Journal of Economic Dynamics & Control 60 (2015) 26–41

Contents lists available at ScienceDirect

Journal of Economic Dynamics & Control

journal homepage: www.elsevier.com/locate/jedc

The method of endogenous gridpoints in theory and practice ☆

Matthew N. White

University of Delaware, Department of Economics, 416B Purnell Hall, Newark, DE 19711, United States

ARTICLE INFO

Article history:
Received 7 February 2015
Received in revised form
30 July 2015
Accepted 4 August 2015
Available online 12 August 2015

JEL classification:
C61
C63
D90

Keywords:
Dynamic models
Numeric solution
Endogenous gridpoint method
Non-linear grid interpolation
Endogenous human capital

ABSTRACT

The method of endogenous gridpoints (ENDG) significantly speeds up the solution to dynamic stochastic optimization problems with continuous state and control variables by avoiding repeated computations of expected outcomes while searching for optimal policy functions. I provide an interpolation technique for non-rectilinear grids that allow ENDG to be used in n-dimensional problems in an intuitive and computationally efficient way: the acceleration of ENDG with non-linear grid interpolation is nearly constant in the density of the grid. Further, ENDG has only been shown by example and has never been formally characterized. Using a theoretical framework for dynamic stochastic optimization problems, I formalize the method of endogenous gridpoints and present conditions for the class of models for which it can be used.

1. Introduction

The traditional solution method for a dynamic stochastic optimization problem is for the researcher to choose a set of gridpoints in the state space, solve for the optimal action at each of these points, and then generate an approximation to the true policy function through interpolation. Because next period's state is subject to risk and there is usually no closed form solution, this method involves extremely costly root-finding with repeated computation of numeric integrals. The method of endogenous gridpoints (ENDG), first proffered by Carroll (2006),[1] provides an alternative approach that inverts the problem: rather than asking what the optimizing agent should *do now* if he begins the period in state x, the method of endogenous gridpoints asks what an agent ending the period in state z must *have just done* to find himself there, assuming he acted optimally to arrive at z. In this way, the pre-decision state space points are solved endogenously rather than exogenously fixed in advance. The method minimizes the number of numeric integrals needed to solve the problem and often avoids root-finding entirely, greatly accelerating the solution.

However, practical application of ENDG to models with multiple endogenous state dimensions and multiple controls has proved to be problematic. As the policy functions are non-linear, the method produces an irregular array of pre-decision

☆ This paper combines two previous drafts: "Endogenous Gridpoints in Multiple Dimensions: Interpolation on Non-Linear Grids" and "A General Theory of the Method of Endogenous Gridpoints".

E-mail address: mnwecon@udel.edu

[1] Originally applied to a problem with a one dimensional state space and one continuous control variable, ENDG has since been extended multiple times: Barillas and Fernández-Villaverde (2007) describe how to accommodate multiple controls, Fella (2014) allows non-concave value functions, Hintermaier and Koeniger (2010) adjust the method for occasionally binding constraints, while Iskhakov et al. (2012) develop a method that allows one continuous and one discrete control variable.

http://dx.doi.org/10.1016/j.jedc.2015.08.001
0165-1889/

state space points rather than a rectilinear grid that can be used with standard linear interpolation. Moreover, the method has only been shown by example and never fully characterized in a general setting – the conditions under which ENDG can be used have never been formally stated. To address these theoretical and practical holes in the literature, this paper provides two main contributions.

First, I present in Section 3 a technique for interpolating on an irregular array of gridpoints that exploit known relationships among the points: the endogenous pre-decision gridpoints are *ordered* identically to the exogenous post-decision gridpoints that generated them. In two dimensions, the pre-decision gridpoints thus subdivide the state space into a grid of irregular quadrilateral sectors, each of which can be continuously mapped to the unit square to allow standard bilinear interpolation. This technique avoids the long construction or evaluation procedures used in other interpolation methods for irregular arrays. Section 4 demonstrates the computational gains of ENDG in combination with "non-linear grid interpolation" using a two dimensional (2D) model introduced in Section 2.

Second, I formalize the method of endogenous gridpoints in a general framework for dynamic stochastic optimization problems, providing sufficient conditions for its application. Presented and discussed in Section 5, these conditions primarily concern the decomposition of the state transition function into "intraperiod" and "interperiod" components and the accompanying existence of a "post-decision state" that serves as a sufficient statistic of both the state and control variables.

2. Benchmark model

To motivate the need for an alternative interpolation technique, this section presents a model with two endogenous state variables and two controls. For ease of reference, the model is a slight extension of Ludwig and Schön (2014), adding additional interperiod risk.[2]

2.1. Statement of model

The individual is a finitely lived lifetime expected utility maximizer whose state at discrete time t is characterized by his money resources (previous wealth plus current income) m_t and his health capital h_t. He derives a flow of utility from consumption c_t via CRRA utility function $u(c_t)$ with coefficient ρ, and discounts future utility at a rate of β per period. At the beginning of period t, the individual receives labor income of $\omega_t h_t$, where wage rate ω_t is distributed log-normally $\mathcal{N}(\overline{\omega}/(1-\mho), \sigma_\omega^2)$, with a probability $\mho$ point mass at $\omega_t = 0$ representing unemployment. The individual must choose his levels of consumption c_t and investment in health capital i_t so that $m_t - c_t - i_t \geq 0$. Investment generates additional health capital according to production function $f(i_t) = (\gamma/\alpha) i_t^\alpha$.

Any unspent resources accumulate at gross interest factor R, so that next period's resources are $m_{t+1} = R(m_t - c_t - i_t) + \omega_{t+1} h_{t+1}$. Health capital depreciates from one period to the next at depreciation rate $\delta_{t+1} \sim U[\overline{\delta} - \sigma_\delta, \overline{\delta} + \sigma_\delta]$, and so next period's capital is $h_{t+1} = (1-\delta_{t+1})(h_t + f(i_t))$. The individual thus faces both permanent (via δ) and transitory (via ω) income risk. Moreover, he faces a mortality risk based on his health capital, with the probability of survival into period $t+1$ given by $s(h_{t+1}) = 1 - \phi/(1+h_{t+1})$; assume there is a terminal period T beyond which the individual cannot live. If the individual does not survive, he receives no more income, nor can he consume and derive utility.

The agent's problem in any non-terminal[3] period can be written in Bellman form as

$$V_t(m_t, h_t) = \max_{c_t, i_t} u(c_t) + \beta \int s(h_{t+1}) V_{t+1}(m_{t+1}, h_{t+1})\, dF(\omega, \delta) \tag{1}$$

$$\text{s.t. } m_t - c_t - i_t \geq 0, \quad m_{t+1} = R(m_t - c_t - i_t) + \omega h_{t+1}, \quad h_{t+1} = (1-\delta)(h_t + f(i_t)).$$

The optimal policy functions for c_t and i_t are defined by replacing max with arg max in (1).

2.2. Solving the model with ENDG

To solve for optimal behavior using ENDG, begin with the first order conditions of the model:

$$u'(c_t) = \beta R \int s(h_{t+1}) V^m_{t+1}(m_{t+1}, h_{t+1})\, dF(\omega, \delta),$$

$$f'(i_t) = \frac{R \int s(\cdot) V^m_{t+1}(\cdot) dF(\omega, \delta)}{\int (1-\delta)\left[s'(\cdot) V_{t+1}(\cdot) + s(\cdot)\left(\omega V^m_{t+1}(\cdot) + V^h_{t+1}(\cdot) \right) \right] dF(\omega, \delta)} \tag{2}$$

[2] Risk generalizes the model and eliminates the need for a constrained optimization routine, maintaining focus on ENDG and non-linear grid interpolation rather than the complications of dual search routines.

