

The Sequential Endogenous Grid Method

Alan Lujan^{1,2}  

¹Krieger School of Arts and Sciences, Johns Hopkins University , ²Econ-ARK

Key Points

- Splitting a decision into stages lets each qualifying stage invert its first-order condition in closed form.
- On the two-asset household block, EGMⁿ solves about thirteen times faster in serial than a matched nested-EGM implementation.
- Swapping in ENGINE for the curvilinear interpolation method cuts full-solve time by a factor of roughly three to four and a half.

Abstract Heterogeneous-agent models with multiple decisions face a computational dilemma. Joint optimization over all choices is slow; strong separability restrictions speed computation but rule out economically interesting interactions. I introduce Sequential EGM (EGMⁿ), which eases this tension by decomposing the problem into sequential stages so that each stage can exploit the Endogenous Grid Method's closed-form inversion, chaining speed gains across decisions wherever separable utility or an invertible transition permits it. The resulting curvilinear grids require interpolation that respects their structure, so I develop ENGINE (ENDogenous Grid INTERpolation and Extrapolation), which exploits the index structure inherited from exogenous grids, with no construction step and no triangulation. In the canonical liquid-illiquid household block, EGMⁿ solves an order of magnitude faster than the nested method in serial and 3.7 to 7.9 times faster at sixteen threads. Replacing existing curvilinear interpolation with ENGINE cuts full-solve time 2.8- to 4.5-fold at matched accuracy.

Keywords endogenous grid method, dynamic programming, interpolation, computational economics

JEL codes C61, C63, E21

Materials

Code
alanlujan91/
sequential_egm
MIT license

Working paper

October 10, 2026

Correspondence

Alan Lujan
alujan@jhu.edu

Funding

This work was supported by the Alfred P. Sloan Foundation through the Econ-ARK project. Econ-ARK (g-2017-9832).

License



© 2026 Alan Lujan. This article is distributed under the terms of the Creative Commons Attribution Non Commercial No Derivatives 4.0 International license, which enables reusers to copy and distribute the material in any medium or format in unadapted form only, for noncommercial purposes only, and only so long as attribution is given to the creator.

1 Introduction

Solving heterogeneous-agent models with multiple simultaneous decisions has traditionally required either joint optimization, which is slow, or restrictive assumptions about preferences that limit economic relevance. These heterogeneous-agent models, in the tradition of Huggett (1993), Aiyagari (1994), and Krusell and Smith (1998), are the workhorses of macroeconomic analysis (Krueger et al. 2016), and increasingly they ask the household to make several decisions at once, such as labor supply, portfolio choice, or illiquid saving, the setting in which joint optimization grows costly. The Endogenous Grid Method (EGM) transformed computation of consumption-saving problems by inverting first-order conditions, but extending that insight to multiple continuous choices has proved difficult. The leading extensions either constrain the problem class or

keep a numerical solve inside the iteration ([Iskhakov 2015](#); [Drue Dahl and Jrgensen 2017](#); [Drue Dahl 2021](#)).

A household choosing consumption and labor supply makes both decisions with the same information. Yet the solver need not treat them jointly. I introduce Sequential EGM (EGM^n , whose superscript records the chain of stagewise inversions), which orders the problem into stages and applies an EGM inversion wherever the stage permits one. Stages without an inversion fall back to lower-dimensional optimization.

Each stage of an EGM^n decomposition applies EGM when its first-order condition and its recovery map, the composite of the inverted condition with the budget identities, satisfy the inversion and optimality conditions developed in the paper. The recovery map carries each post-decision gridpoint to its endogenous pre-decision state, as $m(a) = a + c(a)$ does in one-dimensional EGM, where a is end-of-period saving, c the consumption the inverted Euler equation assigns to it, and m the market resources from which that choice is optimal. Throughout, choices are continuous and satisfy standard constraint qualifications. EGM^n requires stronger stage structure than nested approaches that embed optimization within the EGM iteration, but it avoids optimization at qualifying stages and parallelizes across grid cells because the closed-form inversions have no cross-cell dependencies.

On the two-asset model of [Auclert et al. \(2021\)](#), with both solvers in my own matched implementation, EGM^n is faster than the Nested Endogenous Grid Method (NEGM) at every grid and thread count tested, by roughly an order of magnitude in serial and by about 3.7 to 7.9 times even at sixteen matched threads. Threads narrow the gap because the nested search has per-node work to spread across them, while the closed-form inversion has almost none. For the curvilinear grids the inversions produce, I develop ENGINE (ENdogenous Grid INTERpolation and Extrapolation), with which the full solve runs roughly three to four and a half times faster than with the curvilinear interpolation previously proposed by [White \(2015\)](#) for such grids, at matched accuracy on the two-asset application.

As a direct product of the backward step, the decomposition also returns intermediate-stage marginal values, quantities single-block solvers never construct. In the two-asset model, their ratio minus one is the adjustment-cost wedge that prices converting illiquid wealth into liquid.

The Endogenous Grid Method was built to avoid the cost of searching over choices: for a three-choice problem with G grid points per dimension, that search costs $O(G^3)$ evaluations at every state, a price paid across

the whole state space.¹ Carroll (2006) introduces it for the consumption-saving problem: starting from a grid of post-decision states, EGM inverts the first-order condition to recover the optimal policy and the pre-decision state that implies it, so a numerical search becomes a function evaluation. Barillas and Fernández-Villaverde (2007) extends it to a second control, the labor-leisure choice, in a neoclassical growth model.²

In the decade after EGM's introduction, its scope broadened along two fronts. On the structural side, Fella (2014) relaxes continuity requirements for non-convex problems and Hintermaier and Koeniger (2010) handles occasionally binding constraints among endogenous variables; Druedahl and Jrgensen (2017) combines both advances into G2EGM, which admits multiple controls and states. On the discrete-choice side, Iskhakov et al. (2017) incorporates jumps between alternatives through extreme value errors. Their method, however, remains restricted to a single control and state variable.³ Separately, Clausen and Strub (2020) develops Reverse Calculus for nested and non-convex optimization; G2EGM (Druedahl and Jrgensen 2017) builds its first-order-condition arguments on an earlier working-paper version of these differentiability results.

EGMⁿ builds most directly on the multivariate extensions. White (2015) formalizes the conditions under which EGM applies to multidimensional problems and builds interpolation methods for the curvilinear grids that result; Ludwig and Schön (2018) proposes Delaunay triangulation for the same grids, though triangulation costs can offset the time saved by avoiding grid search. Iskhakov (2015) characterizes a triangular structure in the system of first-order conditions that permits sequential solution by substitution, with no root-finding.⁴ Triangular EGM asks for triangularity of the entire system; EGMⁿ asks only for sequential invertibility, stage by stage, which admits the mixed structures described above. Druedahl (2021) develops NEGM, which pairs an upper envelope with a one-dimensional numerical solve nested in the outer EGM iteration; nesting cuts complexity relative to joint optimization but keeps the embedded solve, which EGMⁿ removes.⁵

¹See Maliar and Maliar (2014) for a thorough survey of numerical methods for such problems.

²See also Maliar and Maliar (2013).

³On the preference side, Hallengreen et al. (2024) develop iEGM for utility functions lacking a closed-form inverse marginal utility, and Jrgensen (2013) provides Monte Carlo evidence that EGM dominates value function iteration and MPEC in speed and robustness for structural estimation.

⁴A parallel line exploits first-order conditions without EGM's grid inversion: the envelope condition method of Maliar and Maliar (2013) and the fixed-point iteration of Mendoza and Villalvazo (2020) for models with occasionally binding constraints.

⁵For state spaces beyond EGM's reach, global-solution methods approximate policies or equilibrium conditions with Gaussian processes or neural networks, trading interpretable policy functions for scale (Scheidegger and Bilonis 2019; Maliar et al. 2021; Azinovic et al. 2022); separately, Bayer et al. (2026) carry the endogenous-gridpoint idea to the law of motion of the wealth distribution.

Section 2 develops the decomposition on a two-choice consumption-labor problem, the smallest example that shows the chaining. Section 3 takes it to the canonical two-asset liquid-illiquid household block of [Auclert et al. \(2021\)](#), the household side of a typical HANK model, in which two asset states generate two-dimensional curvilinear grids and the benchmarks above are measured. ENGINE exists to interpolate those grids, and Section 4 develops and tests it. Section 5 gives practical guidelines for splitting a problem into stages and two sufficient structures, separable utility and invertible transition marginals, under which a stage inverts, and Section 6 marks where the method stops applying.

2 The Sequential Endogenous Grid Method

A household that simultaneously chooses consumption and labor supply presents the joint solver with a two-dimensional search at every state space point. But when decisions are separable in stages, though not necessarily globally, sequential decomposition lets EGM inversion solve each stage efficiently.

2.1 Problem Setup

I demonstrate EGMⁿ on a consumption-labor model in which the household chooses leisure and consumption under a risky wage. It is the smallest example that shows the chaining: two simultaneous decisions, a sequential ordering in which each stage sheds a state variable, and an EGM inversion at both stages, only one of which is the consumption margin EGM was built for. The household maximizes the present discounted lifetime utility of consumption and leisure over a finite horizon of $T + 1$ periods, where $t = 0$ denotes the beginning of the planning horizon (the first period of economic life) and $t = T$ the terminal period:

$$V_0(a_0, \theta_0) = \max \mathbb{E}_0 \left[\sum_{t=0}^T \beta^t u(c_t, z_t) \right], \quad (1)$$

where V_0 is the lifetime value function at the start of period zero, a_0 is the stock of liquid assets the household starts with, θ_0 is the transitory wage shock, and $\mathbb{E}_0$ denotes the expectation conditional on information available at the start of the planning horizon.

Period utility is additively separable in consumption and leisure:

$$u(c, z) = u(c) + h(z) = \frac{c^{1-\rho}}{1-\rho} + v \cdot \frac{z^{1-\zeta}}{1-\zeta}, \quad (2)$$

where $\rho > 0$ is the coefficient of relative risk aversion, $\zeta > 0$ determines the curvature of leisure utility (and thus the Frisch elasticity of labor supply) and $v > 0$ scales leisure utility. I chose additive separability because it

gives the leisure margin a marginal utility of its own, and therefore something to invert. The household solves

$$v_t(a_t, \theta_t) = \max_{c_t \geq 0, z_t \in [0,1]} u(c_t, z_t) + \beta \mathbb{E}_t[v_{t+1}(a_{t+1}, \theta_{t+1})]$$

subject to

$$n_t = 1 - z_t, \quad m_t = Ra_t + \theta_t n_t, \quad a_{t+1} = m_t - c_t,$$

where β is the discount factor and R is the risk-free gross return. The terminal condition is $v_{T+1} \equiv 0$, so the household in the final period T consumes all remaining resources; leisure then solves the static first-order condition (or its corners). Throughout, I assume standard constraint qualifications hold, so that interior solutions satisfy first-order conditions.⁶ The intermediate quantities the constraints define are market resources m_t (incoming assets with their return, plus labor income) and end-of-period liquid assets a_{t+1} .

Although the two decisions are economically simultaneous, the dependence structure permits sequential solution. The labor-leisure choice determines market resources, and given those resources the consumption-saving choice determines liquid assets.

The sequential decomposition starts the period with the labor-leisure problem. To distinguish value functions at different stages, I introduce stage superscripts: the original problem has $v_t \equiv v_t^0$, the value at the first decision stage (labor-leisure), and v_t^i is the value after the decisions at stages $0, 1, \dots, i-1$. At this stage, the household observes the liquid assets a_t it carries into the period and the wage offer θ_t , choosing leisure to maximize the sum of current leisure utility and the continuation value from market resources m_t :

$$v_t^0(a_t, \theta_t) = \max_{z_t \in [0,1]} h(z_t) + v_t^1(m_t), \quad n_t = 1 - z_t, \quad m_t = Ra_t + \theta_t n_t. \quad (4)$$

Once market resources are realized, the pure consumption-saving problem determines how to allocate m_t between current consumption and liquid assets. The state space shrinks to a single dimension because the wage offer no longer matters:

$$v_t^1(m_t) = \max_{c_t \in [0, m_t]} u(c_t) + v_t^2(a_{t+1}), \quad a_{t+1} = m_t - c_t. \quad (5)$$

⁶Specifically: (i) utility and value functions are twice continuously differentiable in the interior of the constraint set; (ii) the Inada conditions $\lim_{c \rightarrow 0} u'(c) = \infty$ and $\lim_{c \rightarrow \infty} u'(c) = 0$ hold, so consumption stays away from zero; and (iii) constraint sets are convex with non-empty interior. These conditions ensure first-order conditions are necessary for optimality at interior solutions. Strict concavity of the utility and value functions further ensures sufficiency: any point satisfying the first-order conditions is a global maximum.

No choice remains after consumption, so the period closes with a post-decision value: the discounted expectation of next period's problem, taken over the wage draw the household does not yet know,

$$v_t^2(a_{t+1}) = \beta \mathbb{E}_t[v_{t+1}^0(a_{t+1}, \theta_{t+1})]. \quad (6)$$

The sequential formulation follows the nested approaches of [Clausen and Strub \(2020\)](#) and [Druehl \(2021\)](#) but composes EGM inversions without embedding optimization. Each choice is self-contained in a subproblem, with the structure chosen to minimize state variables at each stage. The sequential formulation is mathematically equivalent to the original joint problem by Bellman's principle of optimality ([Bellman 1957](#)): the expectation operator appears only in the final subproblem, so no uncertainty resolves between stages and every decision uses the same information set. Were shocks to resolve between stages, households would gain information unavailable in the original formulation, changing the problem's solution.

