Fast upper-envelope scan for discrete-continuous dynamic programming *

Isabella Dobrescu† and Akshay Shanker ‡

August 3, 2022

Abstract

We introduce a fast upper-envelope scan (FUES) method to compute solutions for dynamic programming problems with continuous and discrete choices. The FUES method builds on the standard endogenous grid method (EGM). Standard EGM applied to problems with continuous and discrete choices does not by itself generate the optimal solution since the first order conditions used to generate the endogenous grid are necessary but not sufficient. FUES sequentially checks EGM candidate solution points and eliminates points not on the upper-envelope of the value correspondence by only allowing discontinuities in the policy function at non-concave sections of the value correspondence. FUES is computationally efficient, involves minimal coding time or analysis on the researcher side and does not require monotonicity assumptions on the problem. We also show that if the EGM grid size is sufficiently large, FUES is guaranteed to recover the optimal solution without error. FUES can be applied to many standard economic models, involving housing frictions, firm investment frictions, portfolio allocation or labor market participation decisions.

Key Words: dynamic programming, computational methods, non-convex optimization, Euler equations

JEL Classification:

* We thank Anant Mathur for valuable feedback. The code for applications as well as a function implementing FUES can be found at http://github.com/akshayshanker/FUES_EGM. Please download the latest version of the working paper here.

†School of Economics, University of New South Wales, Sydney 2052, Australia. Email: dobrescu@unsw.edu.au.

‡School of Finance, University of Sydney, Australia. Email: akshay.shanker@usyd.edu.au.

1 Introduction

Stochastic dynamic programming has become one of the primary tools used by quantitative researchers in fields such as economics, finance, decision theory and artificial intelligence to characterize optimal behavior through time, where todays' decisions affect the payoffs today but also the payoffs and constraints faced by the agent in the future (Bertsekas, 2022). Dynamic programming theory provides us with a general computational procedure - called value function iteration (VFI) - that can solve problems (under certain boundedness restrictions - see Stachurski, 2009) without assumptions on the convexity and smoothness of the payoffs and constraints. A drawback of VFI, however, is that it relies on numerical optimization or root-finding methods to iterate on the Bellman equation. In richer, more realistic models that demand added state spaces, the grid size grows exponentially with each additional state due to the curse of dimensionality (Rust, 1996).[1] In such cases where the grid size of a problem becomes very large, it is either inefficient or practically impossible to use a VFI that relies on numerical optimization alone to solve the problem.

In practice, computational efficiency can be gained by using first order conditions such as Euler equations. Euler equations can improve computation speed and accuracy via either the Coleman-Reffett iteration (Coleman, 1990) or the endogenous grid method EGM (Carroll, 2006). In contrast to VFI, EGM relies for instance on the analytical inversion of the Euler equation and may completely remove costly numerical root-finding or optimization steps (Iskhakov, 2015). The existing dynamic programming theory behind deriving Euler equations relies however on the concavity of the payoffs, feasibility constraints and transition functions. Without concavity, EGM generates a value correspondence with possibly multiple sub-optimal solutions that satisfy the necessary conditions. At the same time, real world applications are increasingly posing problems where constraints and payoffs are non-concave due to discrete choices (including discrete choices induced by frictions, see Skiba (1978); Rust (1987); Khan and Thomas (2008); Kaplan and Violante (2014); Dobrescu et al. (2016); Attanasio et al. (2018); Kaplan et al. (2020)).[2]

Our contribution in this paper is to present a generic, easy to code and efficient scan method to compute the upper-envelope of the value correspondence generated by EGM and obtain the optimal policy function for dynamic programming problems with discrete and continuous choice variables. The fast upper-envelope scan (FUES) works by noting that the upper-envelope of the value correspondence is the supremum of concave value functions, with each concave value function corresponding to a sequence of future discrete choices. The convex regions of the upper-envelope occur where different choice specific value functions cross, and the policy function experiences discontinuous jumps only at

[1] The high dimensional economics literature is as large as it is varied. High dimensional models are however required in two cases. First, where the agent chooses a number of quantities, for instance, where the decision space is related to household portfolio allocation across assets (Kaplan et al., 2020) or multi-product decisions (Midrigan, 2011). Second, high dimensionality arises when a number of different agents are modelled, possibly interacting via equilibrium prices as for example in Krusell and Smith (1998).

[2] Further examples of models with discrete choices include models used to understand the dynamics of housing stock adjustments (Yogo, 2016; Fagereng et al., 2019), asset pricing in the presence of frictions (Cooper, 2006) and the effect of mortgage re-financing of life-cycle asset allocation (Laibson et al., 2020).

these regions. Accordingly, FUES works by sequentially checking if inclusion of a potential optimal point forms a concave or convex section of the upper-envelope. If it forms a concave section, the point is eliminated as sub-optimal if it induces a jump in the policy function. In contrast, FUES ensures jumps are only allowed if the candidate point forms a convex section of the upper-envelope.

We also show that if the size of jumps between policy functions is bounded below (as is the case when the policy functions only contain finitely many jumps) and if all policy functions form a family of uniformly continuous functions (as is the case in standard bounded optimization problems), then FUES recovers the optimal points from the EGM without error. The main condition required for the proof is that the grid size is large enough for FUES to differentiate a jump in the policy function when it induces a shift in future discrete choices from a smooth change in the policy along a given sequence of choices.

FUES builds on the contribution of previous work on discrete-continuous dynamic programming by Fella (2014) and Iskhakov et al. (2017). These methods can successfully compute the upper envelope of the choice-specific value functions.[3] However, they rely on monotonicity and the ability to accurately identify monotone segments of the policy function containing the optimal solution, which may be difficult to do in more complex higher dimensional models.[4] Our method advances the literature in three ways. First, we prove the efficacy of FUES in the general case, implying FUES can be applied as a black-box method to identify upper-envelopes. Second, FUES is significantly easier to code and requires minimal understanding of the structure of the underlying policy functions, this is particularly useful for large higher-dimensional models. Third, FUES does not require monotonicity in any of the policy functions.

Turning to the broader literature, the scan method we present here is motivated by methods to calculate convex hulls of points, including Graham (1972). However, in the context of existing literature and up to our best knowledge, we are the first to propose a scanning algorithm to identify upper-envelopes. Finally, the class of problems we study are a special case of mixed integer non-linear problems or knap-sack problems, where discrete and continuous choices optimize an arbitrary function. Relatively recent studies have characterized both sufficient and necessary first and second order conditions for these problems (Jeyakumar et al., 2007; Jeyakumar and Srisatkunarajah, 2009), but to the best of our knowledge a condition that is both sufficient and necessary in the general setting remains elusive. Rather than derive sufficient first or second order conditions, our main contribution by contrast uses *computed* first and second order differences to identify the upper-envelope.

Our paper is structured as follows. In Section 2, we start by solving and discussing our main results in the context of three well-know applications: Section 2.1 introduces FUES informally using a simple retirement choice model (Iskhakov et al., 2017) where each period, agents choose their savings and whether or not to exit the workforce. Section 2.2 further demonstrates FUES using a model where agents choose whether or not to adjust an illiquid investment stock each period, a' la Khan and Thomas (2008). Section 2.3 then showcases

[3] See also Druedahl and Jørgensen (2017), who use triangulation methods to construct the upper-envelope.

[4] Without monotonicity, Fella (2014) uses numerical solution of the Bellman equation in non-concave regions of the correspondence to identify optimal points

FUES in the discrete choice model proposed by Fella (2014), where agents choose to hold liquid and illiquid assets (i.e., housing) via a discrete grid. In Section 3, we follow our study of these applications by formally stating FUES in its general form and providing proofs for when FUES can accurately obtain the upper-envelope without error. Section 4 concludes.

2 Illustrative applications

In this section, we illustrate the upper-envelope fast scan method using well-known dynamic optimization applications. We will briefly introduce each problem, discuss how discrete choices result in the non-convexity and detail how the fast scan method can be implemented.

2.1 Application 1: Finite horizon retirement choice model

We first consider the finite horizon retirement and work choice model studied by Iskhakov et al. (2017).

2.1.1 Model environment

Let time start at $t = 0$ and let agents live, work (if they so choose) and consume until time $t = T$. At the beginning of each period, the agent starts as a worker or retiree, with the state variable denoting their start of period work status given by the discrete variable d_t. If the agent works, they earn at wage y in the period t. At each time t, the agent can continue to work during the next period by setting $d_{t+1} = 1$, or they can exit the workforce permanently by setting $d_{t+1} = 0$. If the agent chooses to work the next period, they will incur a utility cost δ at time t. We assume all agents start as workers, so $d_0 = 1$. Agents can also consume c_t and save in capital a_t, with $a_t \in \mathbb{A}$ and $\mathbb{A} \colon = [0, \bar{a}] \subset \mathbb{R}_+$. The intertemporal budget constraint will be:

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

Utility in each period is given by $\log(c_t) - \delta d_t$, thus the agent's maximization problem each period becomes:

$$V_t^{d_t} = \max_{(c_t, d_{t+1})_{t=0}^{T+1}} \sum_{t=0}^{T} \beta^t \log(c_t) - \delta d_{t+1} \tag{2}$$

subject to Equation (1) and the fact that the agent cannot return to work after retiring, $d_{t+1} = 0$ if $d_t = 0$. Each period, $V_t^{d_t}(a_t)$ becomes the beginning of period value function. If the agent starts the period as a worker, the agent's value function will be:

$$V_t^1(a_t) = \max_{d' \in \{0,1\}} u(c) - d'\delta + \beta V_{t+1}^{d'}(a') \tag{3}$$

where $a' = (1 + r)a_t + y - c$ and such that $a' \geq 0$. If the agent enters the period as a retiree, the agent's value function becomes:

$$V_t^0(a_t) = \max_c u(c) + \beta V_{t+1}^0(a') \tag{4}$$

The optimization problem for the retiree is a standard concave problem. However, for the worker, the optimization problem is not concave since the worker optimizes jointly a discrete choice and a continuous choice. Moreover, even conditioned on $d' = 1$, the the next period value function V^1_{t+1} will not be concave since the value function is the supremum over *all future feasible combinations of discrete choices*. The non-concavity of V^1_{t+1} produces the 'secondary kinks' described by Iskhakov et al. (2017).