[3] The solution in the terminal period is trivial, yielding a value function of $V_T(m,h) = u(m)$ and marginal value functions of $V^m_T(m,h) = u'(m)$ and $V^h_T(m,h) = 0$; policy functions are $c_T(m,h) = m$ and $i_T(m,h) = 0$.

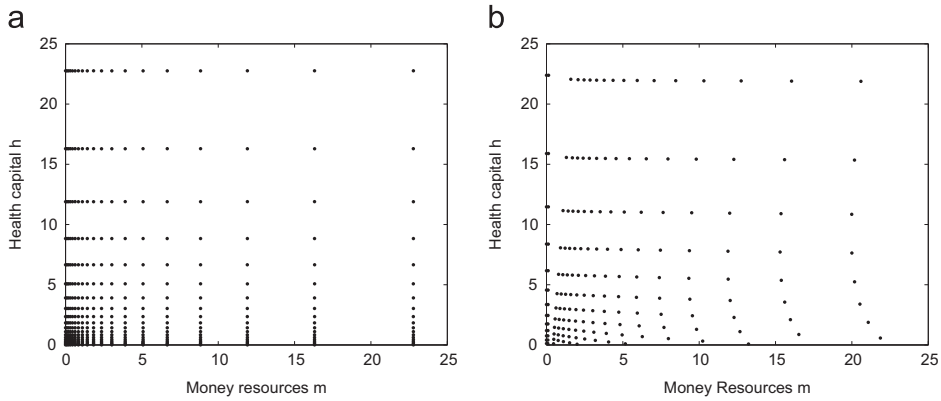

Fig. 1. Interpolation grids formed by the traditional method of exogenous gridpoints (a) and the method of endogenous gridpoints (b).

$$\text{s.t. } m_{t+1} = R(m_t - c_t - i_t) + \omega(1-\delta)(h_t + f(i_t)), \quad h_{t+1} = (1-\delta)(h_t + f(i_t)).$$

For any "pre-decision state" (m_t, h_t), this system has no closed form solution for c_t and i_t as these variables appear on both sides of the equation (and inside integrals on the RHS). An iterative search for the solution involves computing these integrals for each guess of optimal (c_t, i_t); this is obviously a costly procedure that the researcher would like to avoid.

Instead, define a "post-decision state" as a combination of assets (after all actions are accomplished) $a_t = m_t - c_t - i_t$ and post-investment health $H_t = h_t + f(i_t)$; this represents an intermediate state after the individual has carried out his choice of action but before the transition to period $t+1$. The transitions in the third line of (2) simplify to:

$$m_{t+1} = Ra_t + \omega(1-\delta)H_t, \quad h_{t+1} = (1-\delta)H_t. \tag{3}$$

For any post-decision state (a_t, H_t), the first order conditions can now be algebraically solved to yield a closed form solution for both the optimal controls (c_t, i_t) and the pre-decision state (m_t, h_t) from which these actions were taken in order to end the period at (a_t, H_t):

$$c_t = \left(\beta R \int s(h_{t+1}) V^m_{t+1}(m_{t+1}, h_{t+1})\, dF(\omega, \delta)\right)^{-1/\rho},$$

$$i_t = \left(\frac{R \int s(h_{t+1}) V^m_{t+1}(m_{t+1}, h_{t+1})\, dF(\omega, \delta)}{\gamma \int (1-\delta)\left[s'(\cdot) V_{t+1}(\cdot) + s(\cdot)\left(\omega V^m_{t+1}(\cdot) + V^h_{t+1}(\cdot)\right)\right] dF(\omega, \delta)}\right)^{-1/(\alpha-1)}, \tag{4}$$

$$m_t = a_t + c_t + i_t, \quad h_t = H_t - f(i_t).$$

Marginal value with respect to each state variable can be found using the envelope conditions:

$$V^m_t(m_t, h_t) = u'(c_t), \quad V^h_t(m_t, h_t) = u'(c_t)/f'(i_t). \tag{5}$$

In this way, the method of endogenous gridpoints can be used to avoid repeated computations of costly numeric integrals and, in this case, root-finding. Rather than solving the first order conditions on an exogenous grid of pre-decision states, the researcher instead fixes a grid of post-decision states. For each gridpoint, the integrals in (4) must be computed *once*, yielding an optimal control and accompanying endogenous pre-decision gridpoint.

Suppose the method of endogenous gridpoints is used to solve this model in period $T-1$. The right-hand panel of Fig. 1 plots the resulting endogenous gridpoints, which clearly do not form a mesh or an ordinary rectilinear grid. When the level of health for a gridpoint was calculated in (4), curvature was introduced into every row and column of gridpoints through the non-linear investment and production functions; likewise for money resources, which now curve through the pre-decision state space because of the non-linear policy functions. Fig. 1 demonstrates a significant complication of ENDG in models with multiple endogenous state dimensions: ordinary linear interpolation cannot be used, as the "sector" in which a state space point resides cannot be identified with independent bisecting searches, nor can the relative position of the point within the sector be trivially calculated. However, the endogenous gridpoints in Fig. 1(b) are *ordered* identically to the mesh of post-decision gridpoints that generated them: the four vertices of an irregular quadrilateral sector correspond to rectangularly adjacent points in the post-decision grid. The interpolation technique presented below exploits these relationships among the endogenous gridpoints.

3. Interpolation on non-linear grids

This section presents a computationally efficient interpolation technique for the irregular array of pre-decision gridpoints generated by ENDG.[4] I begin with a brief review of interpolation on a rectilinear grid, then extend the method to "non-linear grid interpolation".

3.1. Interpolation on rectilinear grids

Define the (endogenous) state space of the problem as $X \subset \mathbb{R}^n$, with typical element x. Define a *mesh* $\mathcal{J}$ on X as a set of points generated as the cross product of n ordered finite subsets of $\mathbb{R}$, such that $\mathcal{J} \subset X$; that is, $\mathcal{J} = \mathcal{J}^1 \times \cdots \times \mathcal{J}^n$.

First, consider the simple case of a function $f: \mathbb{R} \to \mathbb{R}$. If x^A and x^B are two consecutive gridpoints in a mesh and $x \in [x^A, x^B]$, then the linear interpolation of $f(\cdot)$ at x is

$$\alpha = \frac{x - x^A}{x^B - x^A}, \quad f(x) \approx (f(x^B) - f(x^A))\alpha + f(x^A) = (1-\alpha)f(x^A) + \alpha f(x^B). \tag{6}$$

This effectively maps the gridpoint interval $x \in [x^A, x^B]$ into the unit interval $\alpha \in [0, 1]$.

