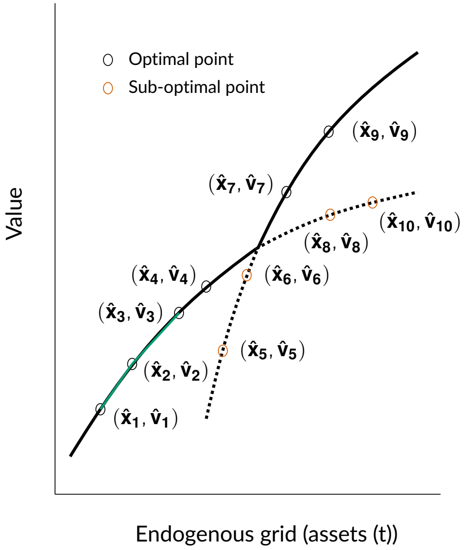
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i + 1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



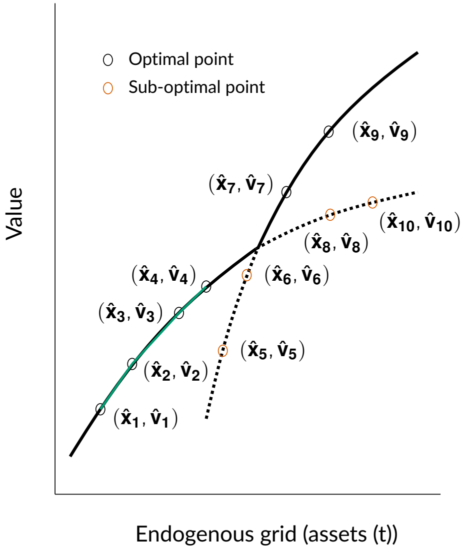
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $\left|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}\right| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i + 1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



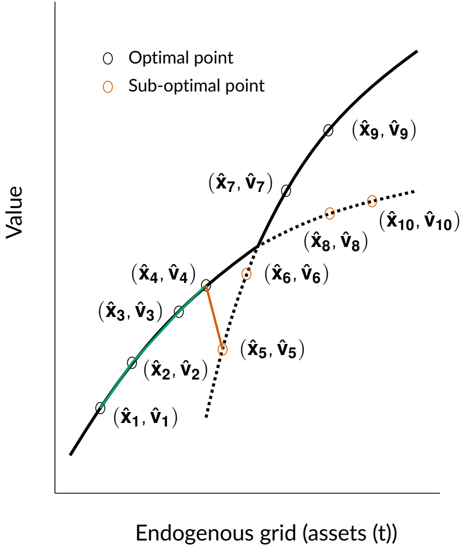
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn $(g_{i+1} < g_i)$, then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



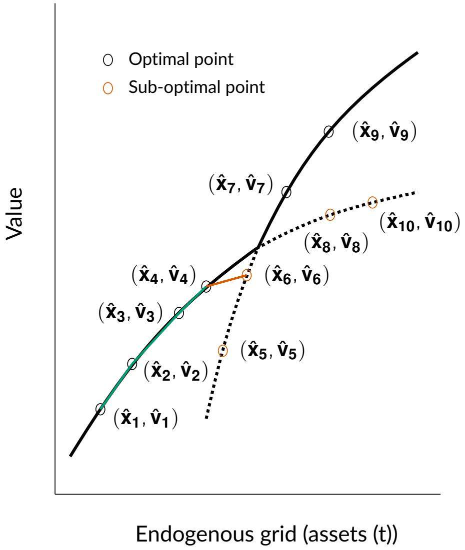
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