To see how the Bellman equation at time t implicitly controls the future discrete choices and produces the secondary kinks, write the time t worker's value function as:

$$V^1_t(a_t) = \max_{d' \in \{0,1\}, c} u(c) - d'\delta + \max_{\mathbf{d} \in \mathbb{D}} \beta Q^{\mathbf{d}}_{t+1}(a') \tag{5}$$

where $Q^{\mathbf{d}}_{t+1}$ is the $t+1$ value function conditioned on a given sequence of future discrete choices $\mathbf{d}$, and $\mathbb{D}$ is the set of all feasible sequences of discrete choices from $t+1$ to T starting with a discrete choice d' as the beginning of $t+1$ period choice. By writing the Bellman equation as above, we are able to see how $Q^{\mathbf{d}}_{t+1}$ for a given sequence of discrete choices $\mathbf{d}$, will be concave. On the other hand, V^1_{t+1} will be the upper envelope of the concave functions, with each concave function corresponding to a different sequence of discrete choices. The situation is characterised by Figure 1, where the upper envelope of concave functions is not concave.

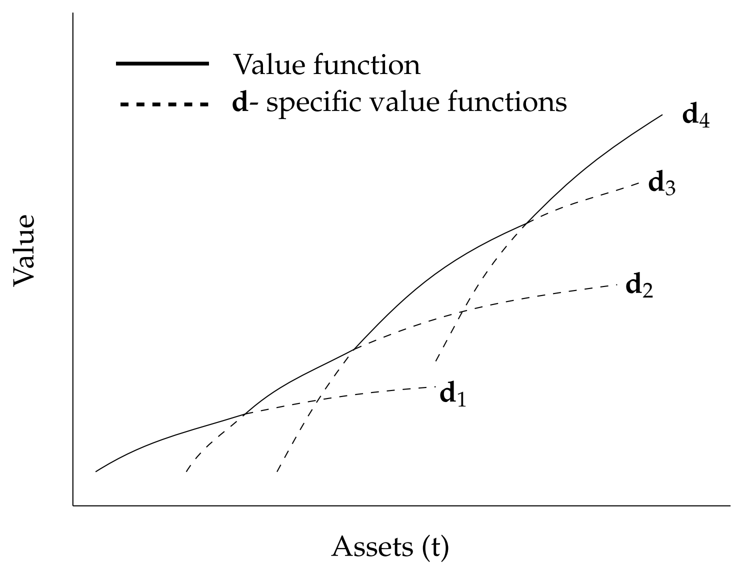

Figure 1: The worker value function at time t (V^1_t) is the upper envelope of concave functions, where each concave function is a value function conditioned on a sequence of future discrete choices.

We now detail the first order conditions and then proceed to our contribution (i.e., the FUES), as a way to use necessary first order information to efficiently compute V_t.

2.1.2 The Euler equation

If the agent chooses $d_{t+1} = 1$ so they remain a worker in $t+1$, we can write the time t worker Euler equation as:

$$u'(c_t^1) = \beta(1+r)u'(c_{t+1})$$

where c_t^1 is the time t consumption policy conditional on $d_{t+1} = 1$, while c_{t+1} is the unconditional time $t+1$ policy. On the other hand, if $d_{t+1} = 0$, then the Euler equation will be:

$$u'(c_t^0) = \beta(1+r)u'(c_{t+1})$$

It will be helpful now to write the Euler equation in functional form. Let $\sigma_t^d \colon \mathbb{A} \times \{0,1\} \to \mathbb{R}_+$ be the conditional policy function for the worker at time t if $d=1$ and retiree if $d=0$. We call σ_t^d the conditional policy because it will depend, through its second argument, on the discrete choice (to work or not to work in $t+1$) made by the worker at time t. The time t and time $t+1$ policy functions will satisfy the functional Euler equation:

$$u'((1+r)a + dy - \sigma_t^d(a, d')) = \beta u'((1+r)\sigma_t^d(a, d') + d'y - \sigma_{t+1}^{d'}(a', d'')) \tag{6}$$

where $a' = \sigma_t^d(a, d')$. Concerning the choice of whether to work or not, the time t worker will chose $d_{t+1} = 1$ if and only if:

$$\begin{aligned} u(\sigma_t^1(a,1)) - \delta + \beta V_{t+1}^1((1+r)a + y - \sigma_t^1(a,1)) \\ > u(\sigma_t^1(a,0)) + \beta V_{t+1}^0((1+r)a - \sigma_t^1(a,0)) \end{aligned} \tag{7}$$

Since the discrete choice is itself a function of the state, we can also define a discrete choice policy function $\mathcal{I}_t \colon \mathbb{A} \times \{0,1\} \to \{0,1\}$. As such, we will have $d' = \mathcal{I}_t(a,d)$ and $d'' = \mathcal{I}_{t+1}(a,d')$, where $\mathcal{I}_t$ is evaluated to satisfy (7) each period conditioned on the $t+1$ value function.

Fix some time k and suppose a sequence of policy functions satisfies the Euler equation (6), and let $\mathbf{d} = (d_t)_{t=k}^T$ be a sequence of discrete choices generated by $d_{t+1} = \mathcal{I}_t(a_t, d_t)$. This sequence of discrete choices will agree with one of the value functions in the stylized Figure 1. Moreover, for the given sequence of discrete choices $\mathbf{d}$, the sequence $(c_t)_{t=k}^T$ that agrees with the budget constraints and policy functions will be optimal. However, the sequence of discrete choices was picked recursively; rather, the entire sequence of discrete choices need to be picked at time k for the Euler equation to be sufficient. As such, the sequence satisfying the Euler equation and the recursive discrete choices may not pick the upper envelope of the value function at time k.

2.1.3 Computation using EGM and FUES

Suppose the value function V_{t+1}^{d} and the optimal policy function σ_{t+1}^{d} for $d=0$ and $d=1$ is known. Let $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}_t'$ be sequences of points such that:

$$u'((1+r)\hat{x}_i + dy - \hat{x}_i') = \beta u'((1+r)\hat{x}_i' + yd' - \sigma_{t+1}^{d'}(\hat{x}_i', d'')) \tag{8}$$

$$\hat{V}_t = u(\hat{c}_i) - d\delta + V_{t+1}^{d}(\hat{x}_i) \tag{9}$$

where $d'' = \mathcal{I}_{t+1}(\hat{x}_i', d')$, $\hat{x}_i \in \hat{\mathbb{X}}_t$, $\hat{c}_i \in \hat{\mathbb{C}}_t$, $\hat{v}_i \in \hat{\mathbb{V}}_t$ and $\hat{x}_i' \in \hat{\mathbb{X}}_t$. Such a sequence of points satisfy the Euler Equation and can be generated analytically using the EGM. In the case of the EGM, $\hat{\mathbb{X}}_t$ is the endogenous grid and $\hat{\mathbb{X}}_t'$ is the exogenous grid of points. We will consider the endogenous grid for the time t choice of $d=1$ (the endogenous grid for the retiree is concave hence we do not discuss it here). Also, order the points in $\hat{\mathbb{X}}_t$, $\hat{\mathbb{C}}_t$, $\hat{\mathbb{V}}_t$ and $\hat{\mathbb{X}}_t'$ by ascending order of the *endogenous grid points*, $\hat{\mathbb{X}}_t$.

Now consider Figure 2. Pick a point $\hat{x}_i$, with $\hat{x}_i \in \hat{\mathbb{X}}_t$ such that $\hat{x}_i'$ is optimal given $\hat{x}_i$. Observe how the curvature of the future choice-specific value functions at each point of the horizontal axis will not be the same. As such, if the points $\hat{x}_{i+1}$ and $\hat{x}_{i+1}'$ imply a different future sequence of discrete choices to $\hat{x}_i$, then $\hat{x}_{i+1}'$ will experience a 'discontinuous jump' from $\hat{x}_i'$. However, for x_{i+1}' to be on the upper-envelope, it must be that x_{i+1}' can only jump if it occurs after the crossing point between two value functions (for instance, the point $\hat{x}_6$). That is $(\hat{x}_{i+1}', \hat{v}_{i+1})$ can only jump if it makes a *convex* 'left turn' from the line joining $(\hat{x}_{i-1}', \hat{v}_{i-1})$ and $(\hat{x}_i', \hat{v}_i)$. On the other hand, if $(\hat{x}_{i+1}', \hat{v}_{i+1})$ makes a *concave* 'right turn', it cannot jump for it to be on the upper-envelope. The reason is that if $(\hat{x}_{i+1}', \hat{v}_{i+1})$ has made a 'right turn', for $(\hat{x}_{i+1}', \hat{v}_{i+1})$ to be on the upper-envelope, it must be on the *concave value function* yielding the same future sequence of discrete choices $\mathbf{d}$ as implied by $(\hat{x}_i', \hat{v}_i)$. If a 'right turn' is associated with a jump, for example, point $\hat{x}_7$, then it must be on a value function associated with a sub-optimal set of future discrete choices. We give a formal proof in Section 3.