Limiting cases confirm that the decomposition nests simpler models. When $v \rightarrow 0$, leisure utility vanishes and the labor-leisure stage degenerates: the household works full time regardless of wages, recovering the standard consumption-saving EGM of [Carroll \(2006\)](#). When $\beta \rightarrow 0$, the fully impatient household consumes all resources immediately, recovering the static problem.

2.2 Sequential Solution

EGMⁿ inverts both first-order conditions of the period in closed form, without grid search or root-finding, applying the standard EGM inversion to the consumption Euler equation and an analogous one to the leisure condition.

The consumption-saving stage is the standard EGM of [Carroll \(2006\)](#). I take it as given and pause only to fix notation: bracketed variables ($[a]$, $[m]$) denote exogenous grids, gothic letters ($\mathfrak{c}$, $\mathfrak{m}$) denote endogenous quantities built by inverting a first-order condition, and primes are ordinary derivatives. Inverting the Euler equation $u'(\mathfrak{c}_t) = (v_t^2)'(a_{t+1})$ for consumption, which is possible because the marginal-utility map u' is strictly decreasing and hence one-to-one, gives consumption on the exogenous post-decision grid, and the budget identity returns the endogenous grid of market resources:

$$\mathfrak{c}_t([a]) = (u')^{-1}((v_t^2)'([a])), \quad \mathfrak{m}_t([a]) = \mathfrak{c}_t([a]) + [a]. \quad (7)$$

The pair $(\mathfrak{m}_t, \mathfrak{c}_t)$ supports an interpolating consumption function for market-resources values off the grid. Through the envelope condition

$(v_t^1)'(m_t) = u'(c_t)$, the same pair gives the labor-leisure stage the marginal value it needs without additional computation.

The leisure choice has a first-order condition of its own. Substituting $n_t = 1 - z_t$ and $m_t = Ra_t + \theta_t n_t$ puts market resources inside v_t^1 , so the whole trade-off between leisure and labor income runs through that one function:

$$h'(z_t) = (v_t^1)'(m_t)\theta_t. \quad (8)$$

This equates the marginal utility of leisure to the opportunity cost of forgone labor income θ_t , scaled by the marginal utility of consumption $(v_t^1)'(m_t) = u'(c_t)$.

The marginal utility of leisure inverts in closed form,⁷ so on an exogenous grid of market resources $[m]$ and wage shocks $[\theta]$ leisure follows directly:

$$\hat{z}_t([m], [\theta]) = (h')^{-1}((v_t^1)'([m])[\theta]). \quad (9)$$

Nothing in that step is special to consumption. The inverse marginal utility of leisure $(h')^{-1}$ carries an exogenous grid of market resources and wages to an endogenous grid of incoming assets, exactly as $(u')^{-1}$ carries post-decision assets to market resources one stage later. EGMⁿ chains the two.

Households with high market resources m_t and low wage offers θ_t , however, may find the unconstrained optimum violates the feasibility constraint $z_t \in [0, 1]$. When this occurs, projecting the solution onto the constraint boundary defines the constrained optimal function $\hat{z}_t([m], [\theta])$ as

$$\hat{z}_t([m], [\theta]) = \max\{\min\{\hat{z}_t([m], [\theta]), 1\}, 0\}. \quad (10)$$

The projection keeps leisure within the time endowment; in practice only the upper bound binds.⁸ Interpolation handles the resulting kink by sorting the endogenous grid and fitting piecewise segments on either side of it.

Labor follows as $1 - \hat{z}_t(m_t, \theta_t)$, and for each θ_t and m_t on the exogenous grid the endogenous grid of incoming assets is $a_t(m_t, \theta_t) = (m_t - \theta_t (1 - \hat{z}_t(m_t, \theta_t))) / R$.

By the envelope theorem, the marginal value the previous period needs is $\partial v_t^0 / \partial a = R(v_t^1)'(m_t)$, which completes the backward step. At inte-

⁷The marginal utility of leisure is $h'(z) = v z^{-\zeta}$, so its inverse at any positive marginal-utility value μ is $(h')^{-1}(\mu) = (\mu/v)^{-1/\zeta}$.

⁸With the functional form $h'(z) = v z^{-\zeta}$, the Inada condition $\lim_{z \rightarrow 0} h'(z) = \infty$ ensures the lower bound $z_t = 0$ never binds for finite wages and marginal values. At the upper bound $z_t = 1$, the household chooses full leisure when $h'(1) = v \geq (v_t^1)'(m_t)\theta_t$, which occurs for sufficiently low wages or high wealth. A zero wage draw carries no labor income at the margin, so there is no interior condition to invert and the projection assigns the full-leisure corner directly.

rior solutions this marginal value equals $Rh'(z_t)/\theta_t$. At a corner, however, only the first equality survives, because a Kuhn-Tucker inequality replaces the first-order condition.

The resulting endogenous grid for the labor-leisure problem is curvilinear rather than rectilinear.⁹

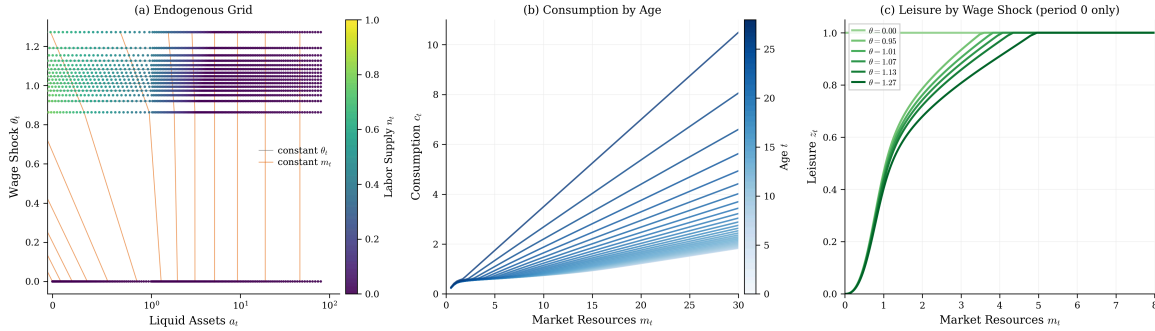


Figure 1 Sequential decomposition produces well-behaved policy functions for the consumption-labor model ($T = 30$, parameters in footnote). Panel (a) shows the curvilinear endogenous grid from the labor-leisure stage at $t = 0$ on a log-scaled asset axis, with points colored by optimal labor supply and the two mesh-line families identified in the legend. Panel (b) shows the consumption functions; it omits the terminal period and the final decision period, where the approaching horizon dominates the policy. Panel (c) shows that lower wages induce more leisure at $t = 0$, since the opportunity cost of not working falls, with the unemployed household ($\theta = 0$) at full leisure throughout.

Figure 1 collects the full solution; no numerical optimization produced any of the policies it shows.¹⁰ In the first period, the marginal propensity to consume falls from about 0.48 at the borrowing constraint to about 0.024 near $m_t = 4.4$, then rises slowly to about 0.06 at higher wealth. The slow rise plausibly mirrors the decline in the household's expected future labor, which reaches zero near $m_t = 18$, shortly before the propensity levels off near $m_t = 20$. Below $m_t = 18$, saving most of an extra unit lets the household work less later in life. The labor-leisure inversion already produces a curvilinear grid, the simplest case of the meshes Section 4 interpolates. The two-asset problem of Section 3, where two state variables persist across stages, provides the more demanding test.

3 A Two-Asset Liquid-Illiquid Household

The consumption-labor problem in Section 2 carries at most one post-decision state variable per stage. The canonical stress test for a sequential EGM step is the two-asset household block at the center of the heterogeneous-agent New Keynesian (HANK) literature (Kaplan et al.

⁹Simpler interpolation methods would suffice for the labor-leisure problem, since the grid warps along only one dimension. The replication code interpolates that stage by nested one-dimensional linear interpolation, which is enough here; the two-dimensional curvilinear meshes of Section 3 are where ENGINE is required.

¹⁰Parameters are $\rho = 5$ and $\beta = 0.9$, with HARK's defaults $R = 1.03$ and a survival probability of 0.98 that multiplies the discount factor. Wage shocks are lognormal with standard deviation 0.1 and a five percent probability of a zero wage. Labor-specific parameters are $\zeta = 2$ and $v = 16$.

2018; Auclert et al. 2021): a household holding liquid wealth alongside an illiquid account, moving funds between them at a convex adjustment cost. Two continuous state variables persist across the EGM stages, so both coordinates of the endogenous grid warp. The same structure, a second persistent stock alongside liquid wealth, appears in durable goods, human capital, and health investment problems; Online Appendix A develops the health investment model of Ludwig and Schön (2018) and White (2015) as a second worked example. I solve the exact household block of Auclert et al. (2021) at its published calibration, so that the benchmark runs on a model the macro audience already uses at scale. I treat only the household block and leave the New Keynesian aggregate machinery that closes the HANK economy to the literature cited above. Solver speed compounds in that setting: a general-equilibrium iteration can solve the household block thousands of times, each pass consuming the policies on the regular state grid, where the non-stochastic histogram method of Young (2010) carries the wealth distribution forward.

Nested methods conventionally handle the illiquid choice with a numerical first-order-condition search embedded in the backward step, exactly the numerical solve EGMⁿ is designed to remove; the decomposition below replaces it with a second closed-form inversion. Timing the two solvers measures what that substitution saves, net of the regrid and the regular-grid Euler inversion it adds.

3.1 The Household Problem

A household enters the period with state (b, a, e) , where b is incoming liquid wealth, a is the incoming illiquid balance, and e is idiosyncratic labor productivity following a discrete Markov chain with transition matrix $\Pi(e' | e)$. The letters follow the two-asset convention of the heterogeneous-agent literature. Following the Auclert et al. (2021) timing convention, a and b are post-decision balances: the choices this period, written a' and b' , become next period's state. Both balances pay out their gross returns, $R_a = 1 + r_a$ on illiquid and $R_b = 1 + r_b$ on liquid, labor income $y(e)$ is received, and market resources are $m = R_b b + R_a a + y(e)$. The household chooses consumption $c \geq 0$ together with new balances $b' \geq 0$ and $a' \geq 0$ subject to the single within-period budget

$$c + b' + a' + \Psi(a', a) = m. \quad (11)$$

The illiquid asset earns the higher return ($r_a > r_b$), but moving funds into or out of it costs Ψ . Adjustment costs take the quadratic form of Auclert et al. (2021),

$$\Psi(a', a) = \frac{\chi_1}{2} \left(\frac{a' - R_a a}{R_a a + \chi_0} \right)^2 (R_a a + \chi_0), \quad (12)$$

with offset $\chi_0 > 0$ and scale $\chi_1 > 0$. The cost vanishes when the household reinvests the illiquid interest ($a' = R_a a$), is strictly convex in a' , and scales with the post-return illiquid stock $R_a a + \chi_0$, so that even small illiquid positions are costly to deplete. The recursive problem is

$$v(b, a, e) = \max_{c, b', a'} u(c) + \beta \mathbb{E}[v(b', a', e') | e], \quad u(c) = \quad (13)$$

subject to the budget and the non-negativity constraints. Unlike the value-based curvilinear EGM of [White \(2015\)](#), the inversions below need only positive risk aversion, $\rho > 0$, because each inverts a strictly positive marginal value rather than a utility level.

In one familiar limit, $\chi_1 \rightarrow \infty$, the illiquid balance moves only at the gross return ($a' \rightarrow R_a a$) and the problem reduces to a one-asset consumption-saving model in (b, e) . A fixed cost paid on any adjustment ($a' \neq R_a a$) would generate a region of inaction and a non-convexity; that case leaves the smooth-EGM class and belongs to the G2EGM upper-envelope methods, the natural extension noted in the conclusion.

3.2 Stage Decomposition and Backward Solution

The joint problem splits into three stages: an illiquid choice, a consumption choice, and a post-decision expectation. Under the stage convention of Section 2, $v \equiv v^0$ is the first decision stage, v^1 is the value entering consumption, and w is the post-decision value, which carries its own letter here, following [Powell \(2011\)](#), because both inversions read off it. With two persistent states, derivatives are partial: they appear as subscripts on stage values ($v_\ell^1 \equiv \partial v^1 / \partial \ell$) and, because w carries no stage label, as superscripts on w ($w^{b'} \equiv \partial w / \partial b'$). Productivity e passes through every stage unchanged and resolves only in the expectation; every stage value therefore carries e as a state. The expectation again appears only in the post-decision stage, so the equivalence argument of Section 2 applies unchanged. Where the consumption-labor stages shed a state variable apiece, here the state count stays at three throughout. Backward induction visits the stages in reverse, and both decision stages turn out to be EGM inversions, linked by envelope conditions.