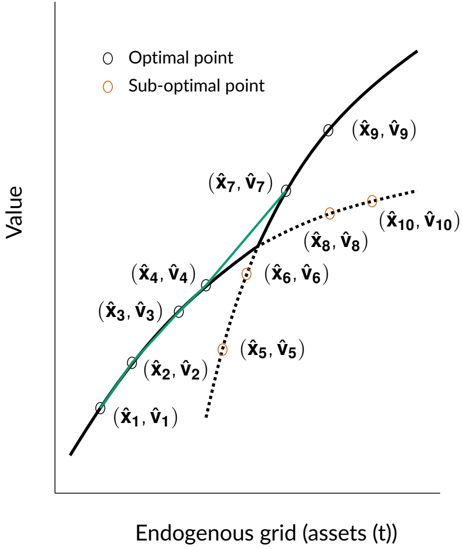
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $\left|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}\right| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



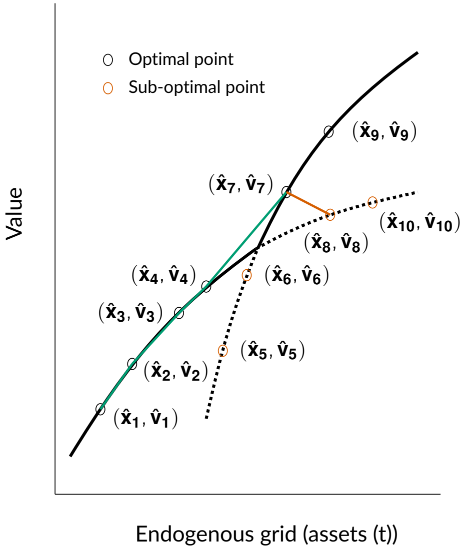
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



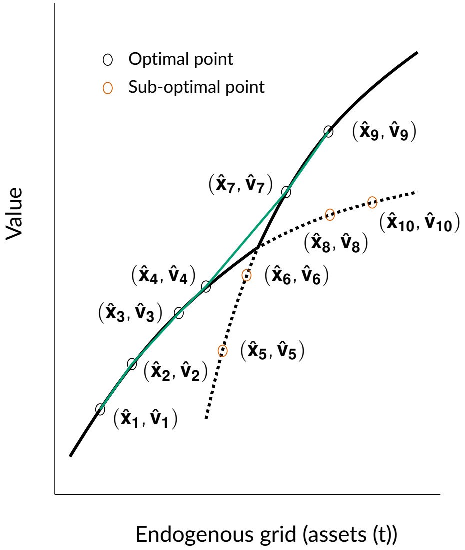
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $\left|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}\right| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



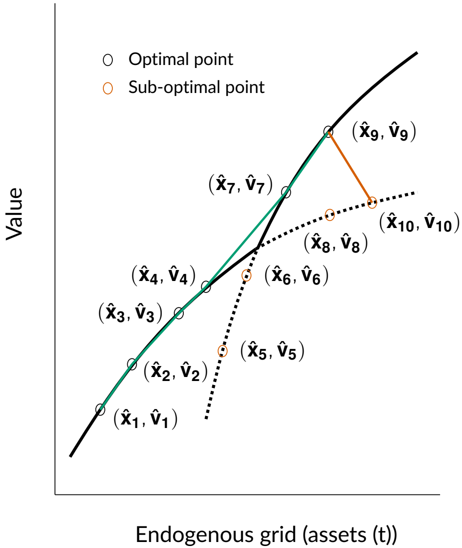
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i + 1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5



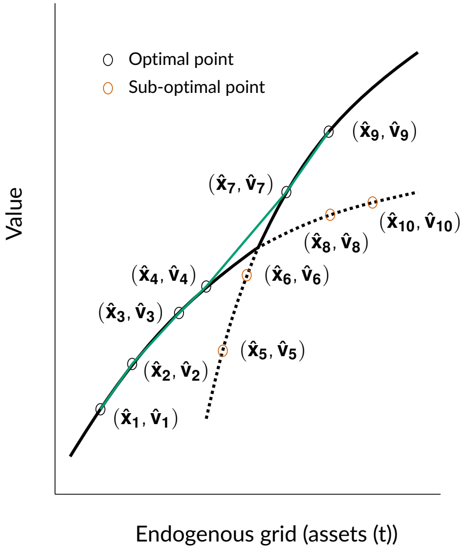
FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $\left|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}\right| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i+1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5