Informally, we can use the intuition from Figure 2 to build the FUES method as follows (see Section 3 for a formal pseudo-code):

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

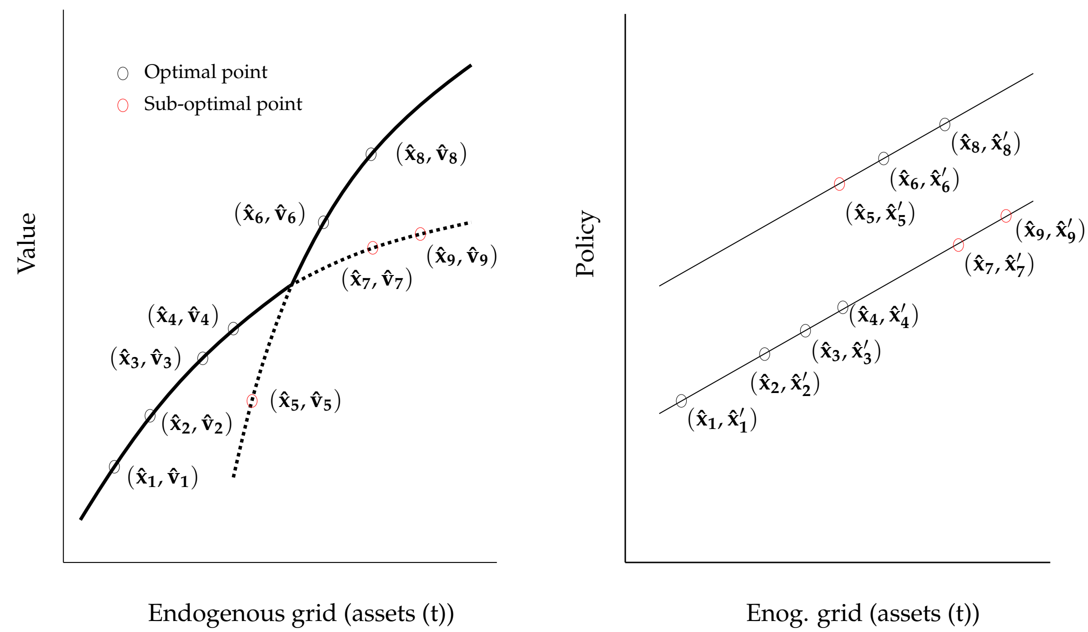

Figure 2: FUES eliminates points that cause a concave 'right turn' (left panel) from an optimal point and also cause a discontinuous jump in policy (right panel).

The method will yield a set of refined grids $\mathbb{X}_t$, $\mathbb{C}_t$, $\mathbb{V}_t$ and $\mathbb{X}'_t$. The policy and value functions can be interpolated over these grids to yield the time t approximated solution for the worker who chooses to remain a worker. The retiree value and policy functions can be calculated using standard EGM since the retiree problem is concave at each t. Once the retiree problem is solved for time t, the discrete choice at time t of whether or not to work at time $t+1$ can be evaluated. The procedure can then be repeated at $t-1$ as per standard backward policy iteration. Finally, to address the occasionally binding lower bound asset constraint, we follow the approach Iskhakov et al. (2017).

We applied FUES to solve the model studied by Iskhakov et al. (2017) and Figure 3 shows how FUES removes sub-optimal points in the value function (left panel) and selects the optimal policy function (right panel). Figure 4 plots the policy functions for workers at different ages (as a direct comparison to Figure 3 by Iskhakov et al. (2017)).[5] We found that despite a modest grid size of 2,000 points and performing only naive linear interpolation, FUES was able to accurately pick the upper-envelope of the future choice-specific value functions and also accurately replicate the shape of the consumption functions. We also found the additional time cost of FUES was a modest proportion of the overall computation time using EGM.[6]

[5] In doing this comparison, note that we retain liquid assets as the state variable (since we do not need to prove monotonicity), while Iskhakov et al. (2017) plot their Figure 3 with wealth as state variable.

[6] On a single core Intel Xeon 'Cascade Lake' CPU, the overall time to compute the policy functions for 20 years was 1.2 seconds, while FUES took 0.3 seconds.

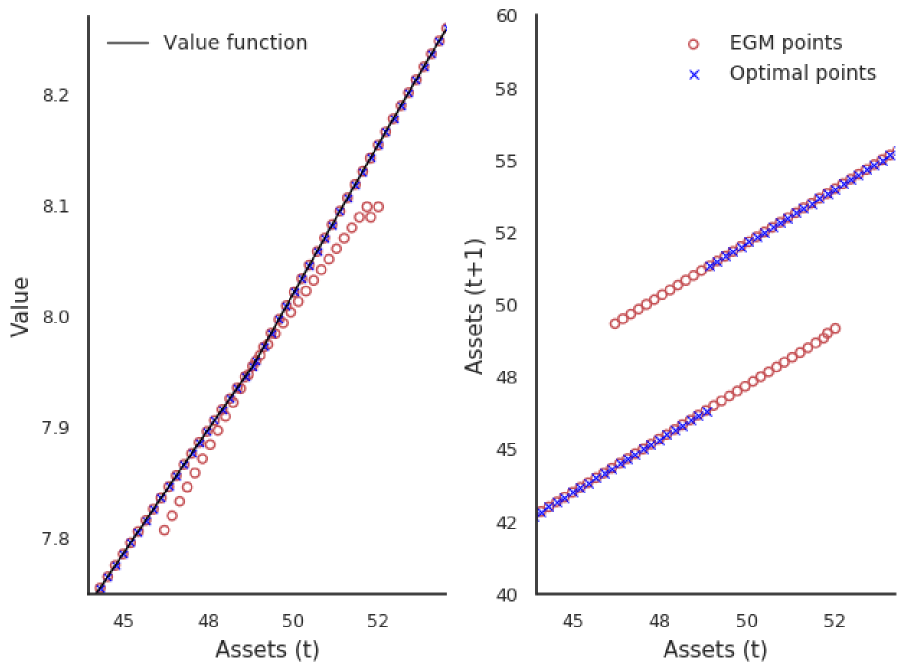

Figure 3: Value correspondence and optimal points for $t = 17$. Parameters from Iskhakov et al. (2017), Figure 3.

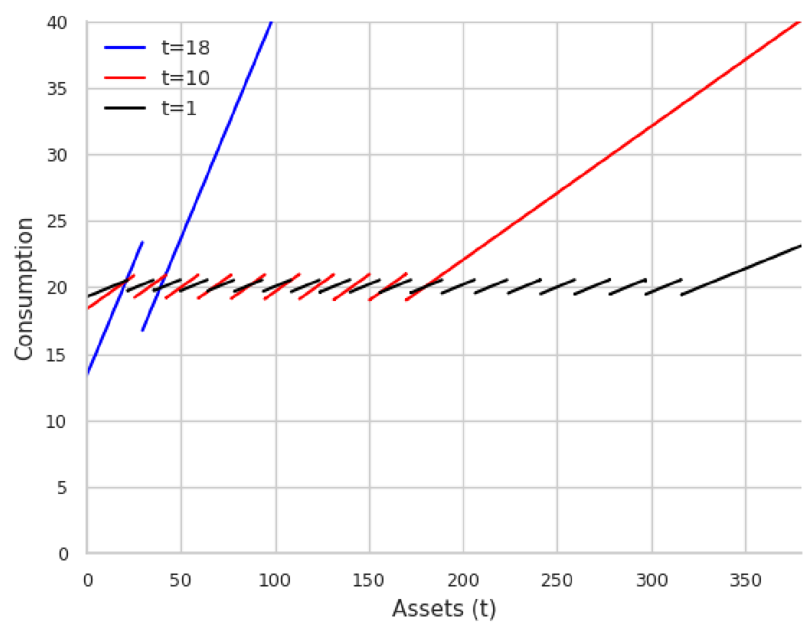

Figure 4: Optimal consumption functions for workers. Parameters from Iskhakov et al. (2017), Figure 3.

Refinements and conditions for zero approximation error

So far, we assumed that if a point $(\hat{x}_{i+1}, \hat{v}_{i+1})$ makes a left turn from point $(\hat{x}_i, \hat{v}_i)$, then it is a sufficient and *necessary* condition for the point $(\hat{x}_{i+1}, \hat{v}_{i+1})$ to lie after a cross-point of value functions between itself and $(\hat{x}_i, \hat{v}_i)$. While sufficiency is true, a point $(\hat{x}_{i+1}, \hat{v}_{i+1})$ may lie after a crossing point, and yet not generate a left-turn with respect to point x_i. To rectify the issue, we can implement a forward scan before a point is eliminated. A forward scan picks a point, q, to the right of x_{i+1} that is on the same value function as x_i and checks to see if the point x_{i+1} dominates (x_{i+1}, q). If x_{i+1} does dominate (x_{i+1}, q), then it means x_{i+1} lies after a crossing point and must be included as an optimal point.

We now turn to discuss the conditions required for FUES to guarantee it deletes all sub-optimal points but only these ones. Observe that every time we move between the two segments of the policy function in Figure 3, a large jump in x' occurs. For FUES to guarantee that an optimal point is picked up, one condition we require is that the jump size between all policies generated by different discrete choice combinations is sufficiently large compared to the distance between endogenous grid points. This ensures that when a jump to a sub-optimal point does occur, it is registered by the algorithm. Moreover, we require that the policy functions are all smooth and have a common Lipshitz constant, which ensures that changes along a policy function, given a sequence of discrete choices, are not so large as to incorrectly delete optimal points. The Lipshitz condition on the problem here is straightforward to satisfy, since there are only finitely many policy functions at any given time t and each one is smooth given a future sequence of discrete choices. While it is possible to analytically derive a threshold value $\bar{M}$ and grid size value (see Section, 3), we find that selecting a reasonable value for $\bar{M}$ based on experimentation and verifying the solution works well in practice.

A limitation of the proofs we present in Section 3 is that when there are infinitely (uncountably) many possible discrete choices, it may not be possible to show that jump sizes are bounded below. However, in the case of a smoothed version of the models presented here, with the addition of taste shocks, we see that jump sizes are sufficiently large and with a higher enough $\bar{M}$, the fast scan method is able to successfully pick up the optimal grid points (see Figure 5 and Figure 6 in the Appendix).

2.2 Application 2: Two capital investment friction model

Consider an infinite horizon model with two continuous capital stocks, one being subject to an investment friction. This type of model is a simplified version of the model used both in the firm frictions literature (Khan and Thomas, 2008) and in the housing frictions literature (Kaplan and Violante, 2014). While we describe any asset as subject to frictions, we will fix the intuition by calling the asset that is subject to friction an 'illiquid housing stock'. The model setup here is an example where EGM cannot be applied alone since the set of Euler equations are not analytically invertible (i.e., the Euler Equations are not triangular as discussed by Iskhakov (2015)). While all root-finding operations cannot be avoided, the EGM and FUES can be applied to one of the Euler equations, still reducing computation time.

