A Simple Upper-Envelope Method for Non-Concave EGM[1]

Jesús Bueren
Católica Lisbon School of Business and Economics

Abstract

In non-concave dynamic optimization problems, the Euler equation is no longer sufficient for optimality, so Endogenous Grid Method (EGM) points that satisfy the first-order condition need not be globally optimal. Existing approaches therefore supplement EGM with an upper-envelope step that can be computationally costly. This paper proposes a simple algorithm that detects non-monotonicities in the endogenous grid and uses local linear approximations to discard dominated points without relying on value function iteration. In the retirement model of Iskhakov et al. (2017), the method is fast, simple to implement, and numerically accurate.

[1]The author thanks Giulio Fella for feedback at an early stage of this project. Email: jbueren@ucp.pt.

1 Introduction

Dynamic stochastic optimization problems are central to many quantitative and structural models in economics, but they can rarely be solved analytically. Researchers must therefore rely on numerical methods, often solving the same model many times in calibration, estimation, and counterfactual analysis. Value function iteration (VFI) remains the standard benchmark because of its simplicity and broad applicability, but its computational cost can quickly become burdensome.

In concave economic problems, computational efficiency can be gained using the Euler equation which is a necessary and sufficient condition for optimality. In this class of problems, Endogenous Grid Methods (EGM) proposed by Carroll (2006) can greatly accelerate the speed and accuracy of the solution. In a nutshell, EGM inverts the Euler equation to solve analytically the state variable as a function of the policy. As such the method delivers an endogenous grid of the state variable which maps with the exogenous policy grid.

Beyond the baseline one-control concave setting of Carroll (2006), Barillas and Fernandez-Villaverde (2007) combine EGM with value function iteration to handle smooth problems with multiple controls. In non-concave problems, however, the resulting policy from EGM is not a bijective function but a correspondence, given that for a given value of the state, multiple policies solve the Euler equation. Fella (2014) extended EGM to this class of problems by supplementing the EGM step with a VFI step to verify whether the policy is not only a local but also a global maximum over the non-concave region.

Methods based on EGM for non-concave problems are already widely used in the structural and quantitative macro literature. There are now many applications, including models of long-term care and late-life saving (Ameriks et al., 2020), life-cycle labor supply, retirement, and pension policy (Iskhakov and Keane, 2021), female labor supply, human capital accumulation, and welfare reform (Blundell et al., 2016), and precautionary borrowing and the credit card debt puzzle (Druedahl and Jørgensen, 2018). Because these applications require solving the model many times during estimation, a faster upper-envelope step can substantially reduce computational cost.

In this paper, I propose a simple local upper-envelope routine for non-concave EGM problems that avoids the VFI step in Fella (2014). The key observation is that non-concavity shows up in the raw EGM output as local reversals in the endogenous grid: when nearby states admit more than one Euler-equation solution, the grid can fold back on itself and some of the associated points cannot belong to the true upper envelope. The algorithm uses this local pattern to identify the problematic region. At each reversal, it takes a first-order Taylor approximation of the value function on each side of the reversal and uses the intersection of the two approximations to locate the point where the optimal policy switches. It then discards the dominated points between the two surviving sides and inserts two points around the approximated switching location. Repeating this local repair step yields a cleaned endogenous grid without global sorting or value function iteration.

The proposed algorithm is related to the monotone segment selection (MSS) method of Iskhakov et al. (2017). MSS addresses the same upper-envelope problem, but it does so more globally. It first breaks the raw endogenous-grid correspondence into monotone pieces, then compares these pieces to determine which one delivers the highest value at each level of cash on hand. Whenever the dominant piece changes, MSS computes the crossing point and inserts it into the refined grid. Relative to the local Taylor routine proposed here, MSS is more computationally costly: sorting the sections is $O(J \log J)$ in the worst case, and repeatedly comparing active sections over the pooled breakpoint grid means that total work rises with the number of non-monotonicities.