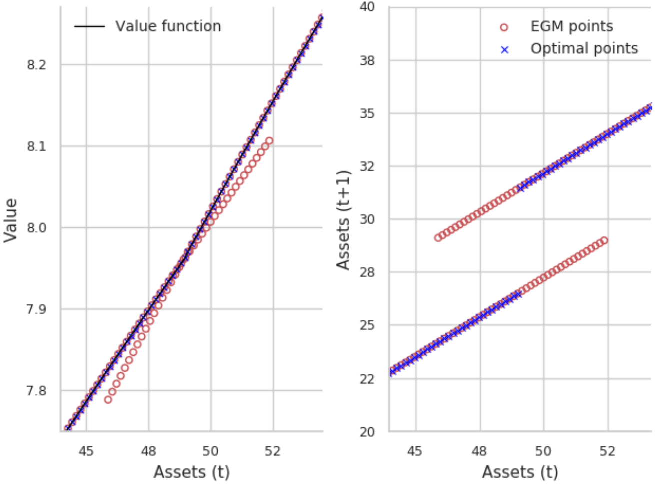


FUES-EGM

1. Compute $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$ using EGM
2. Set 'jump detection' threshold $\bar{M}$
3. Sort all in order of *endogenous* grid $\hat{\mathbb{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i - \hat{x}_{i-1}}$ and $g_{i+1} = \frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i}$
6. If $|\frac{\hat{x}'_{i+1} - x'_i}{\hat{x}_{i+1} - \hat{x}_i}| > \bar{M}$ and right turn ($g_{i+1} < g_i$), then remove point $i + 1$ from grids $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}'_t$
   - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbb{X}}_t|$, then repeat from step 5

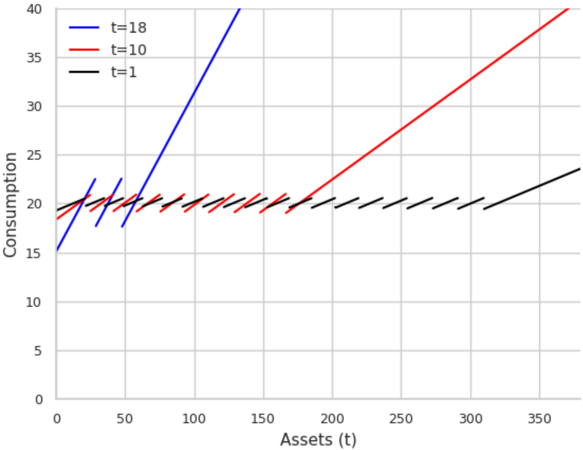


FUES-EGM



Value function
8.2
8.1
8.0
7.9
7.8
Value
45
48
50
52
Assets (t)
EGM points
Optimal points
40
38
35
32
30
28
25
22
20
Assets (t+1)
45
48
50
52
Assets (t)

FUES-EGM



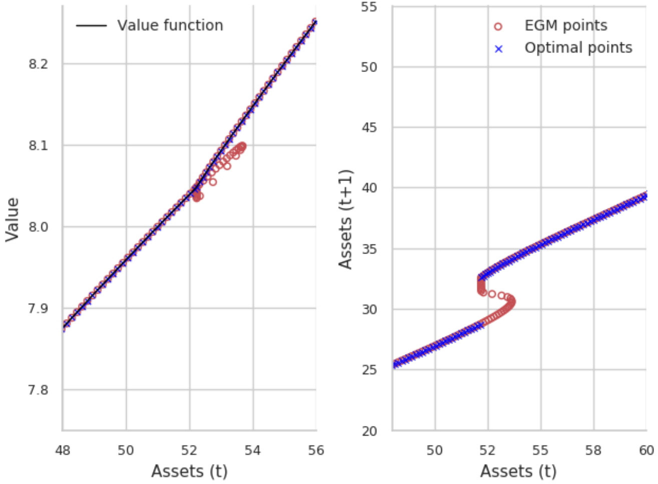
t=18
t=10
t=1
Consumption
Assets (t)
0
5
10
15
20
25
30
35
40
0
50
100
150
200
250
300
350