2.2.1 Model environment

Let a non-negative liquid asset be denoted by a_t and a non-negative illiquid asset be denoted by H_t. The assets earn rates r and r_H, respectively. Investments can be made in and out of the stock of a_t without friction. However, investments in and out of the stock H_t require paying a utility cost χ, with $\chi > 0$. In each period t ($t = 0, 1 \ldots, T$), the agent consumes c_t and invests g_t in the illiquid stock and g_t^a in the liquid stock. The agent also makes a discrete choice d_t, where investment in and out of the illiquid stock can only be made if $d_t = 1$, otherwise if $d_t = 0$ then $g_t = 0$. Finally, in each period, the agent also earns a stochastic wage y_t and we assume $(y_t)_{t=0}^T$ is a stationary Markov process.

The following budget constraints will hold for each t. First, total investment and consumption cannot exceed income in each period:

$$y_t - c_t - d_t g_t - g_t^a \geq 0 \tag{10}$$

Second, the illiquid stock must be non-negative:

$$(1 + r^H)H_t + d_t g_t \geq 0 \tag{11}$$

Third, the liquid stock must be non-negative:

$$(1 + r)a_t + g_t^a \geq 0 \tag{12}$$

The assets will accumulate according to

$$a_{t+1} = (1 + r)a_t + g_t^a \tag{13}$$

$$H_{t+1} = (1 + r_H)H_t + g_t \tag{14}$$

Turning to payoffs, the agent lives up to time T, after which they die and value the bequest they leave behind according to a function $\theta \colon \mathbb{R}_+ \to \mathbb{R}$. Per-period utility is given by a real-valued function $\varphi^u \colon \mathbb{R}_+ \times \{0, 1\} \to \mathbb{R}$ defined as:

$$\varphi^u(c_t, d_t) = \mathbb{1}_{t \leq T} \left(u(c_t) + d_t \chi\right) + \mathbb{1}_{t=T+1} \theta(a_{T+1}) \tag{15}$$

where u is a concave, differentiable, decreasing function. Formally, the agent's problem is:

$$V_t(a_t, H_t, y_t) = \max_{(a_i, H_i, d_i, g_i, g_i^a)_{i=t}^{T+1}} \sum_{i=t}^{T+1} \beta^t \mathbb{E} \varphi_i^u(c_i, d_i) \tag{16}$$

such that (10) to (15) hold. The standard Bellman equation for the problem will be:

$$V_t(a, H, y) = \max_{g, g^a, d} \varphi^u(c, d) + \mathbb{E}_y V_{t+1}(a', H', y') \tag{17}$$

where the prime notation indicates next period state values satisfying the budget constraints and the transition equations.

2.2.2 Euler equations

The problem will feature two Euler equations, one for each state. The Euler equation for the liquid state will be:

$$u'(c_t) \geq \beta \mathbb{E}_t u'(c_{t+1}) \tag{18}$$

while the Euler equation for the illiquid housing stock will be:

$$u'(c_t) \geq \mathbb{E}_t \left(\beta^{\tau - t}(1 + r^H)^{\tau - t} u'(c_\tau) \right), \qquad \text{if} \quad d_t = 1 \tag{19}$$

The intuition for the first Euler equation (18) related to the liquid stock is standard. The Euler equation for the illiquid stock (19), however, features an additional term τ defined as as the next period when $d_\tau = 1$. Since the next time the stock is adjusted will be stochastic, τ becomes a random stopping time. The Euler equation then tells us that the shadow value (price) of investment (or withdrawal) from the illiquid housing stock is given by the discounted expected value of the stock *when the stock is next liquidated*, not in the next period as in the standard Euler equation (see Kaplan and Violante (2014), where a similar intuition is also discussed). Appendix C.3 of Dobrescu et al. (2022) shows the formal derivation of an Euler equation with future adjustments.

Since the solution sequence for the problem will be recursive, there exists measurable functions σ_t^a, σ_t^H and $\mathcal{I}_t$ such that $H_{t+1} = \sigma_t^H(a_t, H_t, y_t)$, $a_{t+1} = \sigma_t^a(a_t, H_t, y_t)$ and $d_t = \mathcal{I}_t(a_t, H_t, y_t)$ for each t. Also let $\sigma_t^{a,d}$ and $\sigma_t^{H,d}$ denote the choice-specific policy functions conditioned on the time t discrete choice $d \in \{0, 1\}$. Inserting the policy functions back into (18) and (19) yields the functional Euler equation for the housing stock:

$$u'(c) \geq \beta R^H \mathbb{E}_y \mathcal{I}_{t+1}(a', H', y') u'(c') + (1 - \mathcal{I}_{t+1}(a', H', y')) \mathbb{E}_t \Theta_{t+1}(a', H', y'') \tag{20}$$

where:

$$c' = R\sigma_t^{a,1}(a, H, y) + R^H \sigma_t^{H,1}(a, H, y) + y' - \sigma_{t+1}^{a,1}(a', H', y') - \sigma_t^{H,1}(a', H', y') \tag{21}$$

$$c = Ra + R^H H + y - \sigma_t^{a,1}(a, H, y) - \sigma_t^{H,1}(a, H, y) \tag{22}$$

and we have $a' = \sigma_t^{a,1}(a, H, y)$ and $H' = \sigma_t^{H,1}(a, H, y)$. The additional term Θ_t is a multiplier denoting the continuation marginal value of the housing stock if time $t + 1$ stock is not adjusted. For a set of recursive policy functions, we can compute Θ_t on the states as

$$\Theta_t(a, H, y) = \mathbb{E}_y \mathcal{I}_{t+1}(a', H', y') u'(c') + (1 - \mathcal{I}_{t+1}(a', H', y')) \mathbb{E}_t \Theta_{t+1}(a', H', y') \tag{23}$$

The functional Euler equation for the liquid capital stock becomes:

$$u'(c) \geq \beta R \mathbb{E}_y u'(c') \tag{24}$$

2.2.3 Computation using EGM and FUES

Suppose we know V_{t+1}, $\{\sigma_{t+1}^{a,d}, \sigma_{t+1}^{H,d}\}_{d\in\{0,1\}}$, $\mathcal{I}_{t+1}$ and Θ_{t+1}. We can first apply standard EGM with FUES to evaluate $\sigma_t^{a,0}$ for non-adjusters since only one Euler equation (i.e., equation (24)) will hold. For each possible time-t housing state H_t in the housing grid and exogenous state y_t, we can approximate $\sigma_t^{a,0}(\cdot, H_t, y_t)$ by first setting an exogenous grid of a' values (holding $H' = H_t$ fixed since no adjustment takes place) and then creating an endogenous grid of t period liquid states using equation (24) and the budget constraint (10). FUES can then be applied as described in Section 2.1.3 to eliminate the sub-optimal points to obtain an approximation of the policy function $\sigma_t^{a,0}$.[7]

Now, we turn to calculate the policy functions for adjusters. To calculate σ_t^H, we first fix H_t and y_t. Then an exogenous grid over H' values can be created and for each value H', all the multiple roots of a' that satisfy equation (20) and (24) need to be calculated. In particular, holding H' fixed, we can use a multiple root finding procedure to evaluate roots to:

$$\begin{aligned} a' \mapsto \beta R \mathbb{E}_y u'(c') - \max\{&\beta R^H \mathbb{E}_y \mathcal{I}_{t+1}(a', H', y') u'(c') \\ &+ (1 - \mathcal{I}_{t+1}(a', H', y'))\mathbb{E}_y \Theta_{t+1}(a', H', y'), \beta R \mathbb{E}_y u'(\tilde{c})\} \end{aligned} \tag{25}$$

where we have

$$\tilde{c} = R^H H' + y' - \sigma_{t+1}^{a,1}(0, H', y') - \sigma_t^{H,1}(0, H', y')$$

For each of the multiple roots of equation (25), we can evaluate the time t value function, and pick the a' with the highest value function. With a candidate value H' and a', we can construct an endogenous grid of current period liquid assets that satisfy the budget constraints (10) - (14). FUES can be applied as detailed in Section 2.1.3 to the endogenous grid points a and exogenous grid points H' to eliminate the sub-optimal points and obtain an approximation of the policy function $\sigma_t^{H,1}$. With $\sigma_t^{H,1}$ approximated, we can then evaluate $\sigma_t^{a,1}$ using the optimal a' value associated with each H' that was derived using the multiple root-finding procedure.

We solve the model using both traditional VFI and the above procedure using FUES. We set $\beta = .945$, $\chi = .4$, $\gamma = 1.5$, $\delta = .1$, $R_H = .05$, $R = .025$, with terminal period $T = 65$. We also calibrate the income shocks and bequest function to the calibration of females in Dobrescu et al. (2016). The consumption policy functions for a 40 year old are shown in Figure 7 in the Appendix. Not only does FUES with EGM result in less numerical errors, we also see significant speed gain. Using a grid-size of 1,000 for liquid assets and 500 for housing, VFI took approximately 456 seconds *parralelized over 48 Intel Cascade Lake CPUs*. On the other hand, *on a single CPU*, FUES with EGM was completed in 35 seconds.

2.3 Application 3: Infinite horizon housing choice model

In our final example, we turn to the discrete choice model of housing decisions posed by Fella (2014). Unlike the previous example, the housing choice grid itself is discrete. Because of this, we no longer have to use root-finding procedures in addition to the EGM. To

[7] Once again, to address occasionally binding constraints, we follow the approach Iskhakov et al. (2017).

demonstrate the efficacy of FUSE in non-monotone applications, we formulate the model such that liquid assets are not a monotone function of the liquid state. [8]

2.3.1 Model environment

Consider an agent who draws an infinite sequence of bounded, Markov labour income shocks $(y_t)_{t=0}^{\infty}$. Each period the agent decides how much to consume of durable goods c_t, how much to save in liquid assets a_t, and how much to save in housing assets H_t. Housing serve a dual role, namely they are a form of investment good but also provide consumption services. Moreover, housing can only be purchased in discrete amounts on a finite grid $\mathbb{H}$, and adjusting the housing stock each period incurs a fixed transaction cost $\phi(H_{t+1} - H_t)$, with $\phi \in [0, 1)$. For simplicity, we assume no borrowing and so, $a_t \geq 0$ must hold.