The post-decision stage integrates over next period's productivity:

$$w(b', a', e) = \beta \sum_{e'} \Pi(e' | e) v(b', a', e'). \quad (14)$$

3.2.1 Stage 1: The Consumption EGM with the Constrained Segment

Given the liquid resources ℓ left after the illiquid choice, the consumption stage chooses c :

$$v^1(\ell, a', e) = \max_{c \geq 0} u(c) + w(b', a', e), \quad b' = \ell - c \geq 0. \quad (15)$$

The illiquid choice a' is already made when consumption is chosen, so at each fixed a' this stage is the consumption EGM of Section 2, run along the b' axis and resting on the same separable utility (Section 5). The stage reads the post-decision marginals $w^{b'}, w^{a'}$ on the exogenous (b', a') grid, inverts the Euler equation,

$$c(b', a', e) = (w^{b'}(b', a', e))^{-1/\rho}, \quad \ell = b' + c, \quad (16)$$

and returns $v_\ell^1, v_{a'}^1$ on a regular grid of liquid resources ℓ . The liquid borrowing constraint $b' \geq 0$ is handled in the manner of [Carroll \(2006\)](#), by prepending the single point $(\ell = 0, c = 0, b' = 0)$ to the endogenous mesh. With b' starting at zero, the first bracket lies on the line $c = \ell$, and linear interpolation reproduces the constrained segment exactly. Through the envelope the illiquid marginal passes unchanged, $v_{a'}^1 = w^{a'}$ evaluated at the Stage 1 optimum.

3.2.2 Stage 0: The Illiquid EGM with Closed-Form Inversion

The illiquid stage, given (b, a, e) , chooses a' and converts market resources into liquid resources:

$$v^0(b, a, e) = \max_{a' \geq 0} v^1(\ell, a', e), \quad \ell = m - a' - \Psi(a', a). \quad (17)$$

EGMⁿ departs from NEGM at this stage, where the first-order condition for a' balances the marginal value of liquid against the marginal value of illiquid net of the adjustment-cost wedge:

$$1 + \Psi_1(a', a) = \frac{v_{a'}^1(\ell, a', e)}{v_\ell^1(\ell, a', e)}, \quad \Psi_1 \equiv \frac{\partial \Psi}{\partial a'}. \quad (18)$$

The right-hand side is read directly from Stage 1's regular-grid envelopes. A nested method searches over a' , but EGMⁿ holds the gridded choice a' fixed and recovers the incoming stock a . Here the enabling structure is the invertible transition marginal of Section 5: the stage has no utility term of its own, and Ψ_1 is strictly decreasing in the unknown a because $\Psi_{12} < 0$. For the quadratic cost, $\Psi_1(a', a) = \chi_1 u$, where $u \equiv (a' - R_a a) / (R_a a + \chi_0)$ is the standardized adjustment wedge, and

$$\Psi_{11} = \frac{\chi_1}{R_a a + \chi_0} > 0, \quad \Psi_{12} = -\frac{\chi_1 R_a (a' + \chi_0)}{(R_a a + \chi_0)^2} < 0, \quad (19)$$

where Ψ_{ij} denotes the corresponding second partial derivative.

Define the wedge target $\tau \equiv v_{a'}^1 / v_\ell^1 - 1$; the inversion is

$$u^* = \frac{\tau}{\chi_1}, \quad a(\ell, a', e) = \frac{a' - u^* \chi_0}{R_a (1 + u^*)}, \quad (20)$$

a single multiplication and a ratio per grid point. The remaining endogenous quantities follow from the two accounting identities: endogenous market resources from the liquid-resources definition, $m = \ell + a' + \Psi(a', a)$, and the endogenous liquid balance from the market-resources definition, $b = (m - R_a a - y(e))/R_b$. There is no root-find and no iteration; the entire illiquid choice is closed form.¹¹

The inversion produces a curvilinear (b, a) mesh while retaining the exogenous grid's adjacency. Section 4 states the two geometric conditions ENGINE places on such a mesh, both of which I verify directly on every reported benchmark mesh. After ENGINE carries the illiquid choice to the regular state grid, the remaining policies follow by composition: the implied liquid resources pull the Stage 1 consumption function back to (b, a, e) , evaluated through the same Euler inversion, and the liquid choice is the budget residual (Online Appendix B). Figure 2 shows the result.¹² The inversion bends the mesh most where the optimal adjustment is largest. Yet mesh lines of constant ℓ and constant a' never cross.

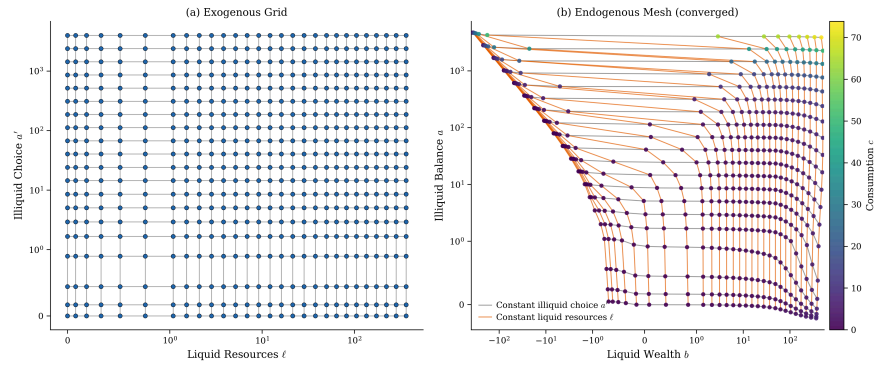


Figure 2 Stage 0 grid transformation at the converged fixed point. (a) Exogenous grid of liquid resources ℓ and illiquid choice a' . (b) Endogenous mesh of incoming states (b, a) , colored by consumption. Withdrawal-financed nodes with negative b lie outside the state space and are never queried.

3.2.3 Boundaries: Two Economic Edges and One Grid-Design Rule

The inversions handle every interior solution; the boundaries come in two kinds. The *economic* boundaries are the non-negativity constraints, neither of which costs a subproblem. The liquid edge ($b' = 0$) is the consumption EGM's own borrowing constraint, handled inside Stage 1 by the prepended constraint point. The illiquid edge ($a' = 0$) reduces the cell to a Stage 1 evaluation at the implied liquid resources $\ell = m - \Psi(0, a)$, so

¹¹The Stage 0 envelope returns $v_b^0 = R_b c^{-\rho}$ and $v_a^0 = (R_a - \Psi_2) c^{-\rho}$, where $\Psi_2 = \partial\Psi/\partial a = -R_a \chi_1 u \left(1 + \frac{1}{2}u\right)$ collects the direct illiquid return and the indirect effect through the adjustment cost.

¹²The figure uses the benchmark calibration on a 20×20 state grid solved to tolerance 10^{-9} , shown at the median of the three productivity states; the Stage 1 liquid-resources axis extends past the liquid state grid by the domain rule discussed below. Both axes are log-scaled away from zero because the underlying grids are log-spaced.

the corner policy is read off Stage 1's own $a' = 0$ slice, and each cell picks between the interior inversion and the corner by the sign of the unconstrained policy. With strictly concave utility and a convex adjustment cost, the stage policies are single-valued and continuous, so no discrete jumps arise and the solver never requires an upper envelope.

The *grid* boundary imposes a design rule with no analogue in one-block solvers. Sequential decomposition materializes the intermediate state ℓ on a regular grid between the stages, and that grid must cover whatever the second inversion can request. Inherited liquid wealth does not bound it: a household rich in illiquid assets finances liquidity by withdrawing, and because the marginal value of illiquid wealth is non-negative the first-order condition bounds the withdrawal wedge by $u^* \geq -1/\chi_1$, so ℓ reaches at most

$$\bar{\ell} = R_b \bar{b} + \bar{y} + \frac{R_a \bar{a} + \chi_0}{2\chi_1}, \quad (21)$$

in closed form, where $\bar{b}$ and $\bar{a}$ are the largest nodes of the liquid and illiquid grids and $\bar{y}$ the highest income realization.

Truncating the ℓ -domain to the liquid state grid, the natural shortcut, strips mesh coverage at high illiquid wealth without warning and forces the realign step to extrapolate there. On this calibration the iteration then converges to a budget-infeasible pseudo-solution in the affected band, which a constraint-valid Euler diagnostic exposes at once (Section 4).¹³

3.3 Comparison to Nested EGM

I compare the two solvers on total time to a converged household solution, at matched grids, matched tolerance (10^{-9}), and matched linear interpolation. All timings are net of one-time compilation cost, best of three repetitions from a warm start.¹⁴ Table 1 reports both solvers' times. The solvers share the consumption-stage EGM, the constrained segment, the post-decision integration, and the fixed-point iteration. NEGM solves the illiquid first-order condition directly on the regular state grid by a safeguarded bracketing iteration whose relative bracket tolerance must sit two orders below the outer norm, at 10^{-11} . With the bracket loosened to 10^{-9} , per-node root-find noise keeps the fixed-point iteration from meeting its own norm and the solve hits the iteration cap, so the comparator

¹³The solver builds the ℓ -grid to this bound (a wider bound may be supplied when a calibration calls for one); with the rule in place, every realign query near the fixed point is interior. At the liquid edge, Euler-implied consumption is capped at the constraint-feasible maximum, and the liquid policy is its budget residual, so the delivered policies satisfy the within-period budget identity by construction.

¹⁴The timing comparison measures the net effect of replacing NEGM's regular-grid root-find with the EGMⁿ backward step, including the latter's extra Stage-0 regrid and regular-grid inversion. In the full-solve experiment of Section 4 I hold the EGMⁿ solver fixed and swap only that regrid to isolate ENGINE from the decomposition.

converges only with the tighter bracket (the archive's inner-tolerance table records the check). EGM^n instead applies the closed-form inversion on its exogenous grid, then pays for a general ENGINE regrid and a regular-grid Euler inversion. The two backward steps are parallelized identically, so the comparison is matched at one thread and at sixteen.¹⁵

Table 1 Time in seconds to a converged two-asset solution, NEGM versus EGM^n , at matched grids, tolerance, interpolation, and thread counts. Parentheses give the speed ratio over NEGM at the same thread count.

Grid	1 thread		16 threads	
	NEGM	EGM^n	NEGM	EGM^n
(40, 60)	9.81	0.78 (12.5x)	0.89	0.24 (3.7x)
(80, 120)	42.22	3.22 (13.1x)	3.27	0.54 (6.1x)
(160, 240)	178.59	13.66 (13.1x)	13.56	1.73 (7.9x)

NEGM and EGM^n converge in a comparable number of backward iterations (at (40, 60), (80, 120), and (160, 240), NEGM takes 473, 481, and 487 iterations and EGM^n takes 395, 410, and 424). The timing gap comes from the illiquid step, where EGM^n makes one closed-form evaluation and NEGM an iterated sequence of interpolated envelope evaluations. EGM^n still wins after paying for the general Stage-0 ENGINE regrid and regular-grid Euler inversion that NEGM does not need, since NEGM's root-find returns the policy on the regular (b, a) grid. Serially the factor is about thirteen at all three grids. At the largest, one serial NEGM solve takes just under three minutes against fourteen seconds for EGM^n . At matched sixteen threads the factor grows from about 3.7 to 7.9 with the grid size. The root-find carries enough arithmetic per node to scale almost linearly with threads, whereas the EGM^n step is so cheap per node that at small grids its solve time is probably dominated by the fixed cost of coordinating the threads themselves. I expect the matched-thread ratio to keep climbing toward the serial factor at the grid sizes where computing time matters.

The two solvers also agree on the solved margins. Median relative distance between the converged policies stays below half a percent for both the liquid and the illiquid balance at every timing grid. At the 192×160 policy-benchmark grid the two asset margins agree to 0.16 and 0.29 percent. On the model's own liquid Euler equation, the comparison depends on where I take the residual. Off the grid, where both converged policies are interpolated and the residual puts the two methods on equal footing, EGM^n 's median is the smaller at six of the seven reported grids

¹⁵The timing script and its output table are in the replication archive, whose README records the full benchmark environment and exact package versions (AMD Ryzen 9 5900X; Python 3.13, NumPy 2.3, Numba 0.62, SciPy 1.16); reported runs cap the thread count at sixteen. The archive's test suite validates that the solver's output is identical across thread counts (the backward step writes each grid cell independently, with no cross-cell reductions).

(three timing, three diagnostic, and the policy benchmark), trailing only at the (80, 120) timing grid, by a quarter of a percent. At the three diagnostic grids the medians differ by less than fifteen percent, and EGMⁿ's advantage grows to factors of 1.4 and 1.6 at the two largest grids overall, the (160, 240) timing grid and the policy benchmark. On the regular grid, EGMⁿ's residual runs one to two orders of magnitude below NEGM's, about 1.5×10^{-5} against 3.9×10^{-4} at (40, 60), because EGMⁿ delivers regular-grid consumption as an Euler-inversion output where NEGM reads it off an interpolated policy (Online Appendix B).¹⁶

Each stage is itself a Bellman problem; every iteration of the backward step therefore also produces the marginal values v_l^1 and v_a^1 , of the consumption stage on a regular grid. Their ratio minus one is the wedge $\tau \equiv v_a^1/v_l^1 - 1$, the adjustment-cost shadow price of converting illiquid into liquid wealth. A modeler can read it off to answer questions that turn on this object (how the adjustment friction varies over the wealth distribution, where it binds), with no auxiliary computation. NEGM shares this property (its root-find runs on the same stage envelopes), but solvers that treat the period as a single block with two controls, the G2EGM family among them, deliver policies without ever constructing the intermediate-stage values. Recovering the same objects on the whole stage grid, beyond the optimal choices, requires re-deriving the stage decomposition that EGMⁿ starts from.