The algorithm is also related to the Fast Upper-Envelope Scan (FUES) of Dobrescu and Shanker (2023). Like the method proposed here, FUES avoids a VFI step and instead refines the raw EGM output directly. Its approach, however, is different. FUES first sorts the endogenous grid by cash on hand and then scans the ordered points to detect jumps and discard dominated observations using information from both the value and policy functions. A distinctive feature of FUES is that it relies on a policy-gradient threshold $\bar{m}$ and forward and backward scans of depth $\bar{L}$ to decide whether nearby points belong to the same branch of the correspondence and to approximate where the upper envelope changes. These tuning parameters must be calibrated to the problem at hand. By contrast, the algorithm proposed here requires neither sorting nor free parameters. It relies only on the observation that non-concavity generates local reversals in the endogenous grid and then uses local Taylor approximations to repair the grid around each reversal.

To evaluate these alternatives, I run a horse race between the proposed Taylor routine, MSS, and FUES in the same retirement model. The comparison focuses on the two dimensions that matter most in practice: computational speed and numerical accuracy. The results show that all three methods are numerically precise, that FUES is substantially faster than MSS, and that the proposed Taylor routine is faster still, delivering speed gains of about 2.7× to 7.0× relative to MSS and about 10% to 32% relative to FUES in the reported calibrations.

The next section presents the retirement choice model of Iskhakov et al. (2017) and shows how the proposed upper-envelope routine fits into the EGM solution. The numerical section then compares the proposed method with MSS and FUES in terms of speed and numerical accuracy.

2 The Retirement Choice Model

2.1 The Model

Consider a deterministic life-cycle problem in which an agent with finite horizon T chooses consumption c_t and a retirement decision d_t in each period:

$$\max_{\{c_t,d_t\}_{t=1}^{T}} \sum_{t=1}^{T} \beta^t \Big(\log(c_t) - \delta d_t \Big) \tag{1}$$
$$\text{s.t.} \quad c_t \leq M_t$$
$$M_t = R(M_{t-1} - c_{t-1}) + y d_{t-1}$$

Let $d_t = 0$ denote retirement and $d_t = 1$ denote working. The agent receives labor income y when working and no labor income when retired. Savings earn gross return R. Period utility is $\log(c_t) - \delta d_t$, so working carries a utility cost δ. The state variable M_t denotes cash on hand at the beginning of period t. Retirement is absorbing, so once the agent retires, returning to work is not allowed.

Let $V_t(M)$ denote the value function of an agent who enters period t as a worker, and let $W_t(M)$ denote the value function of an agent who enters period t already retired. The worker's value function is

$$V_t(M) = \max\{v_t(M,0), v_t(M,1)\} \tag{2}$$

where $v_t(M,d)$ is the choice-specific value associated with retirement choice d:

$$v_t(M,0) = \max_{c \leq M}\{\log(c) + \beta W_{t+1}(R(M-c))\} \tag{3}$$
$$v_t(M,1) = \max_{c \leq M}\{\log(c) - \delta + \beta V_{t+1}(R(M-c)+y)\} \tag{4}$$

2.2 Optimality Conditions

Consider first the case in which a worker chooses to keep working. The corresponding Euler equation is

$$u'(c_t) = \beta R u'(c_{t+1}(R(M-c)+y)) \tag{5}$$
$$\frac{1}{c} = \beta R \frac{1}{c_{t+1}\Big(R(M-c)+y\Big)},$$

where $c_{t+1}(M)$ denotes next period's consumption policy when the agent remains a worker. If instead the worker retires in period t, the Euler equation is

$$u'(c_t) = \beta R u'(c^r_{t+1}(R(M-c))) \tag{6}$$
$$\frac{1}{c} = \beta R \frac{1}{c^r_{t+1}\Big(R(M-c)\Big)}$$

Given the period-$t+1$ policy functions, equations (5) and (6) characterize the period-t choice-specific policies $c_t(M,1)$ and $c_t(M,0)$, respectively.

2.3 EGM step

The EGM step solves these Euler equations by working on an exogenous grid for end-of-period savings, $A = M - c$. When the inverse marginal utility function is available in closed form, the Euler equation can be inverted directly to recover current consumption at each grid point A_j. This is the key idea of the endogenous grid method of Carroll (2006).