Formally, the budget constraint for each household is

$$a_{t+1} = (1 + r)a_t + y_t - (H_{t+1} - H_t) - \phi \mathbb{1}_{H_{t+1} \neq H_t}(H_{t+1} - H_t) - c_t \tag{26}$$

with the per-period utility function being

$$u(c, H) = \theta \log(c) + (1 - \theta) \log(H) \tag{27}$$

The agent maximises

$$V_t(a, y, H) = \max_{(a_i, H_i)_{i=t}^{\infty}} \mathbb{E} \sum_{t=0}^{\infty} \beta^t u(c_i, H_{i+1}) \tag{28}$$

such that (26) and (27) hold, $H_t \in \mathbb{H}$ for each t and $a_0, y_0, h_0 = a, y, H$. We can write the Bellman equation as

$$V(a, y, H) = \max_{c, H'} u(c, H') + \mathbb{E}_y V'(a', y', H') \tag{29}$$

such that a' satisfies the feasibility condition and the conditional expectation is taken with respect to y. Since this is an infinite horizon problem, recall that the fixed point to the Bellman equation yields the value function, and the maximising correspondence yields the policy functions.

2.3.2 The Euler equation

Let us write down the necessary Euler equation. Consider time $t + 1$ policy functions σ^a_{t+1} and σ^H_{t+1}, which map the time $t + 1$ states (i.e., liquid assets a_{t+1} and housing H_{t+1}) to time $t + 2$ states. We then have that $a_{t+2} = \sigma^a_{t+1}(a_{t+1}, H_{t+1})$ and $H_{t+1} = \sigma^H_{t+1}(a_{t+1}, H_{t+1})$. A necessary condition is for a_{t+1} to solve the Euler equation in terms of a' as follows:

$$\begin{aligned} &u'((1 + r)a_t + y_t - \Phi(H_t, H_{t+1}) - a') \\ &\qquad \geq \mathbb{E}_{y_t} \beta u'((1 + r)a' + y_{t+1} - \Phi(H_{t+1}, H_{t+1}) - \sigma^a_{t+1}(a', H_{t+1})) \end{aligned} \tag{30}$$

[8] As Fella (2014) shows, it is possible to formulate the model with total wealth as the state so the policy functions are monotone.

where we let $\Phi(H_t, H_{t+1}) = (H_{t+1} - H_t) - \phi \mathbb{1}_{H_{t+1} \neq H_t}(H_{t+1} - H_t)$ to ease the notation.

Once again, to state the source of the con-concavity here, recall that the standard approach in concave models is to solve a', then approximate σ_t, σ_{t-1}, σ_{t-2} and so on until σ_t converges. However, in the case of discrete choices, even if the choices H_{t+1} and H_{t+1} are held fixed or calculated to maximise the continuation value, the difficulty arises when we try to compute the policy functions recursively. Given the time $t+1$ asset policy function σ^a_{t+1}, the term c_{t+1} may not be monotone and there may be multiple roots to the above equation in a' because σ^a_{t+1} has implicitly inherited discrete choices from future periods. As a result, we may implicitly select a sub-optimal discrete choice for periods after time $t+2$ and hence select a sub-optimal local turning point a'.

2.3.3 Computation using EGM and FUES

In light of our discussion in Section 2.1, the use of FUES to the application here is straightforward. To proceed with computation using EGM and FUES, we can start with an initial guess for the value function and policy functions, V_T, σ^a_T and σ^H_T. Since there is only one Euler equation, we can use EGM without any root-finding steps and calculate the endogenous grid for each H, H' choice fixed. We then follow the procedure detailed in Section 2.1.3 to remove sub-optimal grid points and evaluate the discrete choice policy function. Once approximations of V_{T-1}, σ^a_{T-1} and σ^H_{T-1} are obtained, we continue to iterate until $\|V_{T-i} - V_{T-i-1}\|_\infty < \epsilon$ for some pre-determined $\epsilon > 0$. Figure 8 shows the non-monotone asset policy functions for different beginning of period housing levels (but allowing the end of period housing choice, H' to be endogenous).

We use the same parameters as Fella (2014), except we set the lower bound on assets to zero, two-state i.i.d income shocks and a house grid size with 4 points. Once again, we see that FUES is able to accurately recover the optimal policy function and avoid the numerical errors typical of value function iteration. Moreover, using a grid-size of 1,000 for liquid assets and 4 house sizes, VFI took approximately 84 seconds *parallelized over 48 Intel Cascade Lake CPUs*. On the other hand, *on a single CPU*, FUES with EGM was completed in 4.5 seconds.

Note that while the infinite horizon policy function implicitly defines infinitely many future discrete housing choices, the policy functions only contain finitely many jumps. This implies the jump sizes are bounded below and FUES can accurately detect a jump and eliminate sub-optimal jumps.[9]

3 Theoretical foundations

In this section, we will formally define a general discrete-continuous optimization problem, apply the EGM and state the FUES method as a formal pseudo-code. Once we have

[9]Nonetheless, in a general infinite horizon model, we have not been able to define conditions that guarantee only finitely many jumps. Infinite jumps will also lead to approximation error not only with FUES, but also alternative methods such as the one proposed by Iskhakov et al. (2017) and Fella (2014) which rely on locating overlapping sections of the EGM generated policy functions.

stated the formal method, we turn to state conditions and give a proof to guarantee FUES recovers the optimal value function points from the EGM.

3.1 The general discrete-continuous optimization problem

Consider the problem of maximizing the sum of a concave function $u\colon \mathbb{A} \to \bar{\mathbb{R}}$ and the supremum of a family of smooth concave functions $G^i\colon \mathbb{A} \to \bar{\mathbb{R}}$, with $i \in \mathbb{D}$, where $\mathbb{D}$ is a family of discrete choices and $\mathbb{A} = [0, \bar{K}]$. We can write the problem formally as:

$$V(x) = Q(c, x) \tag{31}$$

where

$$Q(c, x) \colon = u(c) + \max_i G^i(f(c, x)) \tag{32}$$

and such that x is given and $f(c, x) \leq \bar{K}$ for each x.

Next, define $\mathcal{I}(x') = \arg\max_i G^i(x')$. Also, define the optimal policy correspondence:

$$\sigma(x) = \arg\max_{c \geq 0} \quad u(c) + \max_i G^i(f(c, x)) \tag{33}$$

and the policy correspondence and choice-specific value function holding the discrete choice fixed

$$\sigma^i(x) = \arg\max_{c \geq 0} \quad u(c) + G^i(f(c, x)) \tag{34}$$

$$Q^i(c, x) \colon = \max_{c \geq 0} \quad u(c) + G^i(f(c, x)) \tag{35}$$

3.1.1 The EGM and FUES method

Let $\hat{\mathbb{X}}'$ be the exogenous grid of values x', with $|\hat{\mathbb{X}}'| = N$, where N is the grid size. We assume grids are sequences, that is, $\hat{\mathbb{X}}' = \{\hat{x}'_0, \hat{x}'_1 \ldots \hat{x}'_i \ldots\}$. Now let $\bar{f}(c, x')$ denote the inverse of $f(c, x)$ in its second argument. Recall a necessary first order condition for an interior solution to the problem will be:

$$u'(c) + G^{\mathcal{I}(f(c,x)),\prime}(f(c, x)) = 0 \tag{36}$$

Remark 1 *For $x \in A$, consider $\tilde{c}$ that satisfies $u'(\tilde{c}) + G^{\mathcal{I}(f(\tilde{c},x))}(f(\tilde{c}, x)) = 0$. If $i = \mathcal{I}(f(\tilde{c}, x))$ is fixed, then $\tilde{c}$ will satisfy:*

$$\tilde{c} = \arg\max \quad u(c) + G^{i,\prime}(f(c, x)) \tag{37}$$

The endogenous grid of values x is defined as:

$$\hat{\mathbb{X}} \colon = \bar{f}\left(u^{\prime,-1}(G^{\mathcal{I}(\hat{\mathbb{X}}'),\prime}(\hat{\mathbb{X}}')), \hat{\mathbb{X}}'\right) \tag{38}$$

We can also calculate unrefined grids containing candidate policies and values:

$$\hat{\mathbb{C}}\colon = u^{\prime,-1}(G^{\mathcal{I}(\hat{\mathbb{X}}^{\prime}),\prime}(\hat{\mathbb{X}}^{\prime})) \tag{39}$$

$$\hat{\mathbb{V}}\colon = u(\mathbb{C}) + G^{\mathcal{I}(\hat{\mathbb{X}}^{\prime})}(\hat{\mathbb{X}}^{\prime}) \tag{40}$$

Before stating the formal FUES method, we make some further remarks on notation. Note the endogenous grid will be initially ordered in ascending order according to the order of $\hat{\mathbb{X}}^{\prime}$. We now order the grids by the values in $\hat{\mathbb{X}}$ and, through a slight abuse of notation, continue to denote the recorded grids by $\hat{\mathbb{X}}^{\prime}$, $\hat{\mathbb{C}}$, $\hat{\mathbb{V}}$ and $\hat{\mathbb{X}}$. Next, for a sequence of points X, when we state $Y = X \setminus x_i$, we also imply that the points in Y are re-indexed while maintaining order so the final element of Y is y_K where $K = |Y|$.