4 Multivariate Interpolation on Curvilinear Grids

EGM's efficiency comes from working on exogenous grids on which calculations are straightforward, then using the inverted Euler equation to recover endogenous state variables. That recovery, however, gives up the grid's rectilinear geometry, so standard multilinear interpolation, which assumes each dimension varies independently, cannot handle the curvilinear grid that results.

ENGINE relies on the index structure that endogenous grids inherit from exogenous grids despite geometric warping. It reduces multidimensional interpolation to a sequence of one-dimensional interpolations along these inherited indices, because row k remains row k after warping and neighbors remain neighbors, even when geometric regularity (uniform spacing, rectilinear alignment) is lost.

Consider a curvilinear grid indexed by $j \in \{1, \dots, J\}$ and $k \in \{1, \dots, K\}$, where each "row" (fixed k) contains J points. The curvilinear interpolation of White (2015) (the curvilinear method below) locates the containing

¹⁶The script behind these readings and its output table are in the replication archive alongside the timing benchmark; both solvers run at the same tolerance, and the distances are medians over the regular (b, a) grid and every income state.

quadrilateral cell via sector-walking, then solves for normalized coordinates within that cell (a quadratic equation in 2D, Newton iteration in higher dimensions). ENGINE sidesteps both operations: for each of the K rows it asks where a vertical line at x^* intersects that row, a one-dimensional interpolation problem that finds where x^* falls among the J x -coordinates and interpolates the y -coordinate to produce an intermediate point. A final 1D interpolation across these K intermediate points yields the answer. In implementation, a single query touches only the $O(\log K)$ rows its binary search probes, and the dense regrid reuses brackets across sorted targets.

4.1 The ENGINE Algorithm

ENGINE works in index space, where bracketing is efficient, and can map back to physical coordinates because each endogenous point (x_{jk}, y_{jk}) keeps the indices (j, k) of the exogenous post-decision grid point that generated it through the EGM inversion. This index-to-mesh correspondence is a homeomorphism under conditions developed below, a result the appendix states and proves (Theorem 1). Figure 3 makes the correspondence concrete. Because the mapping preserves neighborhoods, a bracket search on the rectilinear index grid is a valid search of the warped physical mesh.

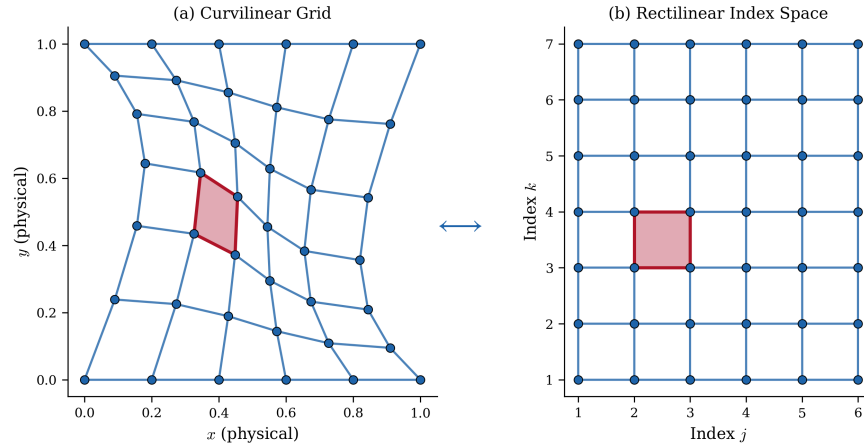


Figure 3 The grid homeomorphism between index and physical space. (a) Curvilinear physical grid. (b) Rectilinear index grid. The shaded cell has the same four nodes in both.

The full two-pass listing and boundary-extrapolation behavior are recorded in the Appendix. Figure 4 traces one query through both passes.

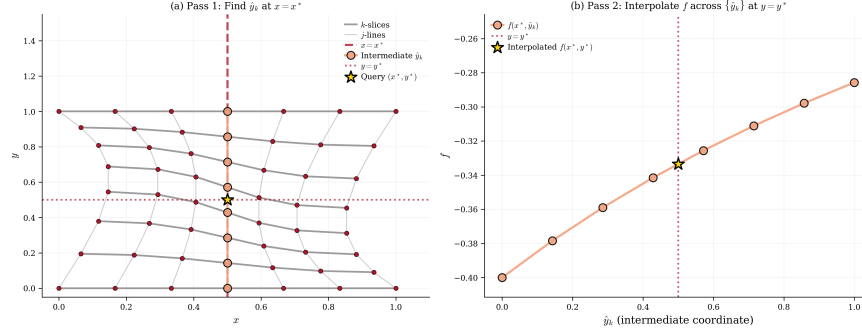


Figure 4 The two passes of ENGINE. (a) Pass 1: a 1D bracket search on each k -slice at $x = x^*$ gives the intermediate ordinates $\hat{y}_k$ (circles). (b) Pass 2: one 1D interpolation across the $\hat{y}_k$ gives the value at the query point (star).

For a query point (x^*, y^*) , the interpolated value is

$$\hat{f}(x^*, y^*) = \text{ENGINE}_k \left(y^*; \{ \hat{y}_k \}_{k=1}^K, \{ \hat{f}_k \}_{k=1}^K \right) \quad (22)$$

where ENGINE_k denotes Pass 2's one-dimensional linear interpolation across the row index k , and $\hat{y}_k$ and $\hat{f}_k$ are the intermediate coordinates and values from Pass 1.

On a warped mesh the two passes do not commute. Walking the regular axis first moves the brackets less between rows.

For sorted targets, each search resumes from the previous target's bracket and never moves backward, so total work is proportional to the number of targets plus their total bracket displacement.

4.2 When ENGINE Applies

The pure consumption-saving problem illustrates the basic phenomenon. An exogenous grid of post-decision liquid assets [a] maps through the inverted Euler equation to an endogenous grid of market resources [m] with different spacing, the nonlinearity of marginal utility making the mapping non-uniform and binding constraints adding kinks. In one dimension, non-uniform linear interpolation handles the distortion. Higher-dimensional problems, however, inherit the warping in each dimension simultaneously. How much of the grid's structure survives then determines the appropriate interpolation method.

Previous approaches to this problem include Delaunay triangulation (Ludwig and Schön 2018) and curvilinear interpolation (White 2015). Delaunay triangulation constructs a mesh of simplices over the scattered points and performs barycentric interpolation within each simplex. Although general and robust, triangulation is computationally expensive. Construction from unstructured point sets requires at least order $N \log N$ operations in the algebraic decision-tree model, a lower bound rather than an implementation artifact: sorting N numbers reduces to triangulating N points on a parabola (Preparata and Shamos 1985). The trian-

gulation must also be recomputed when grids change during iterative solution procedures. The curvilinear method exploits the grid structure to map irregular quadrilateral sectors to the unit square for bilinear interpolation; sector location uses a visibility walk algorithm, and coordinate calculation in 2D uses a closed-form quadratic solution (3D and higher require Newton iteration).

ENGINE requires structural conditions on the endogenous grid. Analytic invertibility of a stage first-order condition (Proposition 1, Proposition 2) is not enough on its own, because the recovery map defined in the Introduction can still fold when continuation is nonconcave or when cross effects reverse orientation. Discrete choices and nonconcave regions are the leading exclusions.

ENGINE imposes two requirements, both geometric: cells cannot fold, and each coordinate must increase with its own index. A third property, a bound on bracket displacement between adjacent rows, governs how far ENGINE must search. The implementation verifies every windowed bracket and falls back to a full row search on a miss, so the search margin affects only speed.

Nonlinear EGM mappings can distort cells severely enough to invert them. The fold-free condition, the classical validity condition from structured mesh generation (Knupp 2001), ensures each cell maintains proper orientation.

Definition 1 (Fold-Free Cell). Each grid cell with corners (x_{jk}, y_{jk}) , $(x_{j+1,k}, y_{j+1,k})$, $(x_{j,k+1}, y_{j,k+1})$, $(x_{j+1,k+1}, y_{j+1,k+1})$ is *fold-free* (also called *valid* or *orientation-preserving*) if the Jacobian determinant $\mathcal{J}(s, t)$ of its bilinear coordinate map is positive at all four corners $(s, t) \in \{0, 1\}^2$. At the lower-left corner, this reduces to

$$\mathcal{J}_{jk} \equiv (x_{j+1,k} - x_{jk})(y_{j,k+1} - y_{jk}) - (x_{j,k+1} - x_{jk})(y_{j+1,k} - y_{jk}) > 0. \quad (23)$$

The remaining three conditions use the cell's edge vectors at the corresponding vertices. A grid is *fold-free* when every one of its cells is.

Four-corner positivity rules out a fold; Lemma 1 shows that it delivers $\mathcal{J}(s, t) > 0$ throughout the cell.¹⁷

Fold-freeness ensures local cell validity, but ENGINE also requires that bracketing along rows and columns be well-defined. IPM ensures coordinates increase monotonically along each index direction.

¹⁷The determinant $\mathcal{J}_{jk}$ equals twice the signed area of the triangle formed by the lower-left, lower-right, and upper-left vertices. The four-corner condition is standard in structured mesh generation (Knupp 2001), is stronger than lower-left positivity alone, and is exactly the classical necessary and sufficient condition for invertibility of the bilinear cell map (Knupp 1990).

Definition 2 (Index-Preserved Monotonicity (IPM)). A grid $\{(x_{jk}, y_{jk}) : j = 1, \dots, J, k = 1, \dots, K\}$ satisfies Index-Preserved Monotonicity if:

1. **Row monotonicity:** for each fixed k , the first coordinate increases along the row, $x_{j+1,k} > x_{jk}$ for all $j \in \{1, \dots, J - 1\}$.
2. **Column monotonicity:** for each fixed j , the second coordinate increases along the column, $y_{j,k+1} > y_{jk}$ for all $k \in \{1, \dots, K - 1\}$.

Strict inequalities ensure non-degenerate grid spacing. Weak IPM (allowing equalities) accommodates constraint regions where optimal policies are constant, but requires care to avoid degenerate cells.¹⁸

Together, fold-free and IPM ensure the grid mapping is a homeomorphism, a continuous bijection with continuous inverse from index space to physical space, so the sequential brackets are well-defined within the interpolation region.¹⁹ Efficiency also depends on how far a query's bracket can move between adjacent rows. The per-row maximum of that drift, the constant α , determines the search window, and a smaller α reduces search work. Economically, α tends to grow where policies change rapidly relative to grid spacing, near occasionally binding constraints or under high value-function curvature, and when one axis's spacing is much finer than the other's. Refining one axis alone creates exactly that mismatch, so refinement should proceed at a common rate with comparable spacing across axes. The constant governs the single-query search window. By contrast, the cost of the dense regrid in the backward step depends on how far the brackets move in total across its sorted targets.

After each solve I compute the empirical estimate $\hat{\alpha}$, the largest bracket displacement observed between adjacent rows over a dense set of test points, as a mesh diagnostic. On the health-model meshes in Online Appendix A, $\hat{\alpha}$ ranges from 10 to 16, well above the default margin of two. Margins matched to those measurements avoid the fallback searches, and ENGINE completes every reported solve.²⁰

I verify directly whether an inverted stage's recovery map delivers a qualifying mesh. On the two-asset block, exhaustive four-corner and row-and-column monotonicity tests confirm both conditions on every cell of the converged mesh at every reported benchmark refinement (Section 3). On the health model of Online Appendix A, every reported mesh of the two

¹⁸Where a constraint holds a policy constant along a grid dimension, adjacent grid points can share coordinates. I skip zero-width intervals in the bracket search, take one-sided limits at the edges of such regions, and treat each constant-policy region as one effective grid point.

¹⁹IPM is a new term introduced here. The standard characterization of a valid curvilinear coordinate system is that the grid mapping is a homeomorphism.

²⁰A dense measurement is an empirical diagnostic, not a guarantee for untested queries. A guarantee would require a theoretical bound or an exhaustive check of a declared finite query set.

structured methods satisfies IPM, and the only folds on those meshes are sliver cells at the lowest health node. In two high-mortality calibrations of that appendix’s warp sweep, under half a percent of evaluation points fall in those cells, and in the other calibrations none do; the reported residuals include every final query inside one. The measured $\hat{\alpha}$ stays between two and three on every reported two-asset mesh, well below the health meshes’ ten to sixteen. Yet where it reaches three it exceeds the default margin of two, which the containment fallback absorbs. On the health meshes I set the margin from the measurement.

Occasionally binding constraints introduce kinks where the recovery map is not smooth. The implementations build the constrained and interior branches separately; on the two-asset block the checks above then verify the assembled mesh, kink-straddling cells included. In the health model of Online Appendix A the kink falls on a mesh line, so no cell straddles it, and the cells next to it are fold-free in every reported mesh of the two structured methods. Discrete choices instead create jumps that no continuous interpolant should cross. Stages with discrete choices therefore belong to upper-envelope methods such as G2EGM or DCEGM.