Period: T

As in standard life-cycle problems, the model is solved by backward induction. In the terminal period T, the agent consumes all available resources, so $c_T(M, d) = M$. Because working carries a positive utility cost, the agent never chooses to work in the last period, that is, $d_T(M) = 0$. For an already retired agent, the terminal policy is likewise $c_T^r(M) = M$.

Period: $T - 1$

In period $T - 1$, the Euler equations determine the choice-specific consumption rules on an exogenous savings grid $\vec{A} = \{A_1, \ldots, A_J\}$. For each savings grid point A_j, the implied choice-specific consumption rules are

$$c_{T-1}(M_j, 1) = \frac{1}{\beta R} c_T\big(RA_j + y\big),$$
$$c_{T-1}(M_j, 0) = \frac{1}{\beta R} c_T^r\big(RA_j\big),$$

where the associated endogenous-grid point is $M_j = A_j + c_{T-1}(M_j, d)$. At these endogenous-grid points, the Euler equation holds exactly. Between them, the policy functions are constructed by interpolation. Because the continuation problem depends on both age and retirement status, the endogenous grid also depends on t and on the discrete choice d.

The EGM step must also be completed with the borrowing-constraint region. Evaluating the Euler equation at zero savings, $A_j = 0$, yields a threshold $\underline{M}$ at which the interior Euler equation holds. For all $M \in (0, \underline{M})$, however, the budget constraint binds and the correct policy is $c(M) = M$. In practice, it is therefore enough to add the point $(0, 0)$ to the endogenous grid. Under linear interpolation, this guarantees that the policy on $(0, \underline{M})$ coincides with the constrained rule $c(M) = M$.

Period: $T - 2$

By period $T - 2$, the Euler equation is no longer sufficient for optimality. The problem is that the worker value function $V_{T-1}(M)$ is not concave around the threshold $\bar{M}_{T-1}$ at which the agent is indifferent between working and retiring. As a result, the solution $c_{T-2}(M, d)$ from EGM is no longer a single-valued policy function but a policy correspondence: for some values of M, multiple consumption choices satisfy the