In the Algorithm below, we let $N = |\hat{\mathbb{X}}|$. Moreover, let M, D, δ and L be chosen by the researcher. The number M represents the common Lipshitz constant for all policy functions, D is the minimum distance between the policy functions and δ is the largest gap between endogenous grid points. Together, M, D and δ determine the threshold at which FUES registers a jump between policy functions, and in practice, these constants can be chosen by the researcher. Finally, let L be the number of steps FUES takes in the 'forward scan' part of the algorithm. With these remarks on notation, we are now ready to formally state the FUES method.

Algorithm 1: Fast upper-envelope scan

Data: $N > 0$, $L > 0$, $M > 0$, $D > 0$, $\delta > 0$ $\hat{\mathbb{X}}'$, $\hat{\mathbb{C}}$,$\hat{\mathbb{V}}$,$\hat{\mathbb{X}}$
Result: $\mathbb{X}'$, $\mathbb{C}$, $\mathbb{X}$ and $\mathbb{V}$

1 $k \leftarrow 2, l \leftarrow 2$;
2 $\mathbb{X}' \leftarrow \hat{\mathbb{X}}', \mathbb{X} \leftarrow \hat{\mathbb{X}}, \mathbb{C} \leftarrow \hat{\mathbb{C}}, \mathbb{V} \leftarrow \hat{\mathbb{V}}$; `/* Refined grids init. to full grids. */`
3 **while** $l < N$ **do**
4 $\quad \tilde{g}_k \leftarrow \frac{\hat{v}_k-\hat{v}_{k-1}}{\hat{x}_k-\hat{x}_{k-1}}$, $\tilde{g}_{k+1} \leftarrow \frac{\hat{v}_{l+1}-\hat{v}_k}{\hat{x}_{l+1}-\hat{x}_k}$, $\tilde{g}^a_{k+1} \leftarrow \frac{\hat{x}'_{l+1}-\hat{x}'_k}{\hat{x}_{l+1}-\hat{x}_k}$;
5 $\quad$ **if** *If* $\tilde{g}_{k+1} \leq \tilde{g}_k$ *and* $|\tilde{g}^a_{k+1}| \geq \frac{D}{\delta} - M$ **then**
6 $\quad\quad j \leftarrow k+2$;
7 $\quad\quad$ **while** $k+1 < j \leq L+k+2$; `/* Forward scan while loop. */`
8 $\quad\quad$ **do**
9 $\quad\quad\quad g_{j,l+1} \leftarrow \frac{\bar{v}_j-\bar{v}_{l+1}}{\bar{x}_j-\bar{x}_{l+1}}$, $g_{j,k} \leftarrow \frac{\bar{v}_j-\bar{v}_k}{\bar{x}_j-\bar{x}_k}$, $\tilde{g}^a_{j,k} \leftarrow \frac{\hat{x}'_j-\hat{x}'_k}{\hat{x}_j-\hat{x}_k}$;
10 $\quad\quad\quad$ **if** $g_{j,l+1} > 0$, $g_{j,l+1} > \tilde{g}_{k+1}$, $g_{j,k} < \tilde{g}_{k+1}$ *&* $|\tilde{g}^a_{j,k}| < \frac{D}{\delta} - M$ **then**
11 $\quad\quad\quad\quad \mathcal{D} \leftarrow k$;
12 $\quad\quad\quad\quad$ break;
13 $\quad\quad\quad$ **else if** $g_{j,l+1} < 0$, $g_{j,l+1} > \tilde{g}_{k+1}$, $g_{j,k} < \tilde{g}_{k+1}$ *&* $|\tilde{g}^a_{j,k}| < \frac{D}{\delta} - M$ **then**
14 $\quad\quad\quad\quad \mathcal{C} \leftarrow k$;
15 $\quad\quad\quad\quad$ break;
16 $\quad\quad\quad$ **else if** $j = L+k+2$ **then**
17 $\quad\quad\quad\quad \mathcal{D} \leftarrow k$;
18 $\quad\quad$ **end**
19 $\quad$ **else if** $\tilde{g}_{k+1} > \tilde{g}_k$ **then**
20 $\quad\quad \mathcal{C} \leftarrow k$;
21 $\quad$ **if** $\mathcal{C} = k$ **then**
22 $\quad\quad k = k+1$
23 $\quad$ **else**
24 $\quad\quad \mathbb{X}' \leftarrow \mathbb{X}' \setminus x'_{k+1}, \mathbb{V}' \leftarrow \mathbb{V}' \setminus v'_{k+1}, \mathbb{C}' \leftarrow \mathbb{C}' \setminus c'_{k+1}$
25 $\quad$ **end**
26 $\quad l \leftarrow l+1$
27 **end**

The main condition of the FUES method, at line 5 of Algorithm 1 checks to see if a discontinuity occurs at a concave 'right turn' and eliminates such points. The purpose of the forward scan is to deal with the case when there is a point after a crossing between two choice-specific value functions and the point does not make a convex 'left turn' from the point before the crossing point. In practice, we find removing the forward scan does not result in poorer performance in applications, since the difference in derivatives of different choice-specific value functions may be large enough. However, the forward scan is essential to the proof that guarantees all optimal points are accurately identified.

3.2 Proof of no approximation error

The FUES method can be guaranteed to remove all sub-optimal points and retain all optimal points for a large enough grid size relative to the jump sizes. To prove the result, we now state some additional definitions and place assumptions on the problem structure.

Definition 1 *Let $T_P \subset \mathbb{A}$ be the set of 'crossing points' between the choice specific value functions. That is, the set of $x \in \mathbb{A}$ such that for some $m, l \in \mathbb{D}$, $Q_m(x) = Q_l(x)$ and for $\epsilon > 0$, $V(x+\epsilon) = Q_l(x+\epsilon) > Q_m(x+\epsilon)$ and $V(x-\epsilon) = Q_m(x-\epsilon) > Q_l(x-\epsilon)$.*

Note that we will have $Q'_m(x) < Q'_l(x)$. Also, let $\mathbb{X}^\star$ be the optimal sub-sequence of points of $\hat{\mathbb{X}}$.

The first assumption we make is a standard restriction on the transition function.

Assumption 1 *The function $f\colon \mathbb{R}^2_+ \to \mathbb{R}$ is invertible, smooth and monotone.*

We now assume that the distance between choice specific policy functions is bounded below and that the rate of change of the policy functions is uniformly bounded above by a common constant.

Assumption 2 *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 $x \leq \bar{K}$.*

Assumption 3 *The family of functions $x \mapsto f(\sigma_i(x), x)$ for $i \in \mathbb{D}$ have a common Lipshitz constant M.*

We also assume the distance between the endogenous grid points are small enough such that FUES is able to use the above two assumptions to differentiate between a jump and a movement along a choice specific policy function.

Assumption 4 *There exists $\delta > 0$ such that for all $j \leq |\mathbb{X}^\star|$, $|x^\star_{j+1} - x^\star_j| \leq \delta$ and $\frac{D}{\delta} > 2M$.*

Finally, we place some assumptions on the grid points around crossing points. In particular, we first assume in items 1. and 2. below that there are 'enough' optimal points between any two separate crossing points. We also assume in item 3. below that the first point after a given crossing point is optimal.

Assumption 5 *Fix $\tilde{x} \in T_P$ and let $x^\star_k$ be the largest element in $\mathbb{X}^\star$ such that $x^\star_k \leq \tilde{x}_i$ and $x^\star_{k+1}$ be the smallest element in $\mathbb{X}^\star$ such that $x^\star_{k+1} \geq \tilde{x}$. The following hold:*

1. *If $v^\star_{k+1} = Q^l(x^\star_{k+1})$ for some l, then $v^\star_{k+2} = Q^l(x^\star_{k+2})$*
2. *If $v^\star_k = Q^m(x^\star_k)$ for some m, then $v^\star_{k-1} = Q^m(x^\star_{k-1})$*
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 identified by Definition 1 as the value function crossing at point $\tilde{x}_i$ from below.*

We are now ready to state and prove our results. The first result says that if a point makes a 'left turn' from an optimal point, then it is also optimal.

Claim 1 *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:*

$$\frac{\hat{v}_{i+1} - \hat{v}_i}{\hat{x}_{i+1} - \hat{x}_i} < \frac{\hat{v}_{i+2} - \hat{v}_{i+1}}{\hat{x}_{i+2} - \hat{x}_{i+1}} \tag{41}$$

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

Proof. Fix l, with $l \in \mathbb{D}$, and such that $V(\hat{x}_{j+1}) = u(\hat{c})_{j+1}) + Q^l(\hat{x}'_{j+1})$. Suppose by contradiction that $\hat{x}_{i+2} \notin \mathbb{X}^\star$. Suppose first that $V(\hat{x}_i) = Q^l(\hat{x}_i)$. By the concavity of Q^l and since Equation (41) holds, we must have $\hat{v}_{i+2} = u(\hat{c})_i + G^m(\hat{x}'_i)$ and $m \neq l$. Moreover, also by concavity and Equation (41), we must have $Q^m(\hat{x}_{i+2}) > Q^l(\hat{x}_{i+2})$. This implies that $V(\hat{x}_{i+1}) = Q^p(\hat{x}_{i+1})$ for $p \neq l$ and $p \neq m$. Moreover, the functions Q^p and Q^l must cross at some point $\tilde{x}_i \in T_p$ with $\tilde{x}_i \in [\hat{x}_{j+1}, \hat{x}_{j+2}]$. However, we have now violated Assumption 5, item 3., since the assumption implies $m = p$ or $m = l$. Thus, in the case where $V(\hat{x}_i) = Q^l(\hat{x}_i)$, we must have $\hat{x}_{i+2} \in \mathbb{X}^\star$

Now suppose $V(\hat{x}_i) \neq Q^b(\hat{x}_i)$ for some $b \neq l$. This implies a crossing point $\tilde{x} \in [\hat{x}_j, \hat{x}_{j+1}]$ and another crossing point $\hat{\tilde{x}} \in [\hat{x}_{j+1}, \hat{x}_{j+2}]$. By Assumption 5, item 3., we again have a contradiction since the assumption requires $v^\star_{j+2} = Q^l(x^\star_{j+2})$ or $v^\star_{j+2} = Q^p(x^\star_{j+2})$, where p is such that $V(\hat{x}_{i+1}) = Q^p(\hat{x}_{i+1})$. Thus, in the case where $V(\hat{x}_i) \neq Q^l(\hat{x}_i)$, we must have $\hat{x}_{i+2} \in \mathbb{X}^\star$, completing the proof. ■

We can now state and prove our main result on the FUES.