The extension of linear interpolation to two dimensions uses the points of the mesh to partition the state space into rectangular sectors (see the left panel of Fig. 1). Just as an interval of $\mathbb{R}$ can be continuously mapped into the unit interval, any rectangular subset of $\mathbb{R}^2$ can be translated into the unit square. Defining x_i as the ith index of x the bilinearly interpolated approximation to a function $f: \mathbb{R}^2 \to \mathbb{R}$ on the rectangle defined by lower-left vertex x^A and upper-right vertex x^B is

$$\alpha = \frac{x_1 - x_1^A}{x_1^B - x_1^A}, \quad \beta = \frac{x_2 - x_2^A}{x_2^B - x_2^A}, \tag{7}$$

$$f(x) \approx (1-\alpha)(1-\beta)f(x_1^A, x_2^A) + \alpha(1-\beta)f(x_1^B, x_2^A) + (1-\alpha)\beta f(x_1^A, x_2^B) + \alpha\beta f(x_1^B, x_2^B).$$

Bilinear interpolation approximates the true function by applying linear weights to the known function values at the four vertices. This method can easily be extended to three (or more) dimensions, with each state in a sector mapped to $(\alpha, \beta, \gamma) \in [0, 1]^3$.

Linear interpolation on an ordinary rectilinear grid is extremely simple and fast in any number of dimensions: once the sector for a point x has been identified, both the translation to relative coordinates and evaluation of the interpolation based on these coordinates are trivial steps. Moreover, finding the sector that contains a point is very straightforward, as the n dimensions can be treated as independent bisecting searches; with J values in each grid dimension, the sector can be identified in $O(n \log J)$ time.

The key insight of non-linear grid interpolation is that the irregular array of gridpoints generated by ENDG also divides the state space into sectors with 2^n vertices (e.g. the four corners of a quadrilateral in 2D, or eight vertices in 3D), and that each of these sectors can be continuously mapped into the unit box. That is, any point in the convex hull of the gridpoints is an element of a single "irregular sector" and corresponds to a unique value in $[0, 1]^n$, so that linear interpolation can be applied. The following subsections address the additional complications that arise with an irregular two dimensional grid in identifying the correct sector for a state space point and calculating the relative coordinates (α, β) within that sector.[5]

3.2. Identifying the sector on a non-linear grid

Consider the problem of locating the sector in which some point $x \in X \subset \mathbb{R}^2$ resides. Unlike with an ordinary rectilinear grid, the bounds of the sectors do not align within "rows" or "columns" in a non-linear grid, so we cannot simply perform a bisecting search. However, we can use a very similar technique to that in Ludwig and Schön for Delaunay interpolation (what they refer to as a "visibility walk"). Starting from some initial guess, we search by "walking" from sector to sector, checking whether x is in the current guess and updating the sector based on which of the boundaries it violates.

The left panel of Fig. 2 shows the sector "walking" search process. At the initial sector guess, the point to be located violates the upper and left boundaries, so the guess is updated one sector up and one to the left; at the second guess, only the left boundary is violated, so the sector is updated one to the left. In the third guess, the point is within all four boundaries and the sector has been located, so we can proceed to calculating α and β for this point within that sector. At least some points to be evaluated in the interpolated function will fall outside of the convex hull of the gridpoints and thus

[4] Ludwig and Schön (2014) show that Delaunay interpolation can be used with this irregular array. However, the computational gains from ENDG are partly eroded by the serial construction of the Delaunay triangulation in each period. When the number of gridpoints becomes very large, ENDG with Delaunay interpolation is actually *slower* than the traditional approach. Other methods for interpolating on an arbitrary collection of points have a similarly long construction step (e.g. polyharmonic splines, Kriging), a costly evaluation step (inverse distance weighting), or do not fit the interpolation points exactly (polynomial approximation and other regression methods); all of these methods can potentially introduce non-concavities into the value function. In contrast, non-linear grid interpolation has none of these deficiencies.

[5] Appendix A discusses adjustments for working in three dimensional (or higher) state spaces.

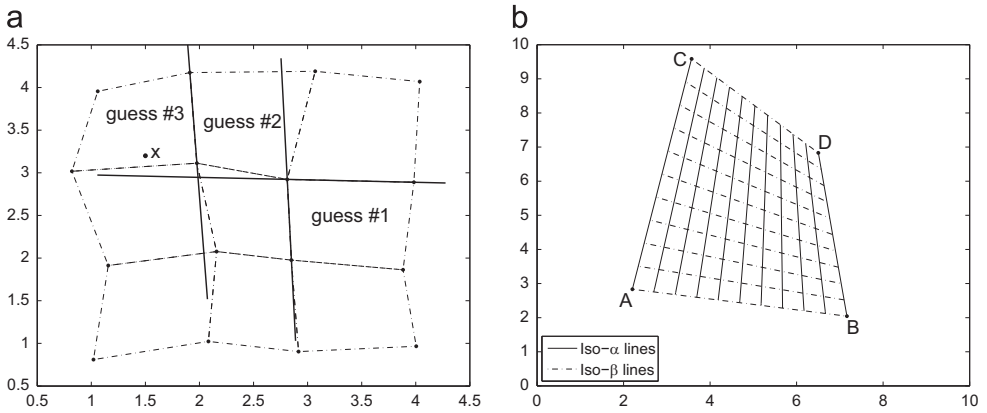

Fig. 2. Interpolation on non-linear grids. Panel (a) identifies the sector containing point x: guess #1 violates top and left boundaries, guess #2 violates left boundary, guess #3 is correct. Panel (b) shows continuous mapping of irregular quadrilateral to unit square.

not inside the bounds of any sector. In this case, the function approximation can be extrapolated using the sector from which the search process exited the convex hull.[6] So long as the highest state point in each dimension is sufficiently large that the function is approximately linear at such high values, this linear extrapolation will be sufficiently accurate.

To evaluate whether the point x violates the boundary defined by x^A and x^B, where x^B is counterclockwise from x^A, we evaluate

$$(x_2^B - x_2^A)x_1 - (x_1^B - x_1^A)x_2 \leq x_1^A x_2^B - x_2^A x_1^B. \tag{8}$$

If this inequality is violated, then x is on the outside of this sector boundary. Referencing the sector in the right panel of Fig. 2, the bounding conditions defined by x^A to x^B ("bottom"), x^B to x^D ("right"), x^D to x^C ("top"), and x^C to x^A ("left") must each be checked to ensure that x is inside the sector, and the indices of the next sector guess are incremented by ± 1 (or zero) depending on the boundary violations.

Note that this process assumes that the endogenous gridpoints that compose the irregular grid are ordered as expected (given their ordering in the fixed grid of post-decision states), allowing the sectors to be easily defined with no additional computations. This assumption is valid if there is a monotonic relationship between pre-decision states and post-decision states when the optimal policy function is obeyed. That is, increasing a pre-decision state cannot cause a decrease in the post-decision state reached after the optimal action is taken.[7] In the context of the benchmark model, a violation of the assumption would imply that beginning a period with a higher amount of money is associated with lower end of period assets (i.e. a marginal propensity to spend greater than one), or that beginning a period with lower health will result in greater health at the end of the period (so that higher health investment overcomes the lower initial health) when optimal levels of c_t and i_t are chosen.