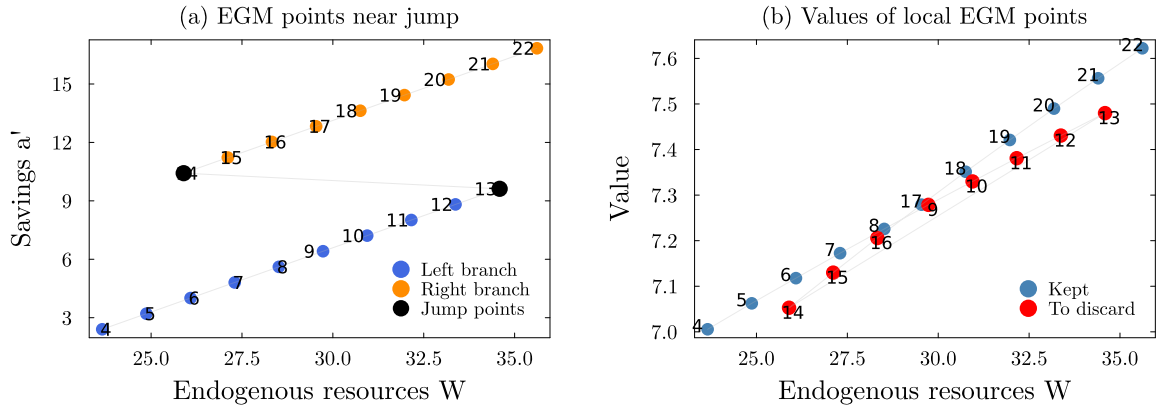

Figure 1: Local EGM points around the backward jump and their associated values

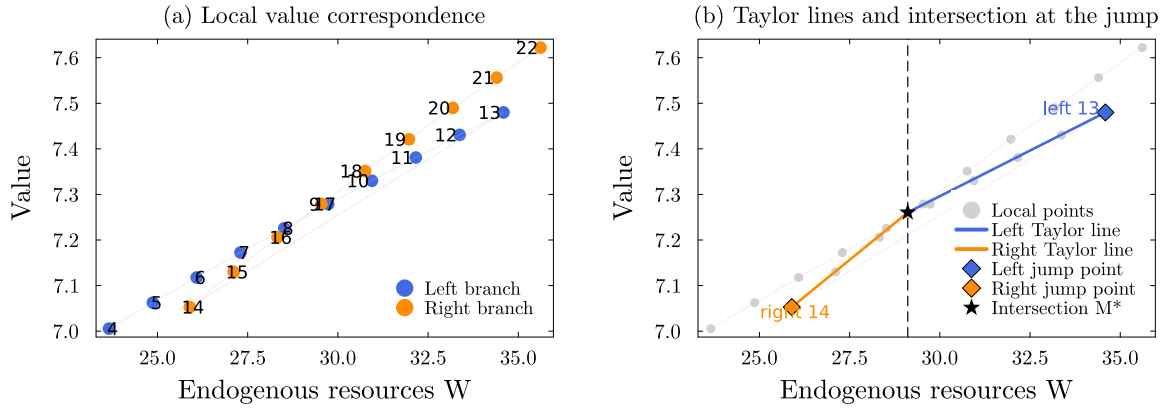

Figure 2: Local value correspondence and the two Taylor lines used to locate the intersection point $M^\star$

Euler equation. The EGM step must therefore be complemented by an upper-envelope procedure that discards points that are only local optima.

2.4 Constructing the Upper-Envelope

Figure 1 shows the local region of the endogenous grid where the first backward jump appears, together with the value associated with those same EGM points. The main implication is that a backward jump between points 13 and 14 signals that the policy correspondence contains points that satisfy the Euler equation locally but do not belong to the upper envelope of the value function.

This is visible in the right panel of Figure 1. Some endogenous-grid points are dominated by nearby alternatives that deliver a higher value at a lower level of W. For example, point 15 is dominated by point 7. The EGM step must therefore be followed by a refinement step that removes these non-optimal points.

Suppose the algorithm detects a backward jump such that $W_j > W_{j+1}$. The two adjacent jump points define the local anchors of the competing branches, and the algorithm

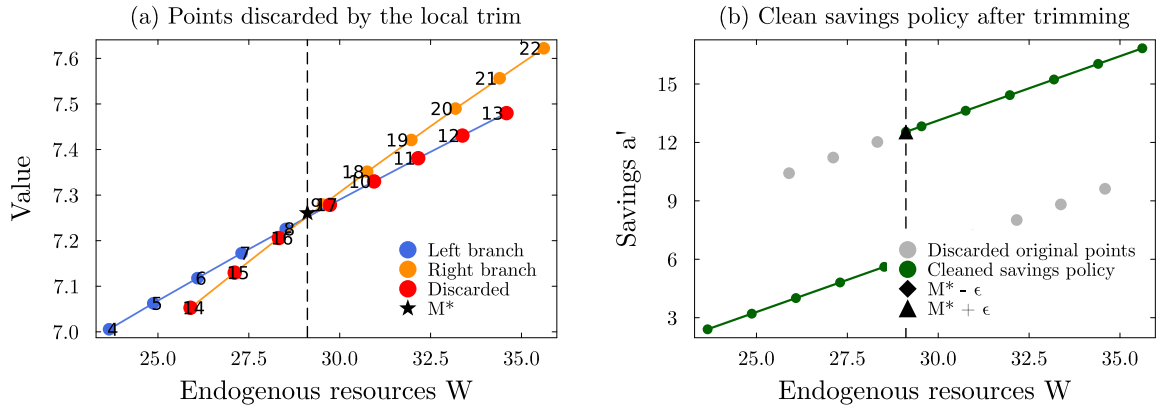

Figure 3: Local trimming around the jump and the clean savings policy after inserting the switching pair

constructs the first-order Taylor approximations

$$\hat{V}_L(W) = V(W_j) + u_c\big(c(W_j)\big)\big(W - W_j\big),$$
$$\hat{V}_R(W) = V(W_{j+1}) + u_c\big(c(W_{j+1})\big)\big(W - W_{j+1}\big).$$

