

Fast upper-envelope scan for discrete-continuous dynamic programming

Isabella Dobrescu¹ and Akshay Shanker²

31 August 2022

¹University of New South Wales

²University 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. Concluding remarks ▶ Conclusion

Introduction

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

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** $V_{t+1}^{\mathbf{d}}$, where

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

Define

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

$Q(c, x)$ 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

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:

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

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_{t+1}^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:

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

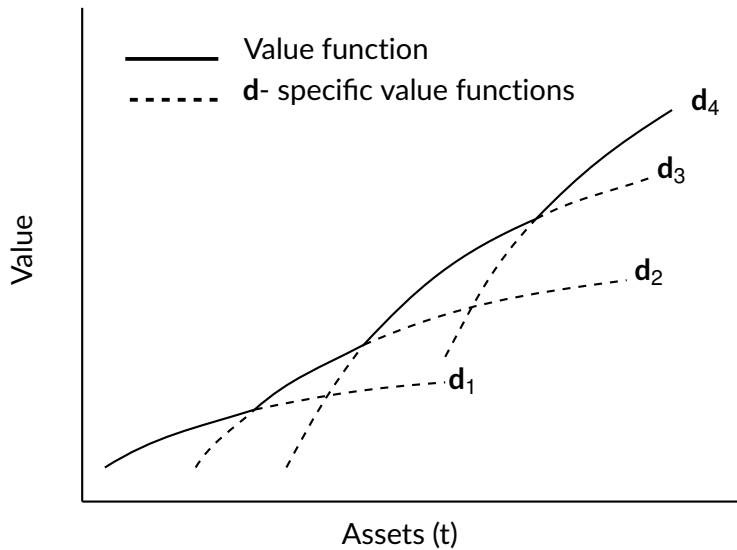
where $Q_{t+1}^{\mathbf{d}}$ is the $t + 1$ value function conditioned on a given sequence of future discrete choices \mathbf{d}

- We have $\mathbf{d} = \{d', d'', \dots\}$
- 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



Necessary Euler equation

Let $\sigma_t^d: \mathbb{A} \times \{0, 1\} \rightarrow \mathbb{R}_+$ 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:

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

Define a **discrete choice policy function** $\mathcal{I}_t: \mathbb{A} \times \{0, 1\} \rightarrow \{0, 1\}$

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

Fix a time t and suppose we know:

- The value function V_{t+1}^d
- Optimal policy function σ_{t+1}^d

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:

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

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:

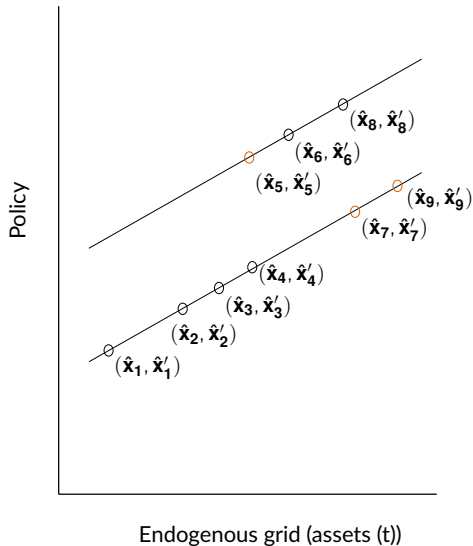
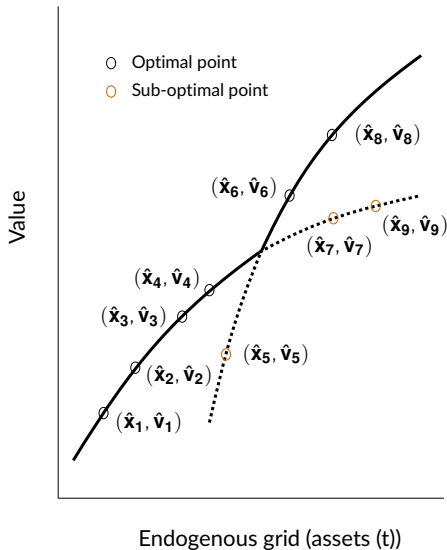
$$u'((1+r)\hat{x}_i + dy - \hat{x}'_i) = \beta(1+r)u'((1+r)\hat{x}'_i + yd' - \sigma_{t+1}^{d'}(\hat{x}'_i, d''))$$