3.3. Calculating relative coordinates within a sector

Once the sector containing the point x has been identified, we must calculate the relative coordinates (α, β) within the sector so that bilinear interpolation can be applied. To visualize this mapping, consider the right panel of Fig. 2. The four vertices form an irregular convex quadrilateral, where the α dimension is defined along the "bottom" (x^A to x^B) and "top" (x^C to x^D) edges and the β dimension along the "left" (x^A to x^C) and "right" (x^B to x^D) edges. To identify points where $\alpha = 0.4$, for example, we would go 40% of the way from x^A to x^B, and 40% of the way from x^C to x^D, and draw an iso-α line between those points; this is the fifth solid line counting from the left. Likewise, the iso-β line for $\beta = 0.7$ is the eighth dashed line counting from the bottom. Every point in the convex hull of the four vertices has a unique (α, β) identifier, just as in a sector of a rectilinear grid. However, each grid sector is non-linearly warped, so that a linear path through the sector almost always corresponds to a non-linear path in (α, β) space.

In a two dimensional state space, (α, β) can be found algebraically from (x_1, x_2). The translation from the unit square to the irregular quadrilateral sector is described by two bilinear equations, one for each dimension:

$$x_1 = (1-\alpha)(1-\beta)x_1^A + \alpha(1-\beta)x_1^B + (1-\alpha)\beta x_1^C + \alpha\beta x_1^D, \tag{9}$$

$$x_2 = (1-\alpha)(1-\beta)x_2^A + \alpha(1-\beta)x_2^B + (1-\alpha)\beta x_2^C + \alpha\beta x_2^D.$$

[6] Ideally, the visibility walk should be prevented from exiting the gridpoints unless it is the only valid move, i.e. all boundaries violated at the current guess would make the next guess out of bounds.

[7] This assumption is formalized as Condition 4 in Section 5.2.

These two equations can be rearranged and relabeled as

$$x_1 = x_1^A + (x_1^B - x_1^A)\alpha + (x_1^C - x_1^A)\beta + (x_1^A - x_1^B - x_1^C + x_1^D)\alpha\beta \equiv a + b\alpha + c\beta + d\alpha\beta, \tag{10}$$

$$x_2 = x_2^A + (x_2^B - x_2^A)\alpha + (x_2^C - x_2^A)\beta + (x_2^A - x_2^B - x_2^C + x_2^D)\alpha\beta \equiv e + f\alpha + g\beta + h\alpha\beta.$$

These expressions can likewise be manipulated to yield an algebraic solution for α and β:

$$\mu = \frac{hb - df}{dg - hc}, \quad \tau = \frac{h(a - x_1) - d(e - x_2)}{dg - hc}.$$
$$\zeta = a - x_1 + c\tau, \quad \eta = b + c\mu + d\tau, \quad \theta = d\tau. \tag{11}$$
$$\alpha = (-\eta \pm \sqrt{\eta^2 - 4\theta\zeta})/(2\theta), \quad \beta = \mu\alpha + \tau.$$

Once the correct sector for a point has been identified and the (α, β) values calculated, the function can be approximated at that point using bilinear interpolation, as in (7).

The quadratic form of the algebraic solution means that two (α, β) pairs seemingly solve the system in (9); however, only one of these pairs is the true solution, as the other will be well outside the unit square. For any given sector, either the "plus" or "minus" solution (from the $\pm$) will be correct for *every* point in that sector, but there does not seem to be a way to know which one ex ante. Instead, the researcher must check the "polarity" of the solution by calculating (α, β) for a point known to be inside the sector using the "plus" solution, evaluating whether it is in the unit square.

As with the sector identification process in Section 3.2, the validity of the relative coordinates depends on the solution to the model being "weakly monotone", so that increasing a pre-decision state never causes the resulting post-decision state to decrease. Moreover, the existence of these coordinates also requires that every sector adheres to a weak form of convexity: no vertex of a sector may be inside the convex hull of all *other* vertices of that sector. In a two dimensional state space, this condition aligns exactly with each sector being a convex quadrilateral, but is weaker than convexity in higher dimensional models, where the sectors are almost surely not convex. Violations of this condition have not yet been found,[8] but could theoretically occur if the behavioral functions rapidly flip from being convex to concave (or vice versa) and the grid is insufficiently dense.

4. Speed and accuracy of non-linear grid interpolation

This section presents a comparison of the speed and accuracy of ENDG with non-linear grid interpolation versus traditional EXOG with ordinary bilinear interpolation, using the model of Section 2 as the benchmark.

4.1. Calibration and methods

All parameters are calibrated identically to those of Ludwig and Schön; these are provided in Table 1. As the model here incorporates uncertainty in the wage and health capital depreciation rates (including the possibility of unemployment), additional parameters governing these distributions are also presented. In order to ensure an interior solution in i we must choose a coefficient of relative risk aversion $\rho < 1$.[9]

The analytic solution using ENDG is fully described in Section 2.2; for each gridpoint, solution by the traditional "method of exogenous gridpoints" (EXOG) uses Newton's method to iteratively converge to an action that sufficiently satisfies the first order conditions in (2).[10] The model is solved at a variety of grid sizes (in $m_t \times h_t$ for EXOG and $a_t \times H_t$ for ENDG), with the grid values in each dimension on roughly $[0, 300]$.[11]

At each grid size, two versions of the model are solved using EXOG and ENDG. First, I solve the model with the wage and depreciation shocks turned off ($\sigma_\omega = \sigma_\delta = 0$, but with unemployment), allowing nearly direct comparison with the results of Ludwig and Schön using Delaunay interpolation. I then solve the full model with wage and depreciation shocks, approximating each distribution with a seven point discretization for a total of $(7+1) \cdot 7 = 56$ discrete shocks.[12] This version more fully illuminates the benefits of ENDG with non-linear grid interpolation, as most models of interest will incorporate interperiod transition risk.

[8] Should the "weak monotonicity" condition be violated and the relative position of gridpoints cannot be known ex ante, then Delaunay interpolation is a viable alternative for constructing the sector map.

[9] If the utility of consumption is negative, agents will optimally choose $i_t = 0$ so as not to prolong life.

[10] The Jacobian of the first order conditions is computed numerically. The search terminates when successive iterations differ by less than $10^{-6} m_t$.

[11] Grids are specified with double log spacing, but similar results are reached with triple log spacing. The minimum non-zero grid value is 0.1 for EXOG and 0.001 for ENDG. The money or asset dimension is augmented with an additional gridpoint at $a_t = m_t = 0$, but no search is conducted for these points.

[12] Experience reveals that this is a sufficient approximation. Using a finer shock grid or more complex numeric integration technique would only make the timing results below even more striking.

Table 1
Parameters used in timing and accuracy exercises.

Parameter	Description	Value
T	Number of non-terminal periods	99
ρ	Coefficient of relative risk aversion	0.5
α	Curvature of health production function	0.35
γ	Magnitude of health production function	1
ϕ	Maximum death probability (as $h_{t+1}\to 0$)	0.5
β	Intertemporal discount factor	0.9615
$\overline{\omega}$	Average wage rate	0.1
$\overline{\delta}$	Average capital depreciation rate	0.05
R	Gross interest factor	1.05
$\mho$	Unemployment rate	0.07
σ_ω	Standard deviation of wage shocks	0.1
σ_δ	Width of depreciation risk band	0.05

The model is solved on a desktop computer with a consumer grade processor (Intel Core i7-3770). The solution code (available upon request) is written primarily in OpenCL, with Matlab used to define the parameters and pass arrays to the solver. Most solution methods are single threaded, using only one core on the CPU and thus a fraction of its power. OpenCL is a relatively new language that allows heterogeneous computing: the source code can be compiled and run on graphics processing units (GPUs) as well as CPUs; it is specifically designed for parallel processing to take advantage of the hardware on a GPU. This allows multiple parameter sets to be solved simultaneously through a multi-threaded process; in practice, this might represent preference or other exogenous heterogeneity across agents, an increasingly standard feature of applied models. To demonstrate the advantages of OpenCL, I present both single-threaded and multi-threaded timings for each solution method.