Because the endogenous grid bends backward at the jump, savings are higher at W_{j+1} than at W_j even though $W_{j+1} < W_j$. It follows that consumption must be lower at W_{j+1} than at W_j, and by strict concavity of utility the marginal utility of consumption must therefore be higher. Hence the first-order Taylor expansion around W_{j+1} is locally steeper than the first-order Taylor expansion around W_j. Their intersection defines the candidate local switching point

$$M^\star = \frac{\alpha_R - \alpha_L}{u_c(c(W_j)) - u_c(c(W_{j+1}))}, \qquad \alpha_s = V(W_s) - u_c(c(W_s))W_s,$$

with common value $V^\star = \hat{V}_L(M^\star) = \hat{V}_R(M^\star)$.

Once $M^\star$ has been computed, the algorithm trims the jump locally. On the left side, it moves backward from j until it reaches the first point with $W \le M^\star$. On the right side, it moves forward from $j+1$ until it reaches the first point with $W \ge M^\star$. Every original point strictly between these two surviving boundaries is discarded.

The algorithm then reconstructs the switch between the two surviving branches by linearly extrapolating consumption to $M^\star - \varepsilon$ from the left branch and to $M^\star + \varepsilon$ from the right branch. Both inserted points are assigned the common value $V^\star$. The dominated block is deleted, the crossing pair is inserted, and the scan resumes from the repaired location.

Figures 2 and 3 summarize the procedure. Figure 2 shows the two local Taylor lines constructed at the jump and their intersection $M^\star$. Figure 3 then shows which original points are discarded and how the cleaned savings policy is obtained after the switching pair has been inserted.

The trimming step therefore stops when the left scan first reaches a point with $W \le M^\star$

and the right scan first reaches a point with $W \geq M^\star$. At that point, the algorithm deletes the dominated block, inserts the crossing pair, and builds the final policy and value interpolants on the refined monotone grid.

Procedure 1 summarizes the trimming step in pseudo-code.

Procedure 1 Discard non-optimal points

Input: Endogenous grid W_{endo}, consumption values $c(W_{endo})$, and value function values $v(W_{endo})$

$x \leftarrow 1$
while $x <$ length(W_{endo}) **do**
  if $W_{endo}[x+1] < W_{endo}[x]$ **then** ▷ Backward jump detected
    $s_L \leftarrow U_c(c(W_{endo}[x])), \quad s_R \leftarrow U_c(c(W_{endo}[x+1]))$ ▷ Taylor slopes at jump
    $\alpha_L \leftarrow v(W_{endo}[x]) - s_L W_{endo}[x]$ ▷ Left intercept
    $\alpha_R \leftarrow v(W_{endo}[x+1]) - s_R W_{endo}[x+1]$ ▷ Right intercept
    $M^\star \leftarrow (\alpha_R - \alpha_L)/(s_L - s_R)$ ▷ Intersection point
    $V^\star \leftarrow s_L M^\star + \alpha_L$ ▷ Common value at $M^\star$
    $\ell \leftarrow x$
    while $\ell \geq 1$ and $W_{endo}[\ell] > M^\star$ **do** ▷ Trim left branch
      $\ell \leftarrow \ell - 1$
    end while
    $r \leftarrow x + 1$
    while $r \leq$ length(W_{endo}) and $W_{endo}[r] < M^\star$ **do** ▷ Trim right branch
      $r \leftarrow r + 1$
    end while
    Delete points $\ell + 1, \ldots, r - 1$ ▷ Remove dominated block
    $c_L \leftarrow$ extrapolate left branch to $M^\star - \varepsilon$
    $c_R \leftarrow$ extrapolate right branch to $M^\star + \varepsilon$
    Insert $(M^\star - \varepsilon, c_L, V^\star)$ ▷ Left switch point
    Insert $(M^\star + \varepsilon, c_R, V^\star)$ ▷ Right switch point
    $x \leftarrow \max(\ell + 1, 1)$ ▷ Resume at repaired region
  else
    $x \leftarrow x + 1$ ▷ No jump here
  end if
end while

Output: Refined monotone grid W^*_{endo} and interpolants $c(W^*_{endo})$, $v(W^*_{endo})$