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

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

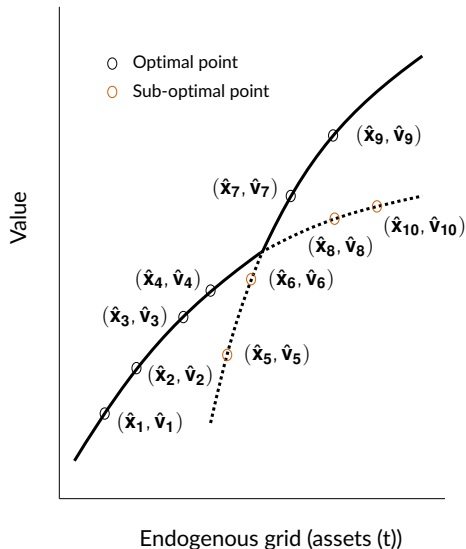
(Interior solution only, follow Iskhakov et al. 2017 occasional binding constrained policy)

FUES-EGM



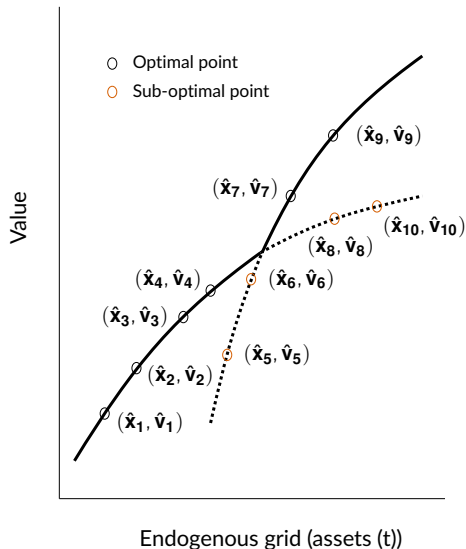
FUES-EGM

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



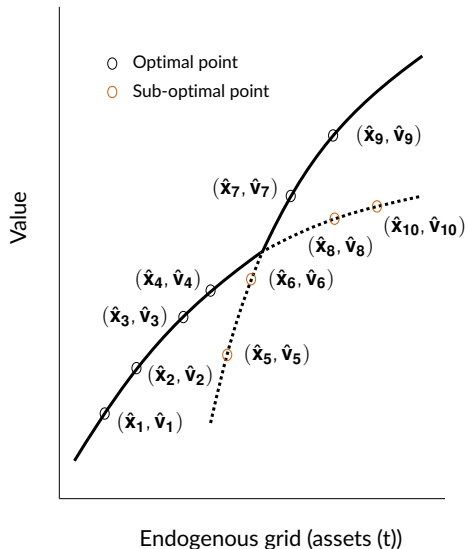
FUES-EGM

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



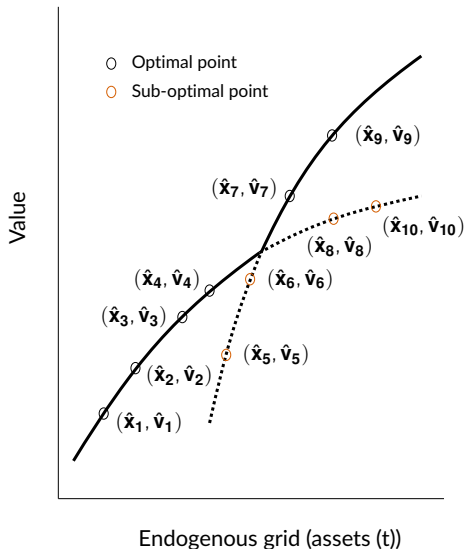
FUES-EGM