To evaluate the accuracy of the solution at each grid size and solution method, I calculate the normalized Euler errors (in the style of Judd, 1992 or Santos, 2000) for consumption and investment in each time period for 100 simulated individuals,[13] who are initialized with (m_0, h_0) states evenly spaced on $[10, 100]\times[50, 100]$. By comparing the action actually taken in period t to the optimal action implied by the first order and envelope conditions (based on the distribution of subsequent actions in period $t+1$) the errors for consumption and investment can be calculated as

$$e_{1,t} = c_t - \left(\beta R \int s(h_{t+1}) c_{t+1}^{-\rho} dF(\omega,\delta)\right)^{-1/\rho}, \tag{12}$$

$$e_{2,t} = i_t - \left(\frac{R\int s(h_{t+1}) c_{t+1}^{-\rho} dF(\omega,\delta)}{\gamma\int(1-\delta)\left(s'(h_{t+1})V_{t+1}(m_{t+1},h_{t+1}) + s(h_{t+1})c_{t+1}^{-\rho}(\omega + i_{t+1}^{1-\alpha}/\gamma)\right)dF(\omega,\delta)}\right)^{1/(\alpha-1)}.$$

These errors are then normalized and converted into "digits of accuracy" by

$$\widetilde{e}_{1,t} = -\log_{10}(|e_{1,t}/c_t|), \quad \widetilde{e}_{2,t} = -\log_{10}(|e_{2,t}/i_t|). \tag{13}$$

A normalized Euler error of $\widetilde{e}_{1,t} = 4$ indicates that the individual makes a \$1 error for every \$10,000 of consumption. Averaged across all individuals and over all non-terminal simulated periods, this yields average digits of accuracy for the behavioral functions; I also report the average of the 0.1% worst Euler errors as a measure of the lower tail of accuracy. So as not to conflate error from numeric integrals (i.e. discrete approximations to the shock distributions) with error from the solution method itself, I report these accuracy measures only for the version of the model with no wage or depreciation shocks, but with unemployment.

[13] Simulated individuals are assumed to survive from period to period.

4.2. Results

Table 2 presents timing results for the model with wage and depreciation rate shocks turned off (but with unemployment) for a variety of grid sizes.[14] For any plausible number of gridpoints, the method of endogenous gridpoints with non-linear grid interpolation is 4.3 to 5.4 times faster than the traditional exogenous gridpoints method. This is a marked improvement over the speedup achieved by Ludwig and Schön for nearly the same model using ENDG with Delaunay triangulation; they report a speedup of about 2.4 times for relatively sparse grids (25×25 to 50×50), but these gains dwindle to less than a 10% reduction in solution time for a very dense 300×300 state grid.

The key difference between the performance of the interpolation techniques is that Delaunay triangulation requires a lengthy, serial construction step that non-linear grid interpolation does not. With J total gridpoints, triangulation construction time is $O(J \log J)$, explaining the decreasing benefit of ENDG relative to EXOG in their paper. This erosion barely occurs with non-linear grid interpolation, shown graphically in the left panel of Fig. 3: solution time with EXOG is linear in the number of gridpoints and almost linear with ENDG. The computational benefits of the method of endogenous gridpoints in multiple dimensions are thus not limited to small grid sizes, but are roughly constant in the size of the problem. This is particularly useful for models with three or four endogenous dimensions, where even relatively sparse grids multiply to very large numbers of gridpoints.

A skeptical reader might question the stark difference in timing between the two non-standard interpolation techniques. That is, does ENDG with non-linear grid interpolation only achieve such a relative speed gain because the traditional EXOG method was poorly coded? The answer is no. Note that the timings for EXOG in Table 2 are just less than twice those of Ludwig and Schön for each corresponding grid size; this is precisely the result that should be achieved. The "no shocks" version of the model is identical to theirs but for the possibility of unemployment, so that two possible future states must be considered for each guess of behavior at each gridpoint, compared to only one future state in their version. The amount of computation necessary is thus roughly twice as large, and the solution time correspondingly twice as long. Even using slightly different optimization methods and possibly different hardware, our traditional EXOG solutions yield congruent timings.

In the version of the model with non-trivial interperiod risk (through wage and health capital depreciation shocks), the computational acceleration of ENDG relative to EXOG is even more dramatic, as presented in Table 3. With risk, ENDG is 7.0 to 7.8 times faster than traditional EXOG, a greater improvement than in the "no shock" model (shown graphically in the right panel of Fig. 3). The additional speed gain with risk can be explained by visibility walks that visit fewer sectors on average. That is, the sector for one discrete shock is fairly likely to be the same as the discrete shock previously considered (or at most one step away); this very good initial sector guess leads to a very short search in most cases. In the model without risk, however, the preceding correct sector corresponds to an entirely different post-decision state half of the time and is thus a worse initial guess. In this way, solving the model with shocks takes only 13–15 times longer than without shocks, even though 28 times more future states are visited.[15]

The right hand panel of Tables 2 and 3 present alternative timings for both versions of the model when the multi-threading capabilities of OpenCL are used. The single threaded timings shown in the left hand panels use only one core on an eight-core CPU, whereas the multi-threaded version uses all eight cores to solve eight parameter sets simultaneously, as if the model included preference heterogeneity among individuals; the timings presented here are *per parameter set*. While this could theoretically result in an eight-fold speed gain over single threading, the acceleration is actually only by a factor of five to six; this might be due to memory limitations as multiple threads try to access the same information. Both with and without wage and depreciation shocks, the relative speedup of ENDG over EXOG is very similar in the multi-threaded version at each gridsize.[16]

The accuracy of EXOG and ENDG at a variety of grid sizes is presented in Table 4. The numbers in the left hand panel should be interpreted as the average digits of accuracy experienced along the simulated paths described in Section 4.1, for consumption and investment respectively. For any number of gridpoints, the method of endogenous gridpoints is slightly more accurate in each control variable, and this accuracy increases as the grid of states becomes finer in each dimension. The same pattern holds in the right hand panel, presenting a "worst case scenario" of accuracy through the lower tail of normalized Euler errors. Further validating the comparison between methods, the accuracy measures are nearly identical to those of Ludwig and Schön (though their implementation of EXOG does not suffer from the slight accuracy loss shown here). The tradeoff between computational speed and accuracy of solution is represented graphically in Fig. 4, plotting average normalized Euler error against log solution time. The average slope of these lines reveals that attaining an extra digit of accuracy costs about ten times more computation time (slope of 2.3 in log terms) for EXOG and sixteen times more for ENDG

[14] For any grid size, there is approximately 0.3–0.4 s of setup time to define grids and load the OpenCL kernel, after which the model can be solved an arbitrary number of times at different parameters, as in an estimation. The times presented below thus represent "marginal time cost" per parameter set.