3 Numerical Accuracy

To compare the proposed routine with the main alternative upper-envelope refinements in the literature, I solved the same deterministic retirement model with three upper-envelope steps: the Taylor upper-envelope routine developed in this paper, a Julia implementation of the polyline-based MSS construction used in the Matlab code distributed with Iskhakov et al. (2017), and a Julia implementation of the FUES scan of Dobrescu and Shanker (2023). The polyline routine splits the policy correspondence into monotone sections, sorts each section, evaluates the piecewise-linear upper enve-

lope on the pooled breakpoint grid, and inserts the implied switching point whenever the upper branch changes. The FUES implementation instead sorts the endogenous grid and applies the scan rule with policy-gradient threshold $\bar{m} = 1$ and look-ahead / look-back window $\bar{L} = 4$. Appendices A and B provide additional intuition for the spirit in which MSS and FUES work.

Table 1 reports how speed and dense-grid Euler residuals change with the number of asset-grid points J. For each J, the code solves the model with all three upper-envelope methods, evaluates the worker Euler residual on a dense grid of 5,000 cash-on-hand points spanning $(0.01, A_{\max})$, and drops constrained points where the worker Euler equation does not bind. For each dense-grid point, c^{impl} denotes the consumption value implied by the Euler equation given next period's numerical policy, and the residual is measured as $|c^{\text{impl}}/c - 1|$. The reported accuracy statistic is the mean of $\log_{10} |c^{\text{impl}}/c - 1|$ at selected horizons, so more negative values correspond to smaller numerical errors. The main message of the table is that all three methods are numerically precise. The difference lies in speed. FUES is already much faster than the polyline MSS benchmark, but the proposed Taylor routine is faster still. Relative to the polyline baseline, current median speedups are about 2.7× and 2.4× at $J = 250$, 6.4× and 5.2× at $J = 500$, 6.8× and 5.7× at $J = 1000$, and 7.0× and 5.3× at $J = 2000$, for Taylor and FUES respectively. Overall, the numerical evidence indicates that the proposed Taylor routine delivers accuracy comparable to the alternatives while being the fastest method in this calibration.

Table 1: Comparison of upper-envelope methods

J	Method	Median ms	Rel. speed	$\tau = 20$	$\tau = 10$	$\tau = 5$	$\tau = 1$
250	Polyline MSS	4.85	1.0x	-13.25	-15.19	-15.56	-15.79
250	FUES	2.05	2.36x	-14.07	-15.34	-15.56	-15.79
250	Taylor upper envelope	1.82	2.66x	-13.74	-15.65	-15.59	-15.79
500	Polyline MSS	21.98	1.0x	-15.4	-15.64	-15.67	-15.79
500	FUES	4.2	5.23x	-15.71	-15.66	-15.7	-15.79
500	Taylor upper envelope	3.43	6.4x	-15.82	-15.68	-15.7	-15.79
1000	Polyline MSS	51.88	1.0x	-15.74	-15.65	-15.74	-15.79
1000	FUES	9.08	5.71x	-15.82	-15.7	-15.75	-15.79
1000	Taylor upper envelope	7.65	6.78x	-15.82	-15.7	-15.75	-15.79
2000	Polyline MSS	104.77	1.0x	-15.79	-15.73	-15.77	-15.8
2000	FUES	19.78	5.3x	-15.81	-15.75	-15.78	-15.8
2000	Taylor upper envelope	15.06	6.96x	-15.81	-15.75	-15.78	-15.8

Note: The table compares the Taylor upper envelope, the polyline MSS routine, and the FUES scan. For each grid size, solve times report the median over 10 full backward-induction runs. Accuracy is summarized by the mean of $\log_{10} |c^{\text{impl}}/c - 1|$ on a dense grid of 5,000 cash-on-hand points spanning $(0.01, A_{\max})$ at selected horizons, excluding constrained points where the worker Euler equation does not bind. Here c^{impl} denotes the consumption value implied by the Euler equation given next period's numerical policy. More negative numbers indicate smaller errors.