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



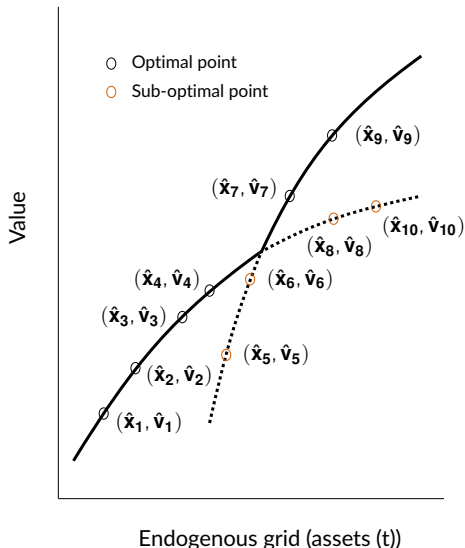
FUES-EGM

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



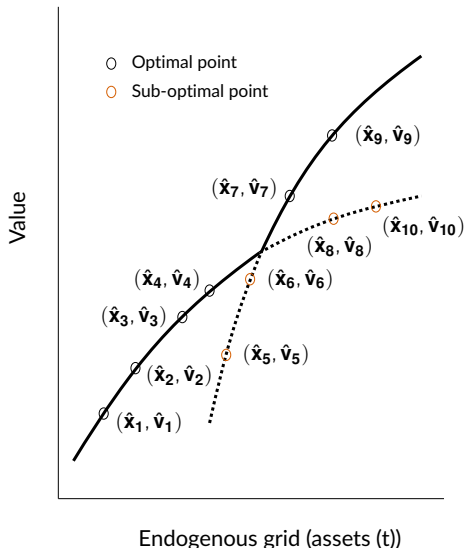
FUES-EGM

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



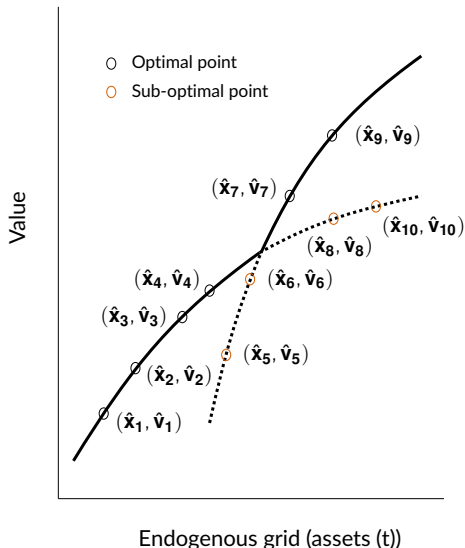
FUES-EGM

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



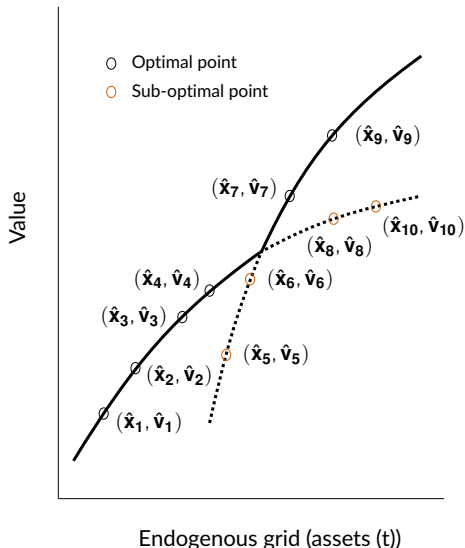
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



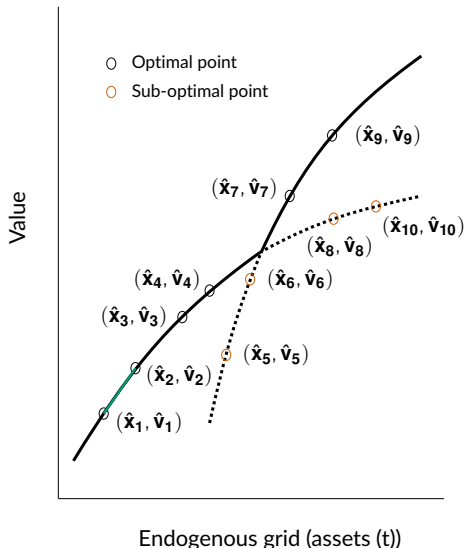
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