[15] Ludwig and Schön do not present a version with interperiod risk, but presumably Delaunay triangulation would show an even greater speedup (relative to their own timings without risk) because the lengthy construction step would be a smaller portion of total computation time.

[16] Even greater computational gains are achieved by solving the model on a GPU rather than a CPU, so that multiple gridpoints are solved simultaneously within each parameter set: solution times are about 10–12 times faster on a mid-grade consumer GPU (Radeon R9 280X) than in the multi-threaded times shown here. This is an additional benefit of non-linear grid interpolation over Delaunay triangulation: the serial construction step is ill-suited for the massively parallel architecture of GPUs, limiting the acceleration.

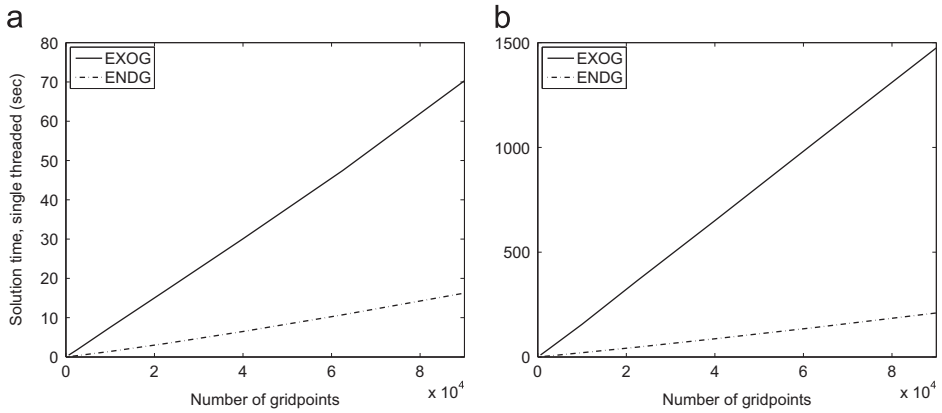

Fig. 3. Solution time of model by number of gridpoints, EXOG with ordinary bilinear interpolation vs ENDG with non-linear grid interpolation. Panel (a) shows timings for model with risk only through unemployment and mortality; panel (b) is for model with wage and health depreciation risk.

Table 2
Timing of solution methods, no shocks (seconds, speed relative to EXOG).

Number of	Single threaded		Multi-threaded	
grid points	EXOG	ENDG	EXOG	ENDG
25×25	(0.445, 1×)	(0.083, 5.4×)	(0.084, 1×)	(0.019, 4.4×)
50×50	(1.803, 1×)	(0.336, 5.4×)	(0.330, 1×)	(0.066, 5.0×)
100×100	(7.537, 1×)	(1.418, 5.3×)	(1.338, 1×)	(0.258, 5.2×)
150×150	(16.90, 1×)	(3.417, 4.9×)	(3.012, 1×)	(0.623, 4.8×)
200×200	(30.08, 1×)	(6.480, 4.6×)	(5.316, 1×)	(1.203, 4.4×)
250×250	(47.45, 1×)	(10.73, 4.4×)	(8.473, 1×)	(1.960, 4.3×)
300×300	(70.29, 1×)	(16.26, 4.3×)	(12.38, 1×)	(2.976, 4.2×)

Table 3
Timing of solution methods, with shocks (seconds, speed relative to EXOG).

Number of	Single threaded		Multi-threaded	
grid points	EXOG	ENDG	EXOG	ENDG
25×25	(9.336, 1×)	(1.216, 7.7×)	(1.765, 1×)	(0.248, 7.1×)
50×50	(38.24, 1×)	(4.893, 7.8×)	(7.099, 1×)	(0.994, 7.1×)
100×100	(156.2, 1×)	(20.27, 7.7×)	(28.90, 1×)	(3.761, 7.7×)
150×150	(364.8, 1×)	(47.11, 7.7×)	(65.70, 1×)	(8.723, 7.5×)
200×200	(650.6, 1×)	(86.88, 7.5×)	(118.1, 1×)	(15.91, 7.4×)
250×250	(1023, 1×)	(140.4, 7.3×)	(183.9, 1×)	(25.92, 7.1×)
300×300	(1475, 1×)	(210.1, 7.0×)	(266.8, 1×)	(38.29, 7.0×)

Table 4
Accuracy of solution methods, no shocks (negative $\log_{10}$ Euler error).

Number of	Overall average (c, i)		Average of 0.1% worst (c, i)	
grid points	EXOG	ENDG	EXOG	ENDG
25×25	(3.48, 2.45)	(3.87, 2.79)	(1.81, 1.74)	(2.26, 1.80)
50×50	(4.07, 3.11)	(4.26, 3.27)	(2.35, 2.32)	(3.11, 2.53)
100×100	(4.65, 3.65)	(4.90, 3.87)	(2.88, 2.80)	(3.47, 2.97)
150×150	(5.00, 3.97)	(5.17, 4.18)	(3.26, 3.04)	(3.60, 3.14)
200×200	(5.21, 4.18)	(5.41, 4.39)	(3.41, 3.23)	(3.95, 3.44)
250×250	(5.36, 4.35)	(5.55, 4.57)	(3.65, 3.37)	(3.86, 3.43)
300×300	(5.50, 4.48)	(5.66, 4.69)	(3.77, 3.51)	(4.12, 3.62)

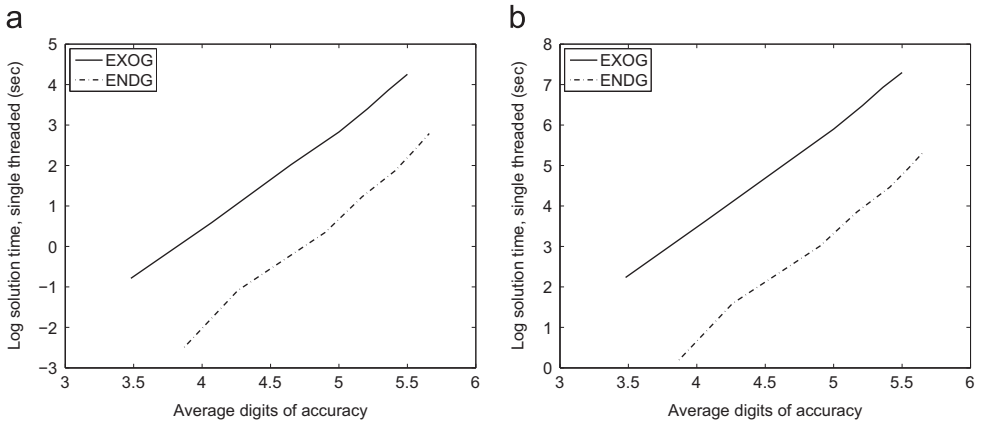

Fig. 4. Digits of accuracy achieved by EXOG and ENDG vs solution time. Panel (a) shows timings for model with risk only through unemployment and mortality; panel (b) is for model with wage and health depreciation risk (accuracy based on no shocks model).

(12- vs 15-times longer in the model with shocks). Between being slightly more accurate and considerably faster for any given number of gridpoints, the advantage of ENDG over EXOG can be alternatively phrased purely in terms of accuracy: for any given computational time, ENDG yields 0.8 to 1.2 more digits of accuracy than EXOG.