Proposition 1 *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 $\hat{x}_i \in \mathbb{X}$.*

Proof. We will show that if claims 1. and 2. of the Proposition hold for all i, with $i \leq j$ for some j, then it will hold for all i with $i \leq j+1$. Thus by the principle of induction, the proposition will hold for all i with $i \leq j$ for any j. In particular, claims 1. and 2. of the Proposition will hold for $j = |\hat{\mathbb{X}}|$.

Assume items 1. and 2. of the Proposition are true for all i with $i \leq j$ for some j. Moreover, let $\hat{x}_{i_l}$ denote the sub-sequence of K points in $\mathbb{X}$ such i_l satisfies $i_l \leq j$.

Part 1: Proof of item 1.

We first show that if $\hat{x}_{j+1} \in \mathbb{X}$, then $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$ (claim 1. of the Proposition). Thus, let $\hat{x}_{j+1} \in \mathbb{X}$. There are two cases. The first case is if $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{\hat{x}_{j+1} - \hat{x}_{i_L}} \leq \frac{\hat{v}_{i_L} - \hat{v}_{i_L - 1}}{x_{i_L} - \hat{x}_{i_L - 1}}$ and the second case is if $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{\hat{x}_{j+1} - \hat{x}_{i_L}} > \frac{\hat{v}_{i_L} - \hat{v}_{i_{K-1}}}{\hat{x}_{i_L} - \hat{x}_{i_{K-1}}}$.

[Case I] Let $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{\hat{x}_{j+1} - \hat{x}_{i_L}} \leq \frac{\hat{v}_{i_L} - \hat{v}_{i_L - 1}}{x_{i_L} - \hat{x}_{i_L - 1}}$. Assume by contradiction that $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) \neq V(\hat{x}_{j+1})$ and let $m \in \mathbb{D}$ be such that $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = u(\sigma_m(\hat{x}_{j+1})) + G^m(f(\sigma_m(\hat{x}_{j+1}), \hat{x}_{j+1}))$. By Assumption 5, item 3., we will have:

$$V(\hat{x}_{i_L}) = u(\hat{c}_{i_L}) + G^m(f(\hat{x}_{i_L}, \hat{c}_{i_L}) \tag{42}$$

$$\hat{x}'_{j+1} = f(\sigma_q(\hat{x}_{j+1}), \hat{x}_{j+1}) \tag{43}$$

with $m \neq q$. Now, by Assumption 2 and using the reverse triangle inequality, we have:

$$|\hat{x}'_{j+1} - \hat{x}'_{i_L}| \geq \left| |\hat{x}'_{j+1} - f(\sigma_m(\hat{x}_{j+1}), \hat{x}_{j+1})| - |\hat{x}'_{i_L} - f(\sigma_m(\hat{x}_{j+1}), \hat{x}_{j+1})| \right| \tag{44}$$

$$\geq |D - M(\hat{x}_{j+1} - \hat{x}_{i_L})| \tag{45}$$

Dividing through, and noting Assumption 4, we get:

$$\frac{|\hat{x}'_{j+1} - \hat{x}'_{i_L}|}{\hat{x}_{j+1} - \hat{x}_{i_L}} \geq |\frac{D}{\hat{x}_{j+1} - \hat{x}_{i_L}} - M| \geq \frac{D}{\delta} - M \tag{46}$$

However, this yields a contradiction to $\hat{x}_{j+1} \in \mathbb{X}$ by lines 5-14 of Algorithm 1, implying $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$.

[Case II] Consider the second case, if $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{\hat{x}_{j+1} - \hat{x}_{i_L}} > \frac{\hat{v}_{i_L} - \hat{v}_{i_{K-1}}}{\hat{x}_{i_L} - \hat{x}_{i_{K-1}}}$, then by Claim 1, we must have that $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$.

Part 2: Proof of item 2.

Now we show that if $Q(\hat{c}_{j+1}, \hat{x}_{j+1}) = V(\hat{x}_{j+1})$, then $\hat{x}_{j+1} \in \mathbb{X}$ (claim 2. of the Proposition).

[Case I] First consider the case of $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{x_{j+1} - \hat{x}_{i_L}} \leq \frac{\hat{v}_{i_L} - \hat{v}_{i_L - 1}}{x_{i_L} - \hat{x}_{i_L - 1}}$. Suppose we have:

$$V(\hat{x}_{i_L}) = u(\hat{c}_{i_L}) + G^q(f(\hat{x}_{i_L}, \hat{c}_{i_L}) \tag{47}$$

$$\hat{x}'_{j+1} = f(\sigma_s(\hat{x}_{j+1}), \hat{x}_{j+1}) \tag{48}$$

$$V(\hat{x}_{j+1}) = u(\sigma_s(\hat{x}_{j+1})) + G^s(f(\hat{x}_{j+1}, \sigma_q(\hat{x}_{j+1}))) \tag{49}$$

and $s = q$. By Assumption 3 and Assumption 4, we have:

$$\frac{|\hat{x}'_{i_L} - \hat{x}'_{j+1}|}{|\hat{x}_{j+1} - \hat{x}_{i_L}|} \leq M < \frac{D}{\delta} - M \tag{50}$$

Thus, by Algorithm 1, we must have $\hat{x}_{j+1} \in \mathbb{X}$. Alternatively, suppose $s \neq q$, then there exists $\hat{x}_{j+w}$, with $1 \leq w \leq K$ such that $\hat{v}_{j+w} = Q^q(\hat{x}_{j+w})$. Since Q^q is concave, we must have that $\frac{\hat{v}_{j+w} - \hat{v}_{i_L}}{x_{j+w} - \hat{x}_{i_L}} \leq \frac{\hat{v}_{i_L} - \hat{v}_{i_L - 1}}{x_{i_L} - \hat{x}_{i_L - 1}}$. By Algorithm 1, lines lines 8-14, we retain the point $\hat{x}_{j+1}$ and $\hat{x}_{j+1} \in \mathbb{X}$.

[Case II] Now consider the second case of $\frac{\hat{v}_{j+1} - \hat{v}_{i_L}}{\hat{x}_{j+1} - \hat{x}_{i_L}} > \frac{\hat{v}_{i_L} - \hat{v}_{i_L - 1}}{x_{i_L} - \hat{x}_{i_L - 1}}$. By line 16. of Algorithm 1, we immediately have $\hat{x}_{j+1} \in \mathbb{X}$.

To conclude, we have shown that if claim 1. and claim 2. of the Proposition hold for all i with $i \leq j$, then it will hold for all i with $i \leq j+1$. Finally note that by Assumption $V(\hat{x}_i) = Q(\hat{c}_i, \hat{x}_i)$ for $i = 1, 2$ and $\hat{x}_i \in \mathbb{X}$ for $i \in \{0, 1\}$, thus the claim is true for $i = 2$. By the principle of induction, the claims of the Proposition hold for all i, completing the proof. ■

4 Conclusion

This paper provides a fast upper-envelope scan (FUES) method that computes the optimal value function for a dynamic optimization problem with discrete and continuous choices. FUES uses the observation that the upper-envelope of the value correspondence is only convex in regions where concave choice-specific value functions intersect. Accordingly, FUES removes all points that cause a jump in the policy function and do not form a convex region of the value correspondence.

FUES is a general method and can be applied efficiently with minimal coding time to dynamic programming problems without assumptions on monotonicity. We prove this method can accurately recover the optimal policy under the assumption the grid size is large enough relative to the smallest jump size between choice-specific value functions. This assumption is straightforward to satisfy in finite horizon models with finitely many choices. The infinite horizon models and models with infinitely many choices due to tastes shocks also display finitely many jumps in the policy functions. Nonetheless, formal conditions under which an infinite state space produces only finitely many discontinuities remain unknown. We also note that if a policy function does contain infinitely many arbitrarily small jump discontinuities, existing methods to solve discrete choice models that rely on identifying regions where policy functions overlap also face a challenge. A formal exploration on the nature of policy function discontinuities in infinite state space models with non-convexitites is thus an interesting further area of research.

5 Appendix: additional figures

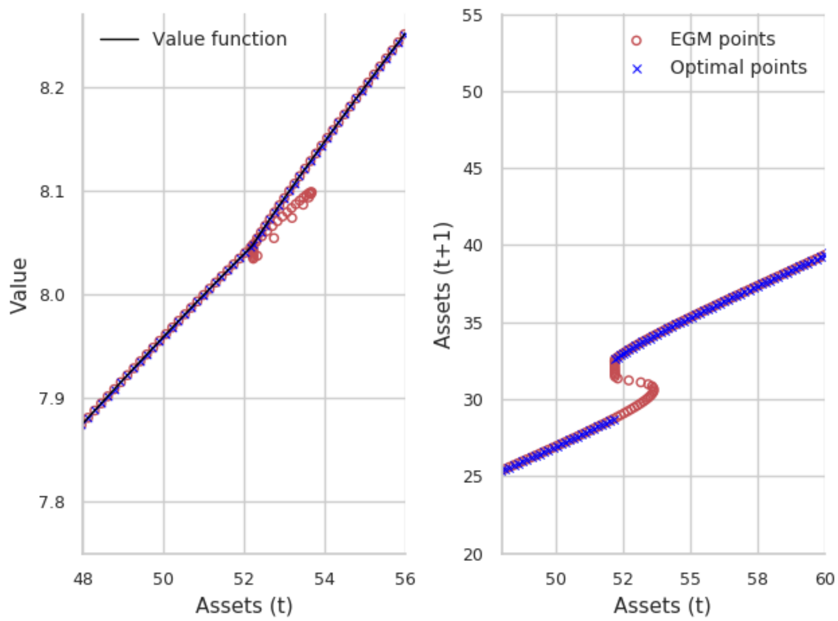

Figure 5: Value correspondence and optimal points for $t = 17$. Parameters from Iskhakov et al. (2017), Figure 4 with smoothing parameter to $\sigma = 0.5$.