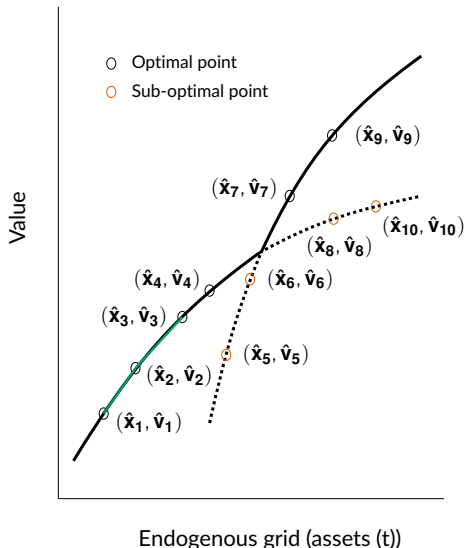
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



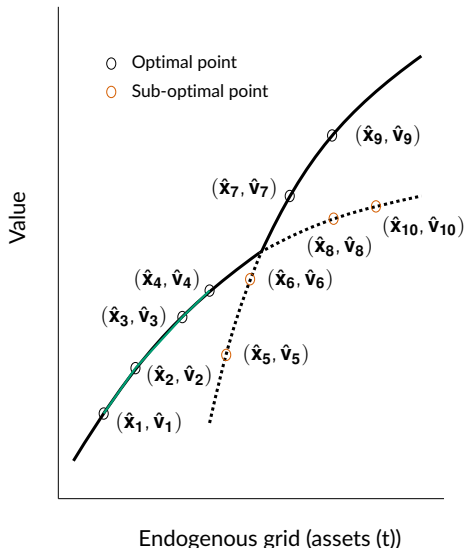
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



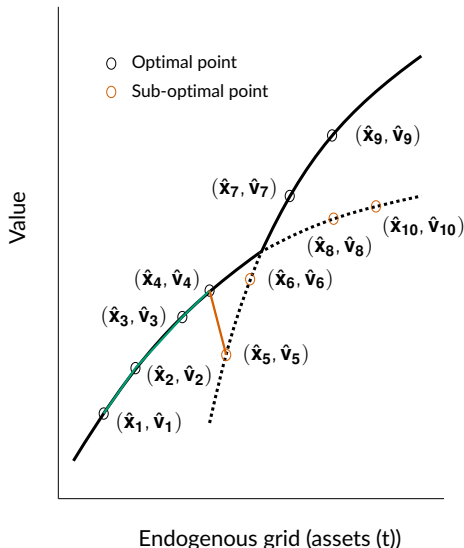
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



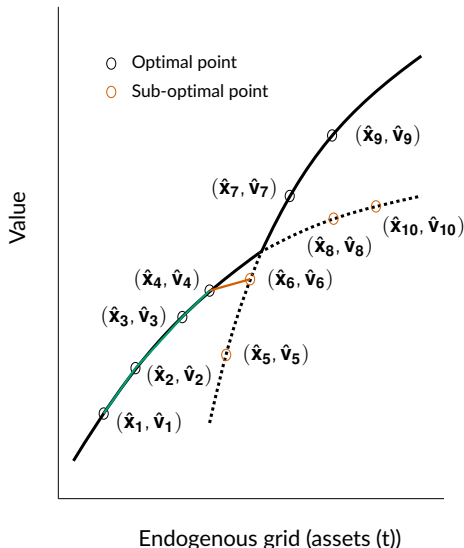
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