5. A generalized method of endogenous gridpoints

The benchmark model of Section 2 was chosen because it can be solved with the method of endogenous gridpoints, a property not shared by all dynamic models.[17] For example, ENDG cannot be used when an agent allocates resources between a risky and riskless asset, nor when he spends resources to obtain a more informative signal about an unknown quantity. To better understand the circumstances in which the method of endogenous gridpoints can be used, this section formalizes the method in a theoretical framework for dynamic stochastic optimization problems. I present sufficient conditions to employ ENDG (and non-linear grid interpolation) and then discuss the applicability of the method.

5.1. Generic dynamic optimization problem

This subsection describes a general intertemporal optimization problem in which an agent seeks to maximize the stream of utility he receives over the course of his lifetime by taking an action in each discrete time period. The individual's action is constrained depending on his current state in each period, and the transition to next period's state depends stochastically on both the current state and the choice of action.

Time t begins at period 0 and ends after period $T \in \mathbb{N}$.[18] At the beginning of period t, the individual finds himself in a state (w_t, x_t), conceptually split between the *exogenous* state $w_t \in W_t \subseteq \mathbb{R}^p$ and the *endogenous* state $x_t \in X_t \subseteq \mathbb{R}^n$. This distinction is only relevant for the individual's transition into the next period (described below): his action in this period will affect endogenous state x_{t+1} but not w_{t+1}.

The individual must make a choice of action y_t from the closed and convex set of allowable choices $\Gamma_t(w_t, x_t)$, where the correspondence $\Gamma_t(\cdot)$ represents the constraint or budget set for the current state: $\Gamma_t: W_t \times X_t \rightrightarrows Y_t$. The control set $Y_t \subseteq \mathbb{R}^m$ represents the overall set of actions that are feasible from at least one state: $Y_t = \cup_{(w_t,x_t) \in W_t \times X_t} \Gamma(w_t, x_t)$. Assume that the constraint correspondence $\Gamma_t(\cdot)$ is continuous and that the budget set for all (w_t, x_t) is compact. The individual derives a flow of (possibly state-dependent) utility from his action according to the utility function $U_t: W_t \times X_t \times Y_t \to \mathbb{R}$. To ensure that the optimal policy correspondence is single-valued, assume that $U_t(\cdot)$ is continuous, strictly concave, and differentiable on the interior of its domain. At time t, the agent discounts future utility flows at a factor of $\beta_t \in [0, 1]$.[19]

After making his choice, the agent's state moves from (w_t, x_t) to the next period's state $(w_{t+1}, x_{t+1}) \in W_{t+1} \times X_{t+1}$ according to the continuous and differentiable transition function $\Delta_t(\cdot)$. The transition is risky and depends on random shock $s_{t+1} \in S_{t+1} \subseteq \mathbb{R}^\ell$, drawn from a distribution with CDF $F_{t+1}(s)$ and unknown to the agent when the action is chosen: $\Delta_t: W_t \times X_t \times Y_t \times S_{t+1} \to W_{t+1} \times X_{t+1}$. Denote the portion of $\Delta_t(\cdot)$ that yields the exogenous state as $\Delta_t^w(\cdot)$ and the portion that yields the endogenous state $\Delta_t^x(\cdot)$. As suggested above, the distinction between exogenous and endogenous state dimensions is determined by whether the action or endogenous state affects the transition to the next period: $\partial \Delta_t^w / \partial y = 0_{p \times m}$ and $\partial \Delta_t^w / \partial x = 0_{p \times n}$, while $\partial \Delta_t^x / \partial y$ has no rows composed entirely of zeros. The exogenous portion of the transition will be expressed as $\Delta_t^w(w_t, s_{t+1})$, omitting the irrelevant x_t and y_t arguments.

[17] Indeed, even small variations on the benchmark model are not compatible with ENDG; see Section 5.4.
[18] This framework can also be applied to infinite horizon models, with the time subscripts suppressed.
[19] Standard topological assumptions are omitted for brevity; see Stokey and Lucas (1989, chapter 9).

The agent is an expected utility maximizer, so his problem at time t can be written as

$$\max_{y_t \in \Gamma_t(w_t,x_t)} U_t(w_t,x_t,y_t)+E_t\left[\sum_{\tau=t+1}^{T}\left(\prod_{q=t}^{\tau-1}\beta_q\right)U_\tau(w_\tau,x_\tau,y_\tau)\right] \tag{14}$$

$$\text{s.t. } (w_{\tau+1},x_{\tau+1})=\Delta_\tau(w_\tau,x_\tau,y_\tau,s_{\tau+1}) \quad \text{and} \quad s_{\tau+1}\sim F_{\tau+1}(s).$$

Problems of this type can be rewritten using value functions, as in Bellman (1957). The value function at time t, $V_t: W_t \times X_t \to \mathbb{R}$, is recursively defined by

$$V_t(w_t,x_t)=\max_{y_t \in \Gamma_t(w_t,x_t)} U_t(w_t,x_t,y_t)+\beta_t E_t\left[V_{t+1}(w_{t+1},x_{t+1})\right] \tag{15}$$

$$\text{s.t. } (w_{t+1},x_{t+1})=\Delta_t(w_t,x_t,y_t,s_{t+1}) \quad \text{and} \quad s_{t+1}\sim F_{t+1}(s).$$

The expectation here is taken over the possible transition shocks s_{t+1} experienced between periods. Folding the constraints into the maximand, the Bellman form can be rewritten as

$$V_t(w_t,x_t)=\max_{y_t \in \Gamma_t(w_t,x_t)} U_t(w_t,x_t,y_t)+\beta_t \int V_{t+1}(\Delta_t(w_t,x_t,y_t,s))\, dF_{t+1}(s). \tag{16}$$

In the terminal period, the value function is defined as simply:

$$V_T(w_T,x_T)=\max_{y_T \in \Gamma_T(w_T,x_T)} U_T(w_T,x_T,y_T). \tag{17}$$

While the value function is the maximum of the agent's choice problem, define the optimal policy function $\psi_t: W_t \times X_t \to Y_t$ as the maximizer of this problem (i.e. replace max with arg max in (16) and (17)).

The first order conditions for optimality are found by taking the derivative of (16) with respect to the action:

$$\frac{\partial U_t(w_t,x_t,y_t)}{\partial y}+\beta_t \int \left(\frac{\partial V_{t+1}(\Delta_t^w(w_t,s),\Delta_t^x(w_t,x_t,y_t,s))}{\partial x}\cdot\frac{\partial \Delta_t^x(w_t,x_t,y_t,s)}{\partial y}\right)dF_{t+1}(s)=0_{1\times m}. \tag{18}$$

With the problem written in this form, it can be iteratively solved backwards from the final period T. In all non-terminal periods $t\in\{0,\ldots,T-1\}$, the agent uses the value function (and its first derivatives) for the following period to solve (16). However, because $V_{t+1}(\cdot)$ is almost certainly non-analytic, the problem in (18) has no closed form solution that can be found algebraically. Instead, finding a sufficient solution to the agent's problem at a particular state space point (w_t^j,x_t^j) requires several "guesses" of $y_t^j=\psi_t(w_t^j,x_t^j)$ to be considered, with each guess requiring a costly numeric integral of $V_{t+1}(\cdot)$ (and/or its derivatives) over the possible future states that may attain.