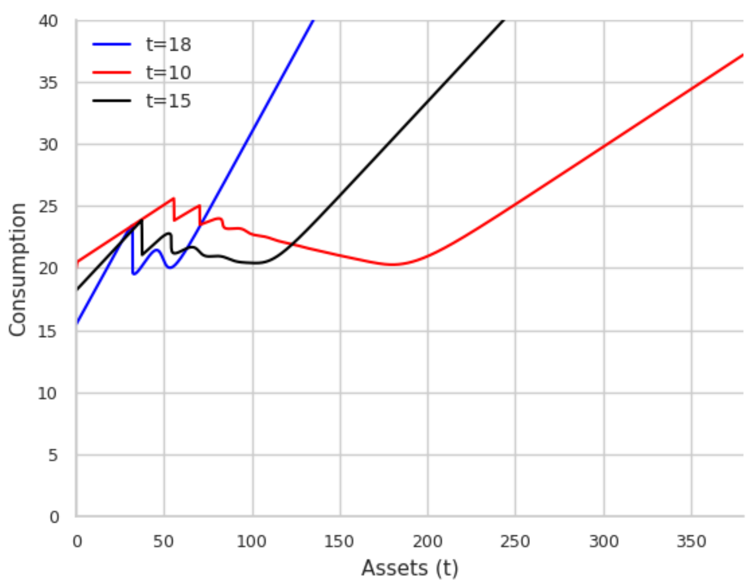

Figure 6: Optimal consumption functions with smoothing for workers.

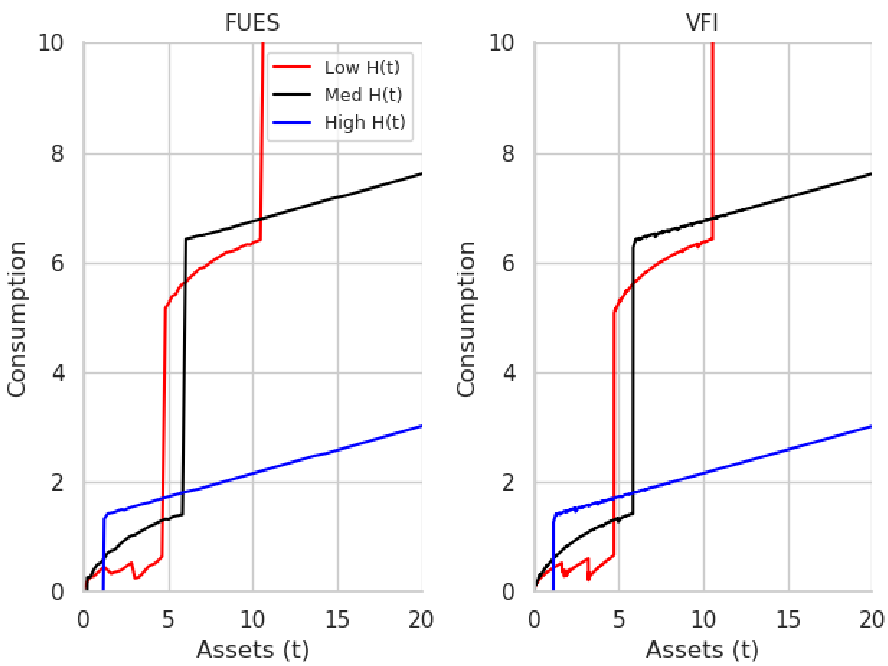

Figure 7: Consumption policy in the housing investment friction model (Application 2) using FUES compared to VFI. The plot involves the lowest income shock and three evenly spaced housing capital stock levels at the start of period t.

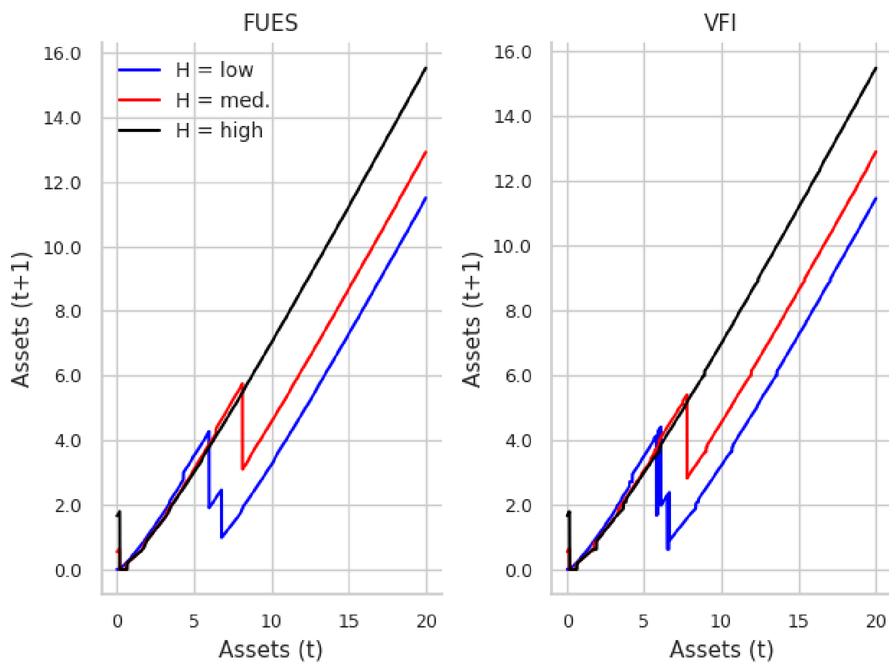

Figure 8: Liquid asset policy function in the infinite horizon housing choice model (Application 3) using FUES compared to VFI. The plot involves the lowest income shock and three evenly spaced housing capital stock levels at the start of period t.

References

Attanasio, O., Levell, P., Low, H., and Sanchez-Marcos, V. (2018). Aggregating elasticities: Intensive and extensive margins of women's labor supply. *Econometrica*, 86(6):2049–2082.

Bertsekas, D. (2022). *Abstract dynamic programming*. Athena Scientific.

Carroll, C. D. (2006). The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics Letters*, 91(3):312–320.

Coleman, W. J. (1990). Solving the stochastic growth model by policy-function iteration. *Journal of Business and Economic Statistics*, 8(1):27–29.

Cooper, I. (2006). Asset pricing implications of nonconvex adjustment costs and irreversibility of investment. *Journal of Finance*, 61(1):139–170.

Dobrescu, L., Shanker, A., Bateman, H., Newell, B. R., and Thorp, S. (2022). Retirement Eggs and Retirement Baskets.

Dobrescu, L. I., Fan, X., Bateman, H., Newell, B. R., and Ortmann, A. (2016). RETIREMENT SAVINGS: A TALE OF DECISIONS AND DEFAULTS. *The Economic Journal*, 128(2016):1047–1094.

Druedahl, J. and Jørgensen, T. H. (2017). A general endogenous grid method for multi-dimensional models with non-convexities and constraints. *Journal of Economic Dynamics and Control*, 74:87–107.

Fagereng, A., Holm, M. B., Natvik, G., and Moll, B. (2019). Saving Behavior Across the Wealth Distribution : The Importance of Capital Gains. *NBER Working Papers, No 26588*.

Fella, G. (2014). A generalized endogenous grid method for non-smooth and. *Review of Economic Dynamics*, 17:329–344.

Graham, R. L. (1972). An efficient algorithm for determining the convex hull of a finite planar set. *Inf. Process. Lett.*, 1:132–133.

Iskhakov, F. (2015). Multidimensional endogenous gridpoint method: Solving triangular dynamic stochastic optimization problems without root-finding operations. *Economics Letters*, 135:72–76.

Iskhakov, F., Jørgensen, T. H., Rust, J., and Schjerning, B. (2017). The endogenous grid method for discrete-continuous dynamic choice models with (or without) taste shocks. *Quantitative Economics*, 8(2):317–365.

Jeyakumar, V., Rubinov, A. M., and Wu, Z. Y. (2007). Non-convex quadratic minimization problems with quadratic constraints: global optimality conditions. *Mathematical Programming*, 110(3):521–541.

Jeyakumar, V. and Srisatkunarajah, S. (2009). Lagrange multiplier necessary conditions for global optimality for non-convex minimization over a quadratic constraint via S-lemma. *Optimization Letters*, 3(1):23–33.

Kaplan, G., Mitman, K., and Violante, G. L. (2020). The housing boom and bust: Model meets evidence. *Journal of Political Economy*, 128(9):3285–3345.

Kaplan, G. and Violante, G. L. (2014). A Model of the Consumption Response to Fiscal Stimulus Payments. *Econometrica*, 82(4):1199–1239.

Khan, A. and Thomas, J. K. (2008). Idiosyncratic shocks and the role of nonconvexities in plant and aggregate investment dynamics. *Econometrica*.

Krusell, P. and Smith, A. a. (1998). Macroeconomy Per Krusell. *Journal of Political Economy*, 106(5):867–896.

Laibson, D., Maxted, P., and Moll, B. (2020). Present Bias Amplifies the Household Balance-Sheet Channels of Macroeconomic Policy. pages 1–57.

Midrigan, V. (2011). Menu costs, multiproduct firms, and aggregate fluctuations. *Econometrica*, 79(4):1139–1180.

Rust, J. (1987). Optimal Replacement of GMC Bus Engines: An Empirical Model of Harold Zurcher. *Econometrica*, 55(5):999–1033.

Skiba, A. K. (1978). Optimal Growth with a Convex-Concave Production Function. *Econometrica*, 46(3):527–539.

Yogo, M. (2016). Portfolio choice in retirement: Health risk and the demand for annuities, housing, and risky assets. *Journal of Monetary Economics*, 80:17–34.