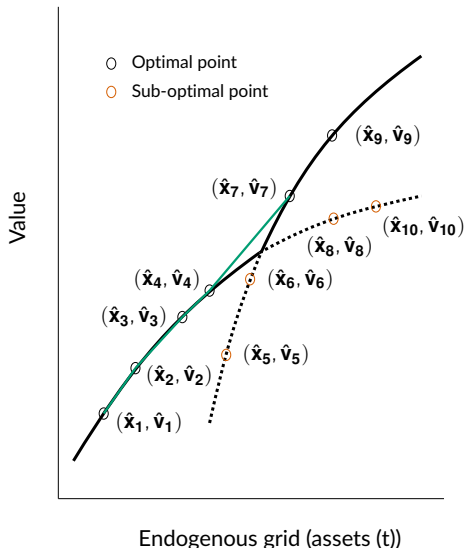
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



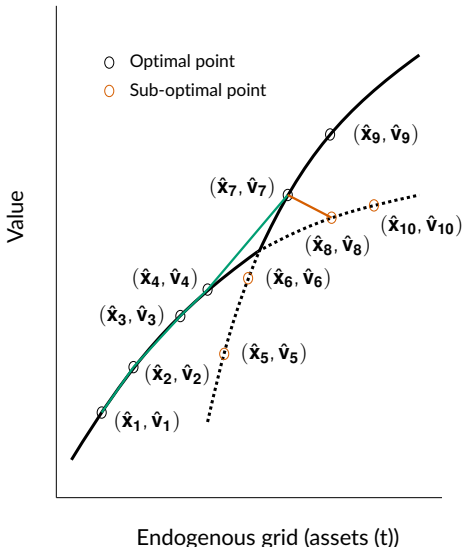
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



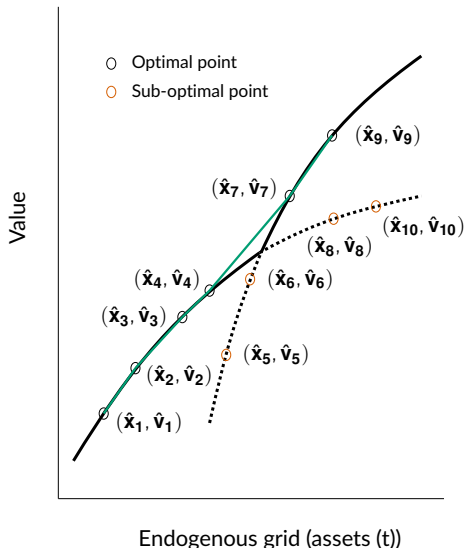
FUES-EGM

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



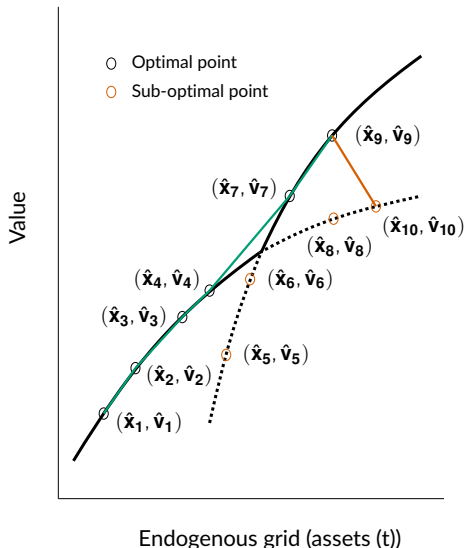
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5



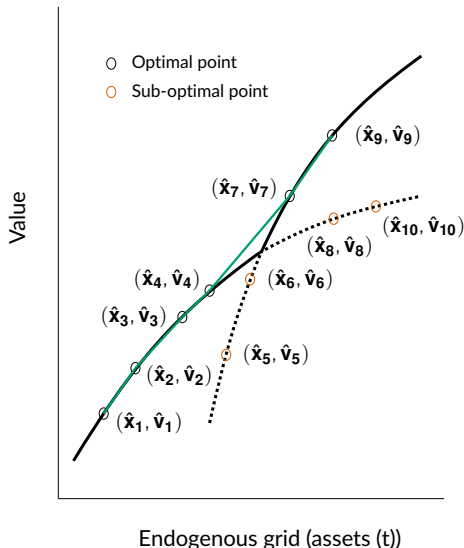
FUES-EGM

1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
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 $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$
 - Otherwise, set $i = i + 1$
7. If $i + 1 \leq |\hat{\mathbf{X}}_t|$, then repeat from step 5