4 Conclusion

This paper proposes a simple local upper-envelope routine for non-concave EGM problems that removes dominated endogenous-grid points without relying on the value function iteration step used in Fella (2014). Using the retirement choice model of Iskhakov et al. (2017), I show how backward jumps in the endogenous grid can be used to identify the problematic region and how local Taylor approximations can be used to repair it. The numerical exercise shows that the proposed method is substantially faster than the polyline MSS benchmark while delivering very similar accuracy, and that its performance is also competitive with FUES. Overall, the results suggest that the proposed routine provides a simple and computationally efficient refinement step for solving non-concave dynamic choice problems with EGM.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process

During the preparation of this work the author used OpenAI Codex in order to assist with editing, rewriting, formatting, coding, and code-based figure and manuscript preparation tasks. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

References

Ameriks, J., Briggs, J. S., Caplin, A., Shapiro, M. D., and Tonetti, C. (2020). Long-term-care utility and late-in-life saving. *Journal of Political Economy*, 128(6):2375–2451.

Barillas, F. and Fernandez-Villaverde, J. (2007). A generalization of the endogenous grid method. *Journal of Economic Dynamics and Control*, 31(8):2698–2712.

Blundell, R., Costa Dias, M., Meghir, C., and Shaw, J. (2016). Female labor supply, human capital, and welfare reform. *Econometrica*, 84(5):1705–1753.

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

Dobrescu, L. I. and Shanker, A. (2023). *A fast upper envelope scan method for discrete-continuous dynamic programming*. CEPAR, ARC Centre of Excellence in Population Ageing Research.

Druedahl, J. and Jørgensen, C. N. (2018). Precautionary borrowing and the credit card debt puzzle. *Quantitative Economics*, 9(2):785–823.

Fella, G. (2014). A generalized endogenous grid method for non-smooth and non-concave problems. *Review of Economic Dynamics*, 17(2):329–344.

Iskhakov, F., Jørgensen, T., 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.

Iskhakov, F. and Keane, M. (2021). Effects of taxes and safety net pensions on life-cycle labor supply, savings and human capital: The case of australia. *Journal of Econometrics*, 223(2):401–432.

A The Polyline MSS Upper Envelope

This appendix summarizes the spirit of the polyline monotone-segment selection (MSS) algorithm implemented in the Julia routine. The purpose of the method is to start from the EGM policy correspondence, split it into sections that are monotone in cash on hand m, and then recover the piecewise-linear upper envelope implied by those sections.

The same local example used in Figures 1 and 2 is helpful for building this intuition, so Figure A1 focuses directly on the key polyline step. The colored lines are the monotone sections obtained after splitting the local value correspondence at its loopback points. Because each section is monotone in m, it can be treated as a well-defined piecewise-linear candidate value function on its own domain.

The polyline MSS algorithm differs from the Taylor routine in how it resolves this geometry. Rather than approximating the switch with two local first-order Taylor lines at the jump, the code compares the monotone sections directly. It pools the breakpoints from all sections, evaluates every active section at those common m-values, and records which section attains the highest value. Whenever the identity of the upper section changes between two consecutive breakpoints, the code solves for the intersection of the two relevant linear pieces in closed form and inserts that crossing into the envelope. In Figure A1, the black line is the resulting polyline upper envelope and the star marks the inserted crossing where the dominant section changes. Repeating this step across the full grid identifies the points that must be discarded and yields the refined value correspondence used to reconstruct the policy function.

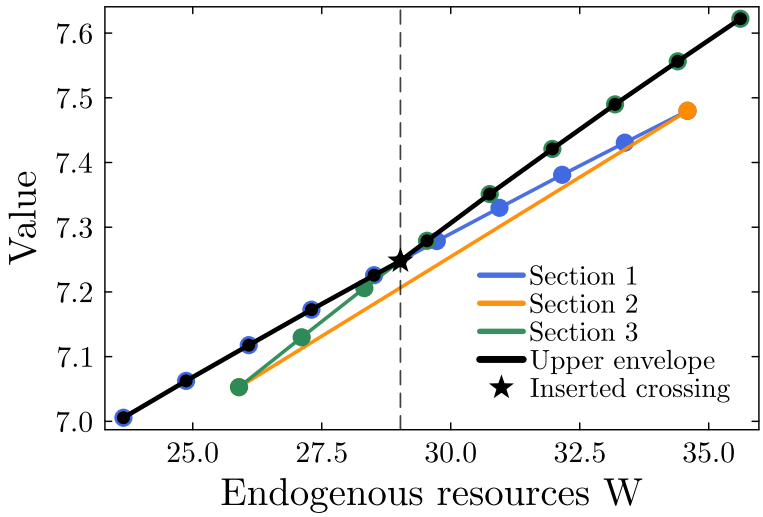

Figure A1: Monotone sections and polyline upper envelope for the same local jump example used in the main text