4.3 Accuracy and Complexity

On rectangular grids the two passes reduce algebraically to standard bilinear interpolation, whose error for twice continuously differentiable functions is second order in the mesh width (Boor 2001). I measure accuracy on warped meshes directly. Under successive grid doublings the robust off-grid Euler residual falls by about a factor of five per doubling, and the converged policy’s distance to a high-resolution benchmark falls about threefold per doubling (Online Appendix B).

For the dense regrid onto a $P \times Q$ target grid in each backward step, each target costs at most the single-query bound, so the regrid is $O(PQ \log J \cdot \log K)$. Sorted target axes, however, let each bracket search resume where the previous one ended, so in the scaling benchmark the cost falls far below that bound. On meshes with monotone row shifts, whose measured $\hat{\alpha}$ nevertheless rises with mesh size, doubling $J = K$ from 32 to 1024 gives $N^{0.98}$ scaling in $N = JK$, about ten nanoseconds per output value at the largest mesh.

The two-dimensional implementation needs neither a construction step nor Newton iteration.²¹

²¹The implementation uses NumPy (Harris et al. 2020) with Numba (Lam et al. 2015) just-in-time compilation within the Econ-ARK HARK toolkit (Carroll et al. 2018), so comparisons against the toolkit’s curvilinear routine (Curvilinear2DInterp) are between equivalently compiled implementations. The comparison is also matched on warm starts: the curvilinear sector walk begins each query at the previously found sector, just as ENGINE resumes its brackets across sorted targets. When multiple functions share the same endogenous grid, the implementation reuses Pass 1’s cell-location work across all of them.

4.3.1 **Choosing the Interpolation Order**

Practitioners should walk the *regular* axis first, the one mapped to a strictly monotone endogenous coordinate with smaller bracket displacement. In the two-asset Stage 0 recovery domain, this is the liquid-resources axis ℓ ; adjustment costs warp the incoming illiquid coordinate.

One limiting case makes the second pass trivial. If the second coordinate is aligned across rows ($y_{jk} = y_k$) with the shared ordinates increasing ($y_{k+1} > y_k$), and every row satisfies $x_{j+1,k} > x_{jk}$, the aligned family supplies column monotonicity while the ordered rows supply row monotonicity and positive cell orientation.²²

On warped meshes, walking the regular axis first reduces bracket displacement and the effective constant α . Reversal keeps IPM in the tested meshes but raises search work toward the $O(\log J \cdot \log K)$ worst case. It also changes the interpolant because sequential one-dimensional interpolation does not commute across warped coordinate lines. Re-solving the two-asset block in reverse order changes total solve time by less than ten percent at every grid, while the median relative consumption difference stays below 0.15 percent and robust Euler medians remain within two percent (Online Appendix B). In this application, order governs the worst-case search cost rather than converged accuracy.

4.3.2 **Higher Dimensions**

The sequential construction extends algorithmically to higher dimensions through a single one-dimensional pass per dimension. Its efficiency would require controlling bracket drift along each added dimension; the implementation and quantitative claims here are two-dimensional.

4.4 **Comparison with Alternative Interpolants**

The three approaches to multidimensional EGM interpolation differ in how much of the endogenous grid's structure they exploit. On a fold-free mesh satisfying IPM, the endogenous grid is the image of the rectilinear exogenous grid under a homeomorphism (Theorem 1), so the rectangular index lattice survives intact and every row and column remains sorted. Delaunay triangulation uses none of this structure; the curvilinear method uses adjacency (which cell neighbors which) but not order; ENGINE uses both:

²²On a target grid sharing the aligned y_k axis, Pass 2 evaluates at those same ordinates, so only one-dimensional non-uniform interpolation along the ordered rows remains. Beyond index alignment, the aligned ordinates and the other coordinate family must also be strictly monotone.

Table 2 Computational cost and grid requirements of multidimensional interpolants.

Method	Construction	Query cost	Interpolation	Grid requirements
ENGINE (this paper)	$O(1)$	$O(\log J \cdot \log K)$	Sequential linear	Fold-free + IPM
Curvilinear (White 2015)	$O(JK)$	$O(J + K)$	Bilinear	Fold-free only
Delaunay (Ludwig and Schön 2018)	$O(N \log N)$	$O(\log N)$	Piecewise linear	None

Among these three methods, construction cost rises as the grid requirements fall, because a method that assumes less structure spends computation reconstructing it. ENGINE’s construction is free because EGM supplies the index lattice. The curvilinear method recomputes its per-cell sector data and the triangulation rebuilds order at each backward step. In the scaling benchmark the dense matched regrid runs at output size, while unstructured triangulation carries a construction cost of at least order $N \log N$ before its first query. A lattice-aware $O(N)$ triangulation would rely on the same index structure ENGINE exploits and would still be undefined outside the mesh’s convex hull.

Figure 5 contrasts the geometric work each method performs to reach the same query point.

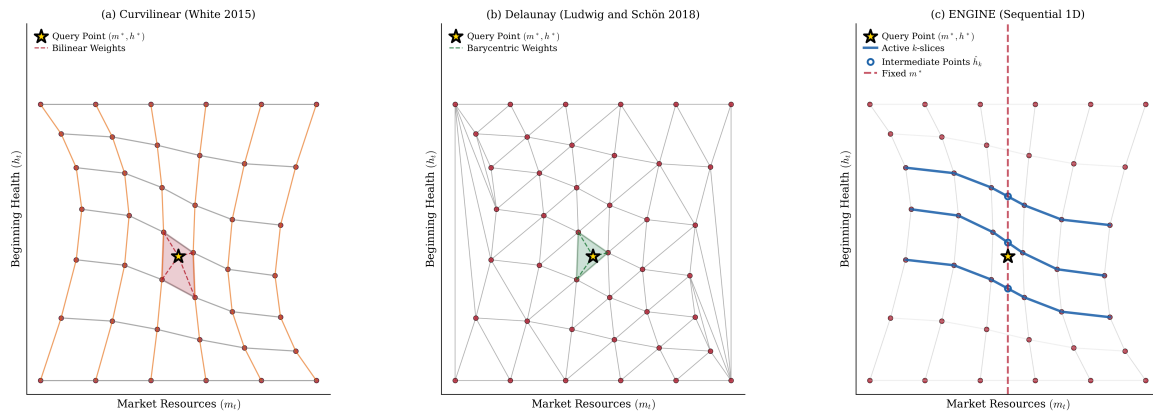


Figure 5 Cell location under three interpolation approaches. The curvilinear method searches for the containing quadrilateral (panel a) and Delaunay triangulation constructs simplices from scratch (panel b); both build a spatial partition and locate the query geometrically. ENGINE needs no spatial partition: it interpolates sequentially along indexed k -slices (highlighted curves) before combining intermediate points (circles) to reach the query point (star) in panel (c).

The solve-time ratio understates the per-query advantage because interpolation is one of several cost components of a backward step, alongside the expectation step and the EGM inversions, so even a large per-query gain translates into a bounded reduction in total solve time. The asymptotic comparison in Table 2 covers only the interpolation component,

where ENGINE can at best tie triangulation. Any comparison-based point location needs at least order $\log N$ comparisons to find the containing cell among N candidates; Delaunay point location attains that floor, and ENGINE's $O(\log J \cdot \log K)$ sits at most a row-search factor above it. The two passes multiply their logarithms rather than add them. Locating the query between rows takes $O(\log K)$ probes, each requiring its own $O(\log J)$ bracket search within a row.

Unlike the solver comparison of Section 3, the full-solve diagnostic of Online Appendix B holds the EGMⁿ solver fixed and swaps only its Stage-0 regrid between the two structured interpolants, scoring both by total solve time to convergence and by a robust Euler error that is valid at constraint-bound queries, since it compares consumption with the lesser of the Euler-implied and constraint-feasible levels.

On accuracy the diagnostic (Table 3) finds ENGINE and the curvilinear method interchangeable, agreeing on the robust Euler error within two percent at every grid and quantile. On speed, ENGINE completes the same solve roughly three to four and a half times faster, widening with the grid. The difference is pure interpolation-method cost, including the comparator's per-step construction, since everything but the regrid step is identical across the solves. Measured directly at the converged meshes, one realign of the regular target grid costs ENGINE 0.02 to 0.26 milliseconds across the three grids against the curvilinear method's 0.54 to 4.7, factors of 18 to 27. Both solves run this step on one thread, even in the sixteen-thread diagnostic, so these timings apply to it unchanged. Realigning alone accounts for more than four fifths of the per-iteration difference between the two solves. The asymptotic entry in Table 2 does not capture this per-call cost. At the converged fixed point, exactly one regular Stage-0 target lies outside the endogenous mesh at each grid. Online Appendix B reports the target totals and outside shares, the protocol, and boundary-sensitivity readings. Extending both axis maxima by half changes medium-grid consumption on the common interior set by 0.05 percent at the median query and by 4.3 percent at the worst.

Table 3 Full solve time and converged-policy accuracy with ENGINE or the curvilinear method of White (2015) as the Stage-0 regrid of an otherwise identical EGMⁿ solve. “Solve (s)” is the best of three warmed solves to tolerance 10^{-9} . Residuals are the median, 95th percentile, and maximum of the $\log_{10}$ robust consumption-unit Euler error at off-grid midpoints below 85 percent of each grid maximum.

Grid	Interpolator	Solve (s)	$\log_{10}$ Euler error		
			Median	p95	Max
Coarse (24 × 20)	ENGINE	0.139	-1.79	-1.10	-1.08
Coarse (24 × 20)	Curvilinear	0.383	-1.79	-1.10	-1.08
Medium (48 × 40)	ENGINE	0.233	-2.47	-1.73	-1.70
Medium (48 × 40)	Curvilinear	0.822	-2.48	-1.73	-1.70
Fine (96 × 80)	ENGINE	0.570	-3.14	-2.38	-2.11
Fine (96 × 80)	Curvilinear	2.540	-3.14	-2.38	-2.11

The two-asset diagnostic includes only the two structured interpolants. Online Appendix A adds the third alternative, Delaunay triangulation, on the health investment model where it and the curvilinear method were originally proposed and tested. I use that model because it favors those two methods, so any advantage ENGINE shows there is conservative. There EGMⁿ with ENGINE solves 2.8 to 4.7 times faster than the simultaneous solver with the curvilinear interpolation of White (2015), and swapping only the interpolant on that solver reproduces the bulk of that speedup. Across nine calibrations chosen to bend the endogenous meshes hardest, the triangulation’s worst consumption residual lies between 0.25 and 0.66, while ENGINE’s never exceeds 0.20. Against the curvilinear method, however, the tails split: ENGINE’s 95th-percentile consumption residual is smaller at seven of nine calibrations, while its maximum is larger at six of nine.

On fold-free meshes that satisfy IPM, ENGINE matches the curvilinear method’s accuracy at lower cost; when IPM fails, ENGINE does not apply. Section 5 asks which economic problems produce the qualifying stages and recovery maps.

5 When the Method Applies

When does EGMⁿ apply? The consumption-labor problem (Section 2) and two-asset problem (Section 3) illustrated two enabling structures: separable utility functions and explicit inversion of a transition marginal followed by state recovery. In economic terms, separability means the marginal utility of one choice does not depend on the level of another, while the second structure recovers a choice or incoming state from how that choice changes the states carried forward. Below, *invertible*

transition marginal is shorthand for this full marginal-inversion and state-recovery condition.

5.1 Splitting the Problem into Subproblems

Decomposition begins by counting the problem's distinct controls. A problem with n control variables might decompose into n subproblems. Do not double-count: the consumption-saving choice ($c + a = m$) is one decision, not two. Similarly, labor-leisure is a single choice despite involving two variables.

Each control variable requires a structure that permits EGM inversion. Two broad sufficient structures are separable utility and an invertible transition marginal in the shorthand just defined. More generally, an interior stage qualifies when its first-order condition has an explicitly evaluable, single-valued inverse, the target lies in its range, the transition identities recover the remaining endogenous state, and feasibility and curvature confirm the candidate as the unique optimum. Where a stage fails these conditions, it uses standard optimization, and the gains accrue only from the stages that invert.²³

The ordering of subproblems should shed state variables early. When subproblems affect disjoint state dimensions, the ordering is immaterial. When they share one, however, poor sequencing propagates unnecessary state variables through later stages. In the consumption-labor problem, placing labor-leisure first resolves the wage rate before the consumption stage, keeping that subproblem one-dimensional. Choosing consumption first would force the labor decision to track both liquid assets and wages, doubling its dimensionality.²⁴

5.2 Formal Conditions

The formal statements below use uppercase calligraphic functions (V , W , U , T , G) for generic value, continuation, utility, within-stage transition, and next-period state functions, and Γ for the feasible-choice correspondence, the set of controls the constraints permit at a given state, with generic state variables x , y , s in place of the model-specific variables:

²³A risky portfolio share is a worked instance of the fallback. Appended to the consumption-labor model of Section 2 in the tradition of Bodie et al. (1992), the share stage carries no utility component of its own, and its first-order condition equates the expected product of the excess return and the marginal value of wealth to zero with the share inside the expectation, so no closed-form inversion exists. The implementation evaluates this condition on a one-dimensional grid of shares, linearly interpolates its first nonnegative-to-nonpositive crossing, and selects zero or one when the endpoint signs imply a corner.