Uncountably many future choices

Formal theory of FUES (so far) relies on:

- A large enough jump associated with a change in future discrete choices
- Not the case when there are un-countably many choices
- In practice FUES method also works with more than finitely many future choices

Retirement choice model with smoothing



Forward scan

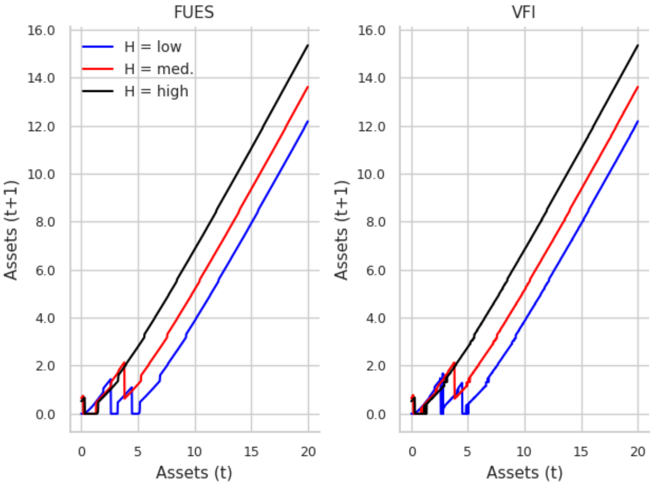
Discussion so far requires first point after a cross-point to make a left turn:

- May not be the case. Solution: scan forward to see if the point is dominated before elimination

There may not be points from the upper-envelope after a cross-point:

- More serious issue, will affect all upper-envelope algorithms
- Theory so far assumes first point after crossing point is on the upper-envelope

Discrete housing choice model (Fella, 2014)



Theoretical foundations

Intuition

Proof for FUES needs to distinguish between a 'jump' in the policy function (which can only occur at a convex region) and a continuous movement along the policy function (which can occur at concave regions of the value function)

We will need:

- Policy functions need to have a common bound on derivative
- Jump sizes to be large enough
- Subset of optimal endogenous points need to be close enough
  - $\Rightarrow$ difference quotient at jump $\rightarrow \infty$

Assumptions

Consider introductory set-up with finitely many future-specific choices $\mathbb{D}$

- One dimension, $\geq 0x \leq \bar{K}$

(Remove time subscripts)

Let $\mathbb{X}^\star$ be the set of optimal points and let T_P be the set of cross-points

Assumption

The function $f : \mathbb{R}^2_+ \to \mathbb{R}$ is invertible, smooth and monotone.

Assumptions

Assumption

The term $|f(\sigma_i(x), x) - f(\sigma_j(x), x)|$ *is bounded below by a constant* D *for all* $i, j \in \mathbb{D}$, $i \neq j$ *and* $x \leq \bar{K}$

Assumptions

Assumption

The family of functions $x \mapsto f(\sigma_i(x), x)$ *for* $i \in \mathbb{D}$ *have a common Lipshitz constant* M

Assumptions

Assumption

There exists $\delta > 0$ such that for all $j \leq |\mathbb{X}^\star|$, $|x^\star_{j+1} - x^\star_j| \leq \delta$ and $\frac{D}{\delta} > 2M$

Jump detection threshold becomes

$$\frac{D}{\delta} - M$$

Assumptions

Assumption

Fix $\tilde{x} \in T_P$ and let $x_k^\star$ be the largest element in $\mathbb{X}^\star$ such that $x_k^\star \leq \tilde{x}_i$ and $x_{k+1}^\star$ be the smallest element in $\mathbb{X}^\star$ such that $x_{k+1}^\star \geq \tilde{x}$. The following hold:

1. *If $v_{k+1}^\star = Q^l(x_{k+1}^\star)$ for some l, then $v_{k+2}^\star = Q^l(x_{k+2}^\star)$*
2. *If $v_k^\star = Q^m(x_k^\star)$ for some m, then $v_{k-1}^\star = Q^m(x_{k-1}^\star)$*
3. *If $\hat{x}_{j+1}$ is the smallest element in $\hat{\mathbb{X}}_t$ such that $\hat{x}_{j+1} \geq \tilde{x}_i$, then $\hat{v}_{j+1} = Q^l(\hat{x}_{j+1})$ where l is the value function crossing at point $\tilde{x}_i$ from below*

Left turn implies optimal

The above Assumption and concavity of the future-choice specific value functions gives

Claim

Fix the triple $\hat{x}_i, \hat{x}_{i+1}, \hat{x}_{i+2}$ for some i and assume $\hat{x}_i, \hat{x}_{i+1} \in \mathbb{X}^\star$. If we have:

$$\frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i} < \frac{\hat{v}_{i+2} - \hat{v}_{i+1}}{\hat{x}_{i+2} - \hat{x}_{i+1}}$$

then $\hat{x}_{i+2} \in \mathbb{X}^\star$.

Main result

Proposition

Let Assumptions 1 to 4 hold and let $(\mathbb{X}, \mathbb{X}', \mathbb{C}, \mathbb{V})$ *be the tuple of outputs of Algorithm 1. If* $V(\hat{x}_i) = Q(\hat{c}_i, \hat{x}_i)$ *for* $i = 1, 2$*, then for each* $i \leq |\hat{\mathbb{X}}|$*:*

1. *If* $\hat{x}_i \in \mathbb{X}$*, then* $Q(\hat{c}_i, \hat{x}_i) = V(\hat{x}_i)$*.*
2. *Conversely, if* $Q(\hat{c}_i, \hat{x}_i) = V(\hat{x}_i)$*, then* $\hat{x}_i \in \mathbb{X}$*.*

Note $\mathbb{X}$ will be a sub-sequence of $\hat{\mathbb{X}}$.

Proof (sketch)

For illustrative purposes, assume $\hat{x}_j$ and $\hat{x}_{j-1}$ are optimal. Now we check whether $\hat{x}_{j+1}$ is optimal

There are two cases:

- The first case is if $\frac{\hat{v}_{j+1}-\hat{v}_j}{\hat{x}_{j+1}-\hat{x}_j} \leq \frac{\hat{v}_j-\hat{v}_{j-1}}{x_j-\hat{x}_{j-1}}$
- the second case is if $\frac{\hat{v}_{j+1}-\hat{v}_j}{\hat{x}_{j+1}-\hat{x}_j} > \frac{\hat{v}_j-\hat{v}_j}{\hat{x}_j-\hat{x}_{j-1}}$

Second case follows immediately from Claim 1

Proof (sketch)

We show if $\hat{x}_{j+1} \in \mathbb{X}$, then $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$

Suppose by contradiction $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) \neq V(\hat{x}_{j+1})$

If $\frac{\hat{v}_{j+1}-\hat{v}_j}{\hat{x}_{j+1}-\hat{x}_j} \leq \frac{\hat{v}_j-\hat{v}_{j-1}}{\hat{x}_j-\hat{x}_{j-1}}$

- If we are making a concave right turn and new point is not optimal, then we must be switching discrete choices
- However, by the reverse triangle inequality and assumption on jump size, we have:

$$\frac{|\hat{x}'_{j+1} - \hat{x}'_j|}{\hat{x}_{j+1} - \hat{x}_j} \geq \left|\frac{D}{\hat{x}_{j+1} - \hat{x}_j} - M\right| \geq \frac{D}{\delta} - M$$

- Implies point is eliminated, contradiction

Proof (sketch)

Next, we show $\hat{x}_{j+1} \in \mathbb{X}$ if $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$

If $\frac{\hat{v}_{j+1} - \hat{v}_j}{\hat{x}_{j+1} - \hat{x}_j} \leq \frac{\hat{v}_j - \hat{v}_{j-1}}{\hat{x}_j - \hat{x}_{j-1}}$