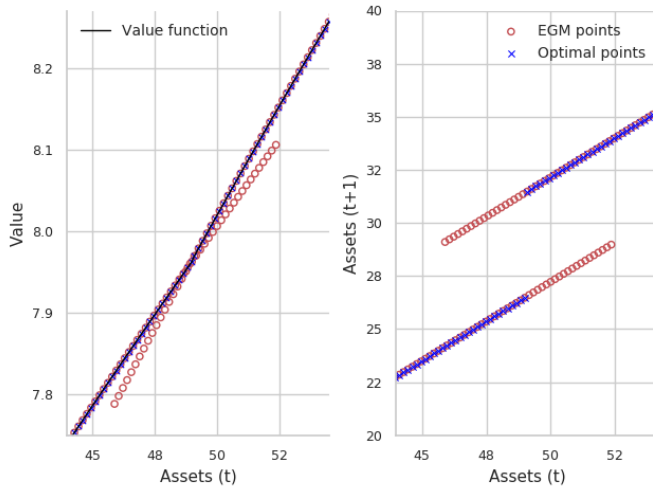


FUES-EGM

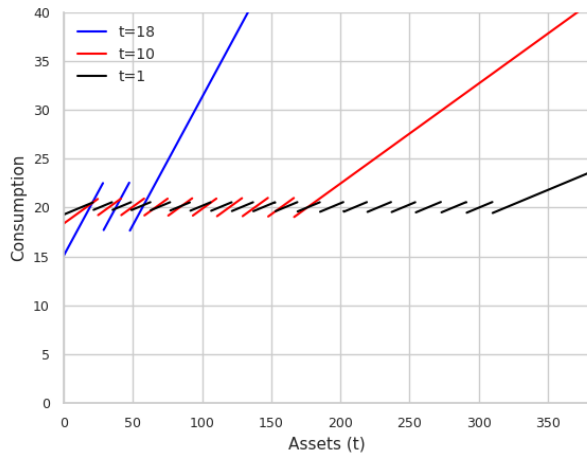
1. Compute $\hat{\mathbf{X}}_t, \hat{\mathbf{C}}_t, \hat{\mathbf{V}}_t$ and $\hat{\mathbf{X}}'_t$ using EGM
2. Set 'jump detection' threshold \bar{M}
3. Sort all in order of *endogenous* grid $\hat{\mathbf{X}}_t$
4. Start from point $i = 2$
5. Compute $g_i = \frac{\hat{v}_i - \hat{v}_{i-1}}{\hat{x}_i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



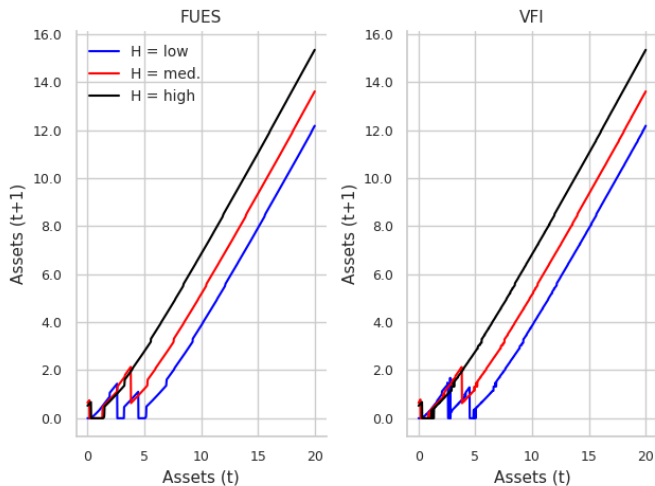
FUES-EGM



FUES-EGM



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 $\rightarrow \infty$

Extending the foundations

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

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