²⁴To see this concretely, placing consumption first makes its subproblem two-dimensional: $v^0(a_t, \theta_t) = \max_{c_t} u(c_t) + v^1(\tilde{a}_t, \theta_t)$ with post-consumption balance $\tilde{a}_t = Ra_t - c_t \geq -\theta_t$, requiring interpolation on an (a_t, θ_t) grid instead of on a_t alone. The labor-leisure subproblem would then carry the feasibility constraint through: $v^1(\tilde{a}_t, \theta_t) = \max_{z_t} h(z_t) + v^2(a_{t+1})$ subject to $0 \leq z_t \leq 1$ and $a_{t+1} = \tilde{a}_t + \theta_t(1 - z_t) \geq 0$.

consumption c , assets a , and cash-on-hand m of preceding sections; the notation reference of the Appendix collects the generic-stage symbols.

5.2.1 **Separable Utility**

Once the problem is split into subproblems, the Endogenous Grid Method applies to each qualifying subproblem during backward iteration from the terminal period.

Consider a generic scalar subproblem with differentiable utility, transition, and continuation-value functions:

$$\begin{aligned} V(x) &= \max_{a \in \Gamma(x)} U(x, a) + W(y) \\ &\text{subject to} \\ y &= T(x, a), \end{aligned} \tag{24}$$

where $W(y) = \beta \mathbb{E}[V_{+1}(y)]$ is the continuation value.²⁵ For an interior solution, the first-order condition is

$$\frac{\partial U(x, a)}{\partial a} + W'(y) \frac{\partial T(x, a)}{\partial a} = 0. \tag{25}$$

For a concave scalar stage with an interval constraint, a corner solution is obtained by projecting the interior candidate onto the feasible interval, as Section 2 demonstrates for leisure.

The question is when this first-order condition can be inverted to yield the optimal control directly. Proposition 1 gives the standard case used in Section 2: separable utility with a transition linear in consumption or labor.

Proposition 1 (Separable Utility). *Suppose $U(x, a) = u_-(x) + u(a)$ is additively separable, so $\phi(a) \equiv u'(a)$ depends only on the control. Assume ϕ^{-1} is single-valued and explicitly evaluable at the target, the recovered control is interior and feasible, and the full stage objective is strictly concave. If $\partial T / \partial a = k \neq 0$ depends only on quantities known before x is recovered, and if $x \mapsto T(x, a)$ has an explicitly evaluable, single-valued inverse at each relevant (y, a) , the optimal control on an exogenous post-decision grid is*

$$a([y]) = \phi^{-1}(-k \cdot W'([y])). \tag{26}$$

The pre-decision state is the unique solution to $T(x([y]), a([y])) = [y]$, written $x([y]) = [T(\cdot, a([y]))]^{-1}([y])$.

Each hypothesis does distinct work in the displayed formula. Because x is recovered endogenously, ϕ may not depend on x . Without strict concavity the first-order condition can hold at a local minimum or at one of several

²⁵Time-additive expected utility is maintained throughout. Epstein-Zin preferences violate it, since the aggregator over current consumption and the certainty equivalent of continuation value is not additive; Lujan (2026) extends EGM to that case via a power transformation.

stationary points, so the formula could return a candidate that is not the optimum, and nothing in the step would flag it. Without a single-valued inverse the step must select among branches, a selection the one-shot formula does not provide. If k depended on x , the right-hand side would contain the state the formula recovers, and the displayed inversion would not close.

Standard budget constraints in consumption-saving models satisfy the linearity assumption $\frac{\partial T}{\partial a} = k$. For example, $a = m - c$ implies $\frac{\partial a}{\partial c} = -1$, so $k = -1$. Similarly, the labor income constraint $m = Ra + \theta \cdot n$ has $\frac{\partial m}{\partial n} = \theta$. The wage shock may enter k because it is observed before the stage is solved; the unrecovered endogenous state may not.

Computationally, the inversion replaces a numerical root-find with a single function evaluation at each grid point.

To see when the method fails, consider non-separable CES preferences (Cobb-Douglas within the period, so the elasticity of substitution is one) $U(c, z) = \frac{(c^\xi z^{1-\xi})^{1-\rho}}{1-\rho}$, where $\xi \in (0, 1)$ is the weight on consumption. The first-order condition for consumption involves leisure: $\frac{\partial U}{\partial c} = \xi c^{\xi(1-\rho)-1} z^{(1-\xi)(1-\rho)}$. Inverting for c requires knowing z , and the symmetric condition for z requires knowing c . The first-order conditions are coupled, so neither can be solved in isolation by EGM inversion, and the stages do not separate. Homotheticity, however, still allows a closed-form solution here: given the wage, the intratemporal condition makes leisure proportional to consumption, and substituting it back leaves a consumption marginal utility that inverts in closed form, the triangular case of [Iskhakov \(2015\)](#). Without such a substitution, coupled conditions require a nested numerical solve (NEGM) or upper envelope techniques (G2EGM).²⁶

5.2.2 Invertible Transition Marginals

Some subproblems have no within-period utility at all, only transitions affecting multiple post-decision states. The illiquid-choice stage of Section 3 is such a case, since moving funds between the accounts has no utility of its own and matters only through the two quantities it sets, the liquid resources left after the adjustment cost and the new illiquid balance. Proposition 2 shows that EGM still applies under the invertible-transition-marginal shorthand defined above, even without separable utility. Consider a problem with two endogenous state variables and two post-decision states, where the first transition may also carry the second state:

²⁶The iEGM of [Hallengreen et al. \(2024\)](#) would solve such a stage by numerical rather than analytical inversion of the marginal utility.

$$\begin{aligned}
V(x_1, x_2, s) &= \max_{a \in \Gamma(x_1, x_2, s)} W(y_1, y_2, s) \\
&\text{subject to} \\
y_1 &= T_1(x_1, x_2, a) \\
y_2 &= T_2(x_2, a),
\end{aligned} \tag{27}$$

where the differentiable continuation value is

$$W(y_1, y_2, s) = \beta \mathbb{E}[V_{+1}(G_1(y_1, s), G_2(y_2, s), s') \mid y_1, y_2, s]. \tag{28}$$

The first-order condition becomes

$$\frac{\partial W(y_1, y_2, s)}{\partial y_1} \cdot \frac{\partial T_1(x_1, x_2, a)}{\partial a} + \frac{\partial W(y_1, y_2, s)}{\partial y_2} \cdot \frac{\partial T_2(x_2, a)}{\partial a} = 0. \tag{29}$$

Proposition 2 (Invertible Transition Marginals). *Suppose that, with post-decision states fixed, the first-order condition contains one unknown and has an explicitly evaluable, single-valued inverse whose target lies in its range. Suppose the transition identities then recover the remaining pre-decision quantities in closed form, and feasibility and curvature confirm the candidate as the unique optimum. The stage solves without root-finding. Two configurations deliver this structure.*

Case (i): off-grid control. *Suppose both transitions are additively separable in the control:*

$$T_1(x_1, a) = f_1(x_1) + k \cdot a, \quad T_2(x_2, a) = f_2(x_2) + g(a), \tag{30}$$

where $k \neq 0$ is constant, f_1 and f_2 are invertible, g' has an explicit single-valued inverse at the displayed target, and $\partial W / \partial y_2 \neq 0$. Then the first-order condition yields

$$a = (g')^{-1} \left(-k \cdot \frac{\partial W(y_1, y_2, s) / \partial y_1}{\partial W(y_1, y_2, s) / \partial y_2} \right), \tag{31}$$

and the pre-decision states follow by inverting f_1 and f_2 .

Case (ii): gridded control. *Suppose instead the control is itself the second post-decision state, and the first transition may couple the control with the not-yet-recovered state:*

$$T_1(x_1, x_2, a) = f_1(x_1) + \kappa(a, x_2), \quad T_2(x_2, a) = a, \tag{32}$$

where f_1 is invertible, $\partial W / \partial y_1 \neq 0$, and the marginal $\kappa_a \equiv \partial \kappa / \partial a$ has an explicit single-valued inverse in x_2 at the displayed target. Then

$$x_2 = \kappa_a(a, \cdot)^{-1} \left(-\frac{\partial W(y_1, y_2, s) / \partial y_2}{\partial W(y_1, y_2, s) / \partial y_1} \right), \quad x_1 = f_1^{-1} (y_1 - \kappa(a, x_2)). \tag{33}$$

*The two configurations are sufficient, not exhaustive.*²⁷

In case (i), additive separability keeps unrecovered states out of the control derivative. Case (ii) permits state-control coupling because the control itself sits on the exogenous grid, leaving only the unrecovered state in the first-order condition.

The literature offers two neighboring standards for when EGM applies. [White \(2015\)](#) characterizes when EGM applies to the period problem as a whole, requiring that the endogenous transition factor through a post-decision state that absorbs the constraints (whether a choice was feasible can be read off the post-decision state alone), with the controls entering that state through a full-rank map. Under those conditions the expectation is computed once per gridpoint. But the first-order system they leave behind may still demand a multidimensional root-find. That residual root-find disappears under the “triangular” transition structure of [Iskhakov \(2015\)](#). Case (i)’s additive separability is that structure, under which the system solves by substitution with no root-finding, though Triangular EGM requires the entire multidimensional first-order-condition system to be triangular. The stage conditions ask less than triangularity and more than the transition decomposition alone. Each stage’s system must collapse to a scalar closed-form inversion. Meeting the stage requirement does not imply meeting the system-level one. In the two-asset application of Section 3 the stage conditions hold, but the full period first-order-condition system is not triangular.

The health-investment stage of Online Appendix A is case (i); the illiquid stage of Section 3 is case (ii). ENGINE’s separate geometric requirements are fold-freeness and index monotonicity, which Section 4 tests cell by cell on every reported mesh. Neither EGMⁿ nor ENGINE applies under nonconcave continuation, which breaks the curvature check that confirms the optimum and can fold the mesh. Under a discrete choice the policy is discontinuous, so the single-valued inversion the stage conditions require fails. The grids such a choice produces need an upper envelope rather than structured interpolation.

5.3 Classification of Solver Classes

The four solver classes differ along three problem features identified by the EGM literature: whether each stage has additively separable utility, whether its transition marginal is invertible in the sense of Proposition 2, and whether the stage has an occasionally binding constraint. Table 4 maps each (feature, solver) cell to the operation that solver applies: an

²⁷Strict monotonicity ensures the inverse is well-defined in both cases: in case (i) g' is strictly decreasing; in case (ii) the map $a \mapsto \Psi_1(a', a) = \chi_1(a' - R_a a) / (R_a a + \chi_0)$, at fixed a' , is strictly decreasing in the incoming stock ($\partial \Psi_1 / \partial a = -\chi_1 R_a (a' + \chi_0) / (R_a a + \chi_0)^2 < 0$), with the inverse available in closed form as the single division $u^* = \tau / \chi_1$ followed by a linear solve for the stock.

inversion recovers the policy in closed form from the first-order condition, a nested root-find wraps an outer one-dimensional search around the next stage's solution, and an upper envelope adjudicates across candidate constraint regimes.

Table 4 Solver class as a function of stage structure. The comparison covers Triangular EGM (Iskhakov 2015), NEGM (Druedahl 2021), G2EGM (Druedahl and Jrgensen 2017), and EGMⁿ. Each cell names the operation applied to a stage with the row's feature; the operations are defined in the text. An Invertible Transition Marginals stage has no within-period utility and couples post-decision states through a differentiable transition; Triangular EGM requires the full first-order-condition system to be triangular. "Not developed" means the published analysis assumes interior solutions, while EGMⁿ selects the corner from the sign of the inverted policy (Section 3).

Stage feature	Triangular EGM	NEGM	G2EGM	EGM ⁿ (this paper)
Separable utility (Proposition 1)	inversion, if system triangular	inversion (consumption only)	inversion + upper envelope	inversion
Invertible transition marginal (Proposition 2)	inversion, if system triangular	nested root-find	inversion + upper envelope	inversion
Occasionally binding constraint	not developed	corner via outer root-find	corner via upper envelope	corner reads next stage

In the two-asset example, EGMⁿ evaluates an explicit inverse where NEGM nests a root-find.

For n independent scalar bounds under this sequential structure, a simultaneous upper envelope considers up to 2^n binding regimes, while the decomposition handles two branches in each of n stages. At $n = 2$ the counts coincide, so the Section 3 timing comparison does not measure this distinction. This is an architectural count, not a general complexity theorem: it requires that no cross-stage constraint regime be carried forward.

For a new model, the conditions work as a checklist: count the controls, order the stages to shed state variables early, and test each stage against the conditions stated at the start of this section, of which Proposition 1 and Proposition 2 are sufficient cases. A stage that passes inverts in closed form, while one that fails keeps standard optimization. ENGINE then interpolates any resulting mesh that is fold-free and index-monotone.