- If we are making a concave right turn and new point is optimal, then we must be on the same discrete choice
- By the Lipzhitz condition and assumption on grid size, we have:

$$\frac{|\hat{x}'_{j+1} - \hat{x}'_j|}{|\hat{x}_{j+1} - \hat{x}_j|} \leq M < \frac{D}{\delta} - M$$

- Implies point is NOT eliminated

Extending the foundations

Extending the foundations

(WIP)

Extensions of the theoretical foundations need to relax key assumptions

1. Allow for continuous change in discrete choices
2. Relax assumption on jump size and grid size
3. Address lack of no upper-envelope points generated using EGM after a cross

First two show promise of being rectified. Third is pernicious, possibly common to any DC-EGM problem

Is FUES picking up a deeper geometric structure?

- Line-point duality

Point-line duality

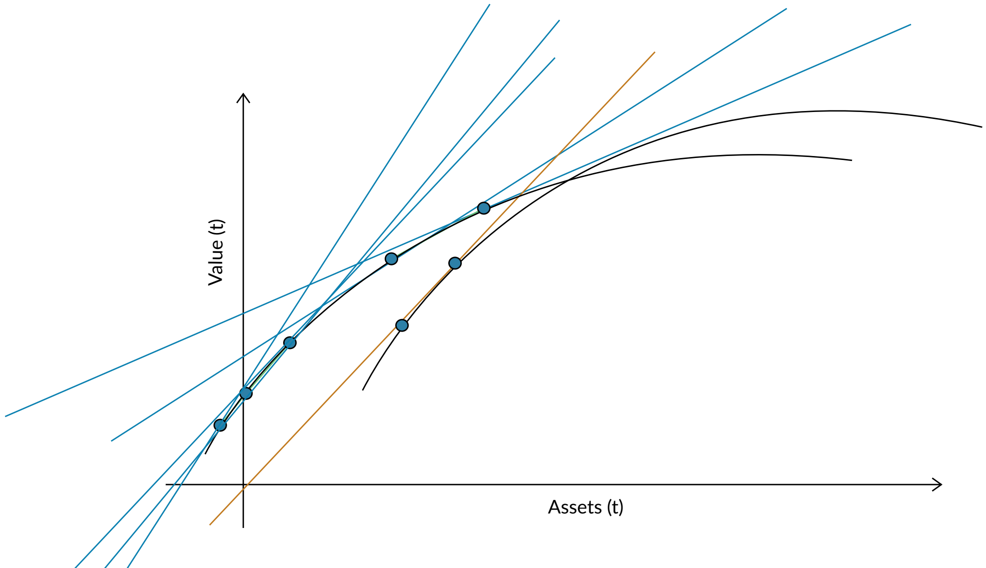
Each line lives in a dual plane to a projective plane

$$y = ax + b \mapsto (a, b)$$

(*Pardon the notation*)

Lower and upper envelope of line intersections in the dual plane are lower and upper convex hulls in the projective plane where the points live

Point-line duality



Conclusion

Concluding remarks and further work

FUES is an easy to code and efficient method to compute the optimal solution for general discrete-continuous dynamic programming problems using EGM

In practice, FUES works for a variety of problems with finite and infinitely many discrete choices

Theoretical results guaranteeing no error depend on assumptions on grid size and jumps between policy functions

- Work in progress to extend the theory using some geometric approaches

Theoretical work may need to focus on error bounds rather than no approximation error conditions

More general further work

1. FOCs and SOCs for box constrained problems are an exciting area of research

2. Under (quasi-)supermodularity, Euler equations will be sufficient
   - What class of dynamic models are supermodular?
   - Is there a tranformation?

3. Convex (biconjugates) relaxations of Hamiltonians
   - Original line of attack
   - Continuous case, see Ekeland and Turnbull 1983
   - Necessity and sufficiency difficult to ‘align’ in discrete time (why?)

Ancient wisdom for modern living

‘...in fact, the great watershed in optimization isn’t between linearity and nonlinearity, but convexity and nonconvexity’ - R.T. Rockafellar