B The FUES Upper-Envelope Scan

This appendix gives a more concrete description of the Fast Upper-Envelope Scan (FUES) of Dobrescu and Shanker (2023). In contrast to the local Taylor routine, FUES first reorders the raw EGM points by cash on hand M. After sorting, the value correspondence is read as a zig-zag polyline, and the algorithm scans that ordered sequence from left to right while keeping track of the last two retained points on the provisional envelope.

At a generic scan step, let k and j denote the last two retained points and let q be the next candidate point in the sorted sequence. FUES compares the secant slope from k to j with the secant slope from j to q. At the same time, it checks whether moving from j to q implies a large change in the savings policy $a'(M) = M - c(M)$. In the implementation used for the numerical comparison, this jump diagnostic is controlled by the policy-gradient cutoff $\bar{m} = 1$: if the implied savings slope between two adjacent retained/candidate points exceeds one, the code treats that movement as evidence that the scan has crossed from one branch of the correspondence to another. Thus, $\bar{m} = 1$ and the look-ahead / look-back window $\bar{L} = 4$ reported in the main text are the actual parameter values used in the Julia implementation.

The economic classification rule is then local. If the move from j to q is both a policy jump and a concave right turn in the value correspondence, FUES treats q as dominated and discards it. If the move is a policy jump but creates a convex left turn, FUES treats q as a possible post-crossing point on the upper envelope. The forward and backward scans of depth $\bar{L} = 4$ then provide a short local check for whether the scan has genuinely switched to a different branch: the forward scan looks for a later point that appears to stay on the same branch as j, while the backward scan looks for an earlier discarded point that appears to lie on the same branch as q. These checks decide whether q should remain on the refined set or whether the previously retained point j should instead be removed.

Figure B1 uses the same local jump example as the main-text trimming figures, but now reordered in the way FUES sees it after sorting by cash on hand. The gray zig-zag line is the sorted local value correspondence generated by that same neighborhood of EGM points. The red markers indicate points that the scan discards as dominated, the blue markers indicate the points retained by the scan, and the black line joins those retained points to form the refined upper envelope.

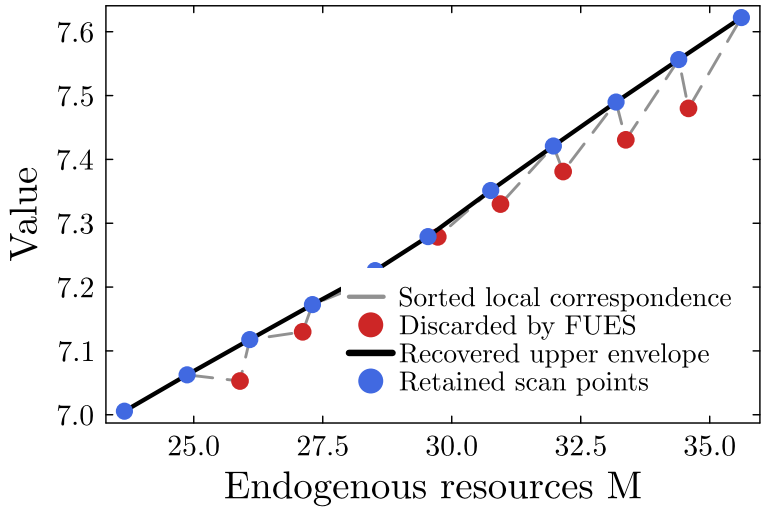

Figure B1: FUES on the same local jump example used in the main text: after sorting the local correspondence by cash on hand, the scan removes dominated points and retains the black upper envelope