6 Conclusion

Economically simultaneous decisions need not be computationally joint. Because both of the consumption-labor problem's first-order conditions invert analytically, EGM^n needs neither grid search nor root-finding at either stage. The leisure margin receives the same closed-form EGM inversion that [Carroll \(2006\)](#) applied to consumption. The two-asset block replaces the nested method's embedded numerical search with a second closed-form inversion. Section 3 documents the speedup at matched grids and tolerance: the serial gap is roughly a factor of thirteen, and sixteen matched threads still leave EGM^n 3.7 to 7.9 times faster. There the two solvers agree to a median of under half a percent on both asset margins; EGM^n carries the smaller regular-grid liquid Euler residual.

With EGM^n held fixed and only the Stage-0 regrid swapped, ENGINE cuts full-solve time by a factor of roughly three to four and a half with robust Euler errors within two percent. On the health model, EGM^n with ENGINE solves 2.8 to 4.7 times faster than the simultaneous solver with the curvilinear method.

EGM^n applies only when a stage has an explicitly evaluable, single-valued first-order-condition inverse and feasibility and curvature confirm the optimum. ENGINE separately requires a fold-free, index-monotone mesh; analytic inversion alone does not guarantee one. Both conditions are instead checked directly, cell by cell, on every reported mesh.

Stages that still require numerical optimization gain only the decomposition itself, which shrinks the grid search or root-find to the stage's own lower dimension, the observation on which NEGM builds ([Drue Dahl 2021](#)). Discrete choices and severe non-convexities break the monotonicity both algorithms require and belong to G2EGM ([Drue Dahl and Jrgensen 2017](#)); in the paper's consumption-labor and two-asset applications, the leisure and asset bounds are handled on separate branches that preserve the required ordering. Repeated estimation solves can accumulate the per-solve savings when their household blocks retain this stage structure. The two-asset consumption model of [Kaplan and Violante \(2014\)](#), the setting closest to this paper's own application, turns on exactly this kind of interacting liquid and illiquid choice; life-cycle wealth accumulation ([Cagetti 2003](#)) and household balance-sheet dynamics ([Fagereng et al. 2021](#)) pose the same structure.

Both algorithms are implemented in the Econ-ARK HARK toolkit ([Carroll et al. 2018](#)), and the replication archive reproduces every figure and table in this paper.

Two extensions look tractable, one widening the problem class and one economizing the grid. A hybrid would invert the separable stages and

hand only the stage carrying a discrete choice to G2EGM's upper envelope, which would reach problems that mix the two structures without paying the envelope's cost at every stage. Adaptive refinement would concentrate grid points where the policy bends, a location the warp sweep of Online Appendix A already measures. Can the index correspondence be rebuilt cheaply enough between refinements to keep ENGINE's arithmetic advantage?

Declarations

During the preparation of this work, the author used Claude Code (Anthropic) to edit and proofread the manuscript and assist with coding. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the published article.

Acknowledgments

I thank the editors, Thomas Lubik and Eric Young, and three anonymous referees at the *Journal of Economic Dynamics and Control*, whose reports substantially improved and streamlined the exposition of this work. I thank Chris Carroll, Matthew White, and Fedor Iskhakov for their helpful comments and suggestions. The remaining errors are my own. This paper was awarded the [Outstanding Graduate Student Paper Award](#) at the 29th International Conference on Computing in Economics and Finance (CEF 2023, Nice, France). Presentations at the Johns Hopkins University Macro Brownbag (2024), the [International Association for Applied Econometrics annual conference](#) (2024, Thessaloniki, Greece), and the [University of Texas at Austin MA in Economics Program 10-Year Reunion Celebration](#) (2024) improved the exposition and applications. During the development of this work, I received support through the Econ-ARK project, funded by a grant from the [Alfred P. Sloan Foundation](#). All figures and other numerical results were produced using the [Econ-ARK HARK toolkit](#) (Carroll et al. 2018).

References

- Aiyagari, S. Rao. 1994. "Uninsured Idiosyncratic Risk and Aggregate Saving." *The Quarterly Journal of Economics* 109 (3): 659–84. <https://doi.org/10.2307/2118417>.
- Auclert, Adrien, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub. 2021. "Using the Sequence-Space Jacobian to Solve and Estimate Heterogeneous-Agent Models." *Econometrica* 89 (5): 2375–408. <https://doi.org/10.3982/ECTA17434>.
- Azinovic, Marlon, Luca Gaegauf, and Simon Scheidegger. 2022. "Deep Equilibrium Nets." *International Economic Review* 63 (4): 1471–525. <https://doi.org/10.1111/iere.12575>.
- Barillas, Francisco, and Jesús Fernández-Villaverde. 2007. "A Generalization of the Endogenous Grid Method." *Journal of Economic Dynamics and Control* 31 (8): 2698–712. <https://doi.org/10.1016/j.jedc.2006.08.005>.
- Bayer, Christian, Ralph Luetticke, Maximilian Weiss, and Yannik Winkelmann. 2026. "An Endogenous Gridpoint Method for Distributional Dynamics." *Journal of Monetary Economics* 158 : 103895. <https://doi.org/10.1016/j.jmoneco.2026.103895>.
- Bellman, Richard. 1957. *Dynamic Programming*. Princeton University Press.
- Bodie, Zvi, Robert C. Merton, and William F. Samuelson. 1992. "Labor Supply Flexibility and Portfolio Choice in a Life Cycle Model." *Journal of Economic Dynamics and Control* 16 (3–4): 427–49. [https://doi.org/10.1016/0165-1889\(92\)90044-F](https://doi.org/10.1016/0165-1889(92)90044-F).
- Boor, Carl de. 2001. *A Practical Guide to Splines*. Revised. Springer. <https://doi.org/10.1007/978-1-4612-6333-3>.
- Cagetti, Marco. 2003. "Wealth Accumulation over the Life Cycle and Precautionary Savings." *Journal of Business & Economic Statistics* 21 (3): 339–53. <https://doi.org/10.1198/073500103288619007>.
- Carroll, Christopher D. 2006. "The Method of Endogenous Gridpoints for Solving Dynamic Stochastic Optimization Problems." *Economics Letters* 91 (3): 312–20. <https://doi.org/10.1016/j.econlet.2005.09.013>.
- Carroll, Christopher, Alexander Kaufman, Jacqueline Kazil, Nathan Palmer, and Matthew White. 2018. "The Econ-ARK and HARK: Open Source Tools for Computational Economics." In *Proceedings of the Python in Science Conference*, edited by Fatih Akici, David Lippa, Dillon Niederhut, and M. Pacer, *Proceedings of the Python in Science Conference*. Austin, Texas. <https://doi.org/10.25080/majora-4af1f417-004>.
- Clausen, Andrew, and Carlo Strub. 2020. "Reverse Calculus and Nested Optimization." *Journal of Economic Theory* 187 (105019): 105019. <https://doi.org/10.1016/j.jet.2020.105019>.
- Druehl, Jeppe. 2021. "A Guide on Solving Non-Convex Consumption-Saving Models." *Computational Economics* 58 (3): 747–75. <https://doi.org/10.1007/s10614-020-10045-x>.
- Druehl, Jeppe, and Thomas Høgholm Jørgensen. 2017. "A General Endogenous Grid Method for Multi-Dimensional Models with Non-Convexities and Constraints." *Journal of Economic Dynamics and Control* 74 (January): 87–107. <https://doi.org/10.1016/j.jedc.2016.11.005>.
- Fagereng, Andreas, Martin B. Holm, and Gisle J. Natvik. 2021. "MPC Heterogeneity and Household Balance Sheets." *American Economic Journal: Macroeconomics* 13 (4): 1–54. <https://doi.org/10.1257/mac.20190211>.

- Fella, Giulio. 2014. "A Generalized Endogenous Grid Method for Non-Smooth and Non-Concave Problems." *Review of Economic Dynamics* 17 (2): 329–44. <https://doi.org/10.1016/j.red.2013.07.001>.
- Gale, David, and Hukukane Nikaidô. 1965. "The Jacobian Matrix and Global Univalence of Mappings." *Mathematische Annalen* 159 (2): 81–93. <https://doi.org/10.1007/BF01360282>.
- Gowda, M. Seetharama, and G. Ravindran. 2000. "Algebraic Univalence Theorems for Nonsmooth Functions." *Journal of Mathematical Analysis and Applications* 252 (2): 917–35. <https://doi.org/10.1006/jmaa.2000.7171>.
- Hallengreen, Adam, Thomas H. Jrgensen, and Annasofie M. Olesen. 2024. *The Endogenous Grid Method Without Analytical Inverse Marginal Utility*. Working Paper Nos. 24–11. <https://doi.org/10.2139/ssrn.4830404>.
- Harris, Charles R., K. Jarrod Millman, Stéfan J. van der Walt, et al. 2020. "Array Programming with NumPy." *Nature* 585 (7825): 357–62. <https://doi.org/10.1038/s41586-020-2649-2>.
- Hintermaier, Thomas, and Winfried Koeniger. 2010. "The Method of Endogenous Gridpoints with Occasionally Binding Constraints among Endogenous Variables." *Journal of Economic Dynamics and Control* 34 (10): 2074–88. <https://doi.org/10.1016/j.jedc.2010.05.002>.
- Huggett, Mark. 1993. "The Risk-Free Rate in Heterogeneous-Agent Incomplete-Insurance Economies." *Journal of Economic Dynamics and Control* 17 (5–6): 953–69. [https://doi.org/10.1016/0165-1889\(93\)90024-M](https://doi.org/10.1016/0165-1889(93)90024-M).
- Iskhakov, Fedor. 2015. "Multidimensional Endogenous Gridpoint Method: Solving Triangular Dynamic Stochastic Optimization Problems Without Root-Finding Operations." *Economics Letters* 135 (October): 72–76. <https://doi.org/10.1016/j.econlet.2015.07.033>.
- Iskhakov, Fedor, Thomas H. Jrgensen, John Rust, and Bertel Schjerning. 2017. "The Endogenous Grid Method for Discrete-Continuous Dynamic Choice Models with (Or Without) Taste Shocks." *Quantitative Economics* 8 (2): 317–65. <https://doi.org/10.3982/QE643>.
- Jrgensen, Thomas H. 2013. "Structural Estimation of Continuous Choice Models: Evaluating the EGM and MPEC." *Economics Letters* 119 (3): 287–90. <https://doi.org/10.1016/j.econlet.2013.02.027>.
- Kaplan, Greg, and Giovanni L. Violante. 2014. "A Model of the Consumption Response to Fiscal Stimulus Payments." *Econometrica* 82 (4): 1199–239. <https://doi.org/10.3982/ECTA10528>.
- Kaplan, Greg, Benjamin Moll, and Giovanni L. Violante. 2018. "Monetary Policy According to HANK." *American Economic Review* 108 (3): 697–743. <https://doi.org/10.1257/aer.20160042>.
- Knupp, Patrick M. 1990. "On the Invertibility of the Isoparametric Map." *Computer Methods in Applied Mechanics and Engineering* 78 (3): 313–29. [https://doi.org/10.1016/0045-7825\(90\)90004-6](https://doi.org/10.1016/0045-7825(90)90004-6).
- Knupp, Patrick M. 2001. "Algebraic Mesh Quality Metrics." *SIAM Journal on Scientific Computing* 23 (1): 193–218. <https://doi.org/10.1137/S1064827500371499>.
- Krueger, Dirk, Kurt Mitman, and Fabrizio Perri. 2016. "Macroeconomics and Household Heterogeneity." In *Handbook of Macroeconomics*, vol. 2. Elsevier. <https://doi.org/10.1016/bs.hesmac.2016.04.003>.

- Krusell, Per, and Anthony A. Smith. 1998. "Income and Wealth Heterogeneity in the Macroeconomy." *Journal of Political Economy* 106 (5): 867–96. <https://doi.org/10.1086/250034>.
- Lam, Siu Kwan, Antoine Pitrou, and Stanley Seibert. 2015. "Numba: A LLVM-Based Python JIT Compiler." In "Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC." Special issue, *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (Austin, Texas), LLVM '15, no. Article 7: 1–6. <https://doi.org/10.1145/2833157.2833162>.
- Ludwig, Alexander, and Matthias Schön. 2018. "Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods." *Computational Economics* 51 (3): 463–92. <https://doi.org/10.1007/s10614-016-9611-2>.
- Lujan, Alan. 2026. "The Endogenous Grid Method for Epstein-Zin Preferences." <https://doi.org/10.48550/arXiv.2601.04438>.
- Maliar, Lilia, and Serguei Maliar. 2013. "Envelope Condition Method Versus Endogenous Grid Method for Solving Dynamic Programming Problems." *Economics Letters* 120 (2): 262–66. <https://doi.org/10.1016/j.econlet.2013.04.031>.
- Maliar, Lilia, and Serguei Maliar. 2014. "Chapter 7 - Numerical Methods for Large-Scale Dynamic Economic Models." In *Handbook of Computational Economics*, vol. 3. Handbook of Computational Economics. Elsevier. <https://doi.org/10.1016/B978-0-444-52980-0.00007-4>.
- Maliar, Lilia, Serguei Maliar, and Pablo Winant. 2021. "Deep Learning for Solving Dynamic Economic Models." *Journal of Monetary Economics* 122 (September): 76–101. <https://doi.org/10.1016/j.jmoneco.2021.07.004>.
- Mendoza, Enrique G., and Sergio Villalvazo. 2020. "Fipit: A Simple, Fast Global Method for Solving Models with Two Endogenous States & Occasionally Binding Constraints." *Review of Economic Dynamics* 37 (July): 81–102. <https://doi.org/10.1016/j.red.2020.01.001>.
- Powell, Warren B. 2011. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. 2nd ed. Wiley Series in Probability and Statistics. Wiley. <https://doi.org/10.1002/9781118029176>.
- Preparata, Franco P., and Michael Ian Shamos. 1985. *Computational Geometry: An Introduction*. Texts and Monographs in Computer Science. Springer. <https://doi.org/10.1007/978-1-4612-1098-6>.
- Scheidegger, Simon, and Ilias Bilionis. 2019. "Machine Learning for High-Dimensional Dynamic Stochastic Economies." *Journal of Computational Science* 33 (April): 68–82. <https://doi.org/10.1016/j.jocs.2019.03.004>.
- White, Matthew N. 2015. "The Method of Endogenous Gridpoints in Theory and Practice." *Journal of Economic Dynamics and Control* 60 (November): 26–41. <https://doi.org/10.1016/j.jedc.2015.08.001>.
- Young, Eric R. 2010. "Solving the Incomplete Markets Model with Aggregate Uncertainty Using the Krusell-Smith Algorithm and Non-Stochastic Simulations." *Journal of Economic Dynamics and Control* 34 (1): 36–41. <https://doi.org/10.1016/j.jedc.2008.11.010>.

7 Appendix

7.1 The Grid Homeomorphism Theorem

Lemma 1 (Vertical-Line Monotonicity). *Let a fold-free curvilinear grid satisfy IPM. Extend its index map Ξ , defined by $\Xi(j, k) = (x_{jk}, y_{jk})$, to $\tilde{\Xi}$ by bilinear interpolation within each cell, parameterizing each cell by coordinates $(s, t) \in [0, 1]^2$ with s along the row index and t along the column index. Then within every cell $\partial x / \partial s > 0$, $\partial y / \partial t > 0$, and $\mathcal{J} \equiv \det D\tilde{\Xi} > 0$. Call the interval from the largest first-node abscissa to the smallest last-node abscissa of a set of rows their common span. For an abscissa x_0 in the common span of the rows bounding a run of bands, the preimage of $x = x_0$ crosses those bands in a single continuous curve along which the y -value increases strictly. Symmetrically, the x -value along the preimage of an ordinate in the common span of a run of columns, defined from the columns' first and last ordinates, increases strictly across those column bands.*

Proof. Within each cell $[j, j + 1] \times [k, k + 1]$, the bilinear map in parametric coordinates $(s, t) \in [0, 1]^2$ takes the form $x(s, t) = (1 - s)(1 - t)x_{jk} + s(1 - t)x_{j+1,k} + (1 - s)t x_{j,k+1} + st x_{j+1,k+1}$, and similarly for y . The partial derivative $\partial x / \partial s$ is a convex combination of the row increments $x_{j+1,k} - x_{jk}$ and $x_{j+1,k+1} - x_{j,k+1}$, both strictly positive by IPM row monotonicity, so $\partial x / \partial s > 0$ throughout the cell; the column argument gives $\partial y / \partial t > 0$. The determinant $\mathcal{J}$ is an affine function of (s, t) , so its minimum over the cell occurs at a corner; fold-freeness therefore gives $\mathcal{J} > 0$ throughout.

Fix a band between rows k and $k + 1$ and write $t \in [0, 1]$ for the position between them. At each fixed t the section $s \mapsto x(s, t)$ is continuous and strictly increasing, since $\partial x / \partial s > 0$ throughout the band, and its endpoint values are the linear interpolations in t of the two rows' first and last abscissas. An x_0 in the two rows' common span therefore lies between the section's endpoints at every t , so the intermediate value theorem yields a crossing $x(s(t), t) = x_0$, unique by strict monotonicity. Within each cell the crossing function $s(\cdot)$ is differentiable with $ds/dt = -(\partial x / \partial t) / (\partial x / \partial s)$ by the implicit function theorem, so the y -value along the curve changes at rate $dy/dt = \mathcal{J}(s, t) / (\partial x / \partial s) > 0$. Because adjacent bilinear pieces agree on shared edges, $s(\cdot)$ is continuous across cell columns, and the crossings form one continuous curve over the band along which y increases strictly. The curve ends on the shared row, which is the next band's own lower edge, so when x_0 also lies in the next pair's common span the construction continues and the concatenated curve's y -value increases strictly across bands. The symmetric argument, with rate $\mathcal{J}(s, t) / (\partial y / \partial t) > 0$ along the horizontal level curve, gives the column statement. $\square$

Theorem 1 (Grid Homeomorphism). *A fold-free curvilinear grid satisfying IPM is homeomorphic to the rectangular index grid. That is, the mapping $\Xi : \{1, \dots, J\} \times \{1, \dots, K\} \rightarrow \mathbb{R}^2$ defined by $\Xi(j, k) = (x_{jk}, y_{jk})$ is injective, and*

extending Ξ via bilinear interpolation within each cell yields a homeomorphism from $[1, J] \times [1, K]$ to the grid's image.

Proof. The bilinear extension $\tilde{\Xi} : [1, J] \times [1, K] \rightarrow \mathbb{R}^2$ is continuous and piecewise C^1 . Within each cell, Lemma 1 gives

$$\frac{\partial x}{\partial s} > 0, \quad \frac{\partial y}{\partial t} > 0, \quad \det D\tilde{\Xi} > 0. \quad (34)$$

Thus every cell-interior Jacobian has positive diagonal entries and a positive determinant. Four positive corner Jacobians are already the classical necessary and sufficient condition for invertibility of a single bilinear cell map (Knupp 1990); the remaining question is global: can two distant cells map to the same physical point? At a point interior to a cell the map is differentiable and its Jacobian carries that sign pattern. At every other point of the closed rectangle, on interior seams and grid vertices and the outer boundary alike, the nonsmooth form of the theorem works with the convex combinations of the one-sided Jacobians the incident cells contribute, and the sign pattern survives the mixing. Both diagonal entries do because each is a convex combination of positive edge increments. The determinant does for a structural reason. Across a seam the two incident Jacobians share the column tangent to that seam, since the two bilinear pieces agree along it, so the determinant of any mixture is the same mixture of the two positive one-sided determinants. At a grid vertex, any mixture of the four incident Jacobians has one column that is a convex combination of the two horizontal edge vectors meeting there and one that is a convex combination of the two vertical edge vectors, so its determinant expands into a positively weighted sum of the four incident corner determinants, each positive by fold-freeness. The theorem of Gale and Nikaidô (1965), in the nonsmooth form of Gowda and Ravindran (2000), therefore applies on the compact rectangle and makes $\tilde{\Xi}$ globally one-to-one. Finally, a continuous injection from a compact space into the plane is a homeomorphism onto its image. $\square$

The direct four-corner calculation verifies fold-freeness on every cell of each finite mesh reported in Section 3, kink-straddling cells included.

7.2 The ENGINE Algorithm

Section 4 develops ENGINE and states its probe costs; this section records the single-query listing and the boundary behavior. The dense regrids in the benchmarks run the same two passes with brackets carried forward across sorted targets, as Section 4 describes.

Algorithm (ENGINE Two-Pass Interpolation). *Inputs:* grid $\{(x_{jk}, y_{jk}, f_{jk})\}$ for $j = 1, \dots, J$ and $k = 1, \dots, K$; query point (x^*, y^*) . *Output:* interpolated value $f(x^*, y^*)$. The listing uses subscripts l , m , and h for the low, mid, and high

indices. The routine BracketSearch verifies that the bracket it returns contains x^* and falls back to a full row search on a miss.

Pass 1: binary search on rows.

1. Set $k_l \leftarrow 1$ and $k_h \leftarrow K$.
2. Locate the initial bracket at the bottom row, $j_l \leftarrow \text{BinarySearch}\left(x^*, \{x_{j,k_l}\}_j\right)$, and compute the intermediate coordinate $\hat{y}_l \leftarrow \text{Interp1D}\left(x^*; x_{j_l,k_l}, x_{j_l+1,k_l}, y_{j_l,k_l}, y_{j_l+1,k_l}\right)$.
3. While $k_h - k_l > 1$:
 1. Set $k_m \leftarrow \lfloor (k_l + k_h)/2 \rfloor$.
 2. Find $j_m \leftarrow \text{BracketSearch}\left(x^*, \{x_{j,k_m}\}_j\right)$, searching near j_l within the configured bracket-displacement window.
 3. Compute $\hat{y}_m \leftarrow \text{Interp1D}\left(x^*; x_{j_m,k_m}, x_{j_m+1,k_m}, y_{j_m,k_m}, y_{j_m+1,k_m}\right)$.
 4. If the intermediate ordinate satisfies $\hat{y}_m \leq y^*$, set $k_l \leftarrow k_m, j_l \leftarrow j_m$, and $\hat{y}_l \leftarrow \hat{y}_m$; otherwise set $k_h \leftarrow k_m$.

Pass 2: final interpolation.

1. Find $j_h \leftarrow \text{BracketSearch}\left(x^*, \{x_{j,k_h}\}_j\right)$.
2. Compute $\hat{y}_h \leftarrow \text{Interp1D}\left(x^*; x_{j_h,k_h}, x_{j_h+1,k_h}, y_{j_h,k_h}, y_{j_h+1,k_h}\right)$.
3. Compute $\hat{f}_l \leftarrow \text{Interp1D}\left(x^*; x_{j_l,k_l}, x_{j_l+1,k_l}, f_{j_l,k_l}, f_{j_l+1,k_l}\right)$.
4. Compute $\hat{f}_h \leftarrow \text{Interp1D}\left(x^*; x_{j_h,k_h}, x_{j_h+1,k_h}, f_{j_h,k_h}, f_{j_h+1,k_h}\right)$.
5. Set $t \leftarrow (y^* - \hat{y}_l)/(\hat{y}_h - \hat{y}_l)$ and return $(1 - t)\hat{f}_l + t\hat{f}_h$.

EGM applications require an extrapolation rule: simulation occasionally produces state values slightly outside the solution grid because of shock realizations or numerical precision. Each 1D interpolation step handles out-of-bounds queries by extending the boundary slope, a continuous extension beyond the grid's convex hull: for $x^* < x_1$, the value is $f_1 + (x^* - x_1)(f_2 - f_1)/(x_2 - x_1)$, and symmetrically for $x^* > x_J$.

Linear extrapolation is unreliable near binding constraints, where policies have kinks and high curvature, and accurate in wealth-rich regions, where precautionary motives fade and policies become nearly linear. The asset grids in this paper extend well past their constraint regions, which confines any extrapolation to that smooth region.

7.3 Notation Reference

Table 5 Notation reference.

Symbol	Meaning
<i>Conventions</i>	
m', a'	Prime: next-period quantity
c, m, a	Gothic: endogenous grid, built by inverting a condition
$[a], [m]$	Brackets: exogenous grid
<i>Section 2, consumption-labor</i>	
ρ, β, R	Risk aversion; discount factor; gross risk-free return
m_t, a_t	Market resources; liquid assets carried in
z_t, n_t	Leisure; labor supply ($n = 1 - z$)
h, v, ζ	Leisure utility; its scale and curvature
λ_t	Endogenous leisure from the first-order-condition inversion
θ_t	Wage shock
v^0, v^1, v^2	Stage value functions; the superscript labels the stage
<i>Section 3, two-asset household</i>	
a, a'	Illiquid balance; illiquid choice (liquid in Section 2)
b, b'	Liquid balance; liquid choice
R_a, R_b	Gross illiquid and liquid returns
m, ℓ	Market resources; liquid resources after the illiquid choice
l	Endogenous liquid-resources grid ($b' + c$)
$e, \Pi(e' e)$	Productivity state; Markov transition
Ψ, χ_0, χ_1	Illiquid adjustment cost (Ψ_1, Ψ_{12} : argument partials); its offset and scale
u, τ	Standardized adjustment wedge; wedge target, the adjustment-cost shadow price
v_ℓ^1, v_a^1	Stage-1 partials (subscripts; stage label in the superscript); w replaces v^2 here
$w, w^{b'}$	End-of-period value; its superscripts denote partials (no stage label)
<i>Section 4, ENGINE</i>	
j, k, J, K	Exogenous-grid column and row indices; their sizes
P, Q, N	Target-grid sizes of the dense regrid; total mesh size $N = JK$
$\alpha, \hat{\alpha}$	Bracket displacement between adjacent rows; its measured value
<i>Section 5 and Online Appendix A, generic stage</i>	
x, y, s, a	Pre-decision, post-decision, exogenous state; control
V, U, T, W, G	Stage value; utility; within-stage transition; continuation value; next-period state map
Γ	Feasible-choice correspondence
g, κ	Control-only and state-coupled transition terms (Invertible Transition Marginals proposition)

ϕ	Marginal utility of the control, $\phi(a) \equiv u'(a)$
k, f_1, f_2	Constant transition slope; invertible transition components (the two propositions)
H, h	Second state, post- and pre-transition; health in the testbed
n, n	Health investment (Online Appendix A; letter reuse with Section 2 labor n)