The method of endogenous gridpoints provides an alternative approach to solving dynamic optimization problems that requires numeric integration only *once* per gridpoint, greatly accelerating the solution. Rather than exogenously choosing the state space gridpoints points over which to interpolate the policy and value functions, the gridpoints are solved *endogenously* and simultaneously with the optimal actions at those states.

5.2. Conditions for using endogenous gridpoints

The method of endogenous gridpoints can only be employed on a subset of the general class of problems described above. This subsection discusses three conditions that must be met in order to employ ENDG, as well as additional objects (defined from the problem primitives) that are used in the method.

First, the researcher defines $Z_t\subset\mathbb{R}^n$ as the set of "post-decision endogenous states" with typical element z_t. These represent the intermediate state after the agent has chosen and executed his action y_t but before the transition shock s_{t+1} is revealed. This leads directly to the most important condition for employing ENDG: a decomposition of the endogenous state transition function $\Delta_t^x(\cdot)$ into intra- and inter-period components:

Condition 1. There exist functions $\Xi_t: W_t\times X_t\times Y_t\to\mathbb{R}^n$ and $\chi_t: W_t\times Z_t\times S_{t+1}\to X_{t+1}$ such that $\Delta_t^x(w_t,x_t,y_t,s_{t+1})=\chi_t(w_t,\Xi_t(w_t,x_t,y_t),s_{t+1})$ for all $(w_t,x_t,y_t)\in W_t\times X_t\times Y_t$ when $y_t\in\Gamma_t(w_t,x_t)$.

This says that the transition into next period's state does not depend directly on the current state and action, but only on some deterministic function of the two. That is, the combination of a state (w_t,x_t) and control choice y_t generates a post-decision state z_t (via the "intraperiod transition function" $\Xi_t(\cdot)$) that serves as a sufficient statistic for both the endogenous state and control for the purposes of the "interperiod transition function" $\chi_t(\cdot)$. The evolution of the exogenous state $w_{t+1}=\Delta_t^w(w_t,s_{t+1})$ is unaffected. Note that if such a decomposition exists, then $\Xi_t(\cdot)$ and $\chi_t(\cdot)$ are continuous and differentiable.

The constraint correspondence $\Gamma_t(w_t,x_t)$ is not explicitly used in the method of endogenous gridpoints, but instead "absorbed" into the post-decision state. That is, we require that the post-decision state (w_t,z_t) is also a sufficient statistic for evaluating the feasibility of the action y_t the individual took to arrive at it. Combined with a requirement that ensures that the post-decision space spans the entire realm of possible outcomes, this yields a key condition on the nature of the post-decision state space:

Condition 2. For all $y_t \in Y_t$ and $(w_t, x_t) \in W_t \times X_t$, $y_t \in \Gamma_t(w_t, x_t)$ if and only if $\Xi_t(w_t, x_t, y_t) \in Z_t$. Moreover, $Z_t = \cup_{(w_t,x_t) \in W_t \times X_t} \Xi_t(w_t, x_t, \Gamma(w_t, x_t))$.

Consider Condition 2 in the context of a traditional consumption-saving model, so that the pre-decision state is the money resources available to the agent when he makes his decision, while the post-decision state represents his remaining assets after he has consumed but before interperiod risk (e.g. over future income or the interest rate) has been resolved. The constraint that the individual cannot consume more than his resources is entirely captured by the post-decision state: if the agent ends the period with non-negative assets, then whatever (non-negative) level of consumption he chose must have obeyed the budget constraint.

With the change of variables possible if Condition 1 is satisfied, Eq. (18) can be rewritten as

$$\frac{\partial U_t(w_t, x_t, y_t)}{\partial y} = -\beta_t \int \frac{\partial V_{t+1}(\overbrace{\Delta_t^w(w_t, s), \chi_t(w_t, z_t, s)}^{w_{t+1}, x_{t+1}})}{\partial x} \cdot \underbrace{\frac{\partial \chi_t(w_t, z_t, s)}{\partial z} \cdot \frac{\partial \Xi_t(w_t, x_t, y_t)}{\partial y}}_{\frac{\partial \Delta_t^x}{\partial y}} dF_{t+1}(s)$$

$$\text{s.t. } \Xi_t(w_t, x_t, y_t) = z_t. \tag{19}$$

Because $\Xi_t(\cdot)$ does not depend on the random shock s, it can be taken out of the integral as

$$\frac{\partial U_t(w_t, x_t, y_t)}{\partial y} = -\beta_t \int \frac{\partial V_{t+1}(\Delta_t^w(w_t, s), \chi_t(w_t, z_t, s))}{\partial x} \cdot \frac{\partial \chi_t(w_t, z_t, s)}{\partial z} dF_{t+1}(s) \cdot \frac{\partial \Xi_t(w_t, x_t, y_t)}{\partial y}. \tag{20}$$

The right hand side of this equation depends on the choice of action y_t only through the rightmost factor $\partial \Xi_t / \partial y$. Moving this factor to the LHS – so as to isolate the dependency on y_t – requires that it be invertible in some sense. To this end, it is useful to differentiate controls for which $\partial \Xi_t / \partial y$ does not depend on y_t or x_t. In a consumption-saving model, for example, the choice of consumption c affects end-of-period assets a linearly and thus $\partial a / \partial c = -1$ and does not depend on any control or state. Denote a subset of these "linear controls" as $\widehat{y}$ and the remaining controls as $\widetilde{y}$. This leads to a general "invertibility" condition:

Condition 3. There are $m-n$ controls $\widehat{y}$ such that $\partial \Xi_t / \partial \widehat{y}$ is constant with respect to y_t and x_t. Moreover, the complementary set of n controls $\widetilde{y}$ is such that $\partial \Xi_t / \partial \widetilde{y}$ has rank n.

The invertibility condition is trivially met when $m = n$ and $\partial \Xi_t / \partial y$ is of rank m. When $m > n$ so that there are more controls than endogenous state dimensions, ENDG requires that *at least* $m-n$ of the controls have only a linear effect on the post-decision state. When there are more than $m-n$ "linear controls", the researcher may choose among them to define $\widehat{y}$ as is mathematically convenient. If $n > m$, the method of endogenous gridpoints cannot necessarily be employed. Although the $\partial \Xi_t / \partial y$ factor can be moved to the LHS because its right-inverse exists (when $\partial \Xi_t / \partial y$ has rank m), the resulting system of equations is overdetermined and there may be no solution in (x_t, y_t) for a particular (w_t, z_t).

Once the researcher has divided the controls into $\widehat{y}$ and $\widetilde{y}$, define four new functions:

$$\widehat{g}_t(w_t, x_t, y_t) = \frac{\partial U_t(w_t, x_t, y_t)}{\partial \widehat{y}}, \quad \widetilde{g}_t(w_t, x_t, y_t) = -\frac{\partial U_t(w_t, x_t, y_t)}{\partial \widetilde{y}} \left(\frac{\partial \Xi_t(w_t, x_t, y_t)}{\partial \widetilde{y}} \right)^{-1}, \tag{21}$$

$$\widehat{V'}_{t+1}(w_t, z_t) = -\int \frac{\partial V_{t+1}(\Delta_t^w(w_t, s), \chi_t(w_t, z_t, s))}{\partial x} \cdot \frac{\partial \chi_t(w_t, z_t, s)}{\partial z} dF_{t+1}(s) \cdot \frac{\partial \Xi_t(w_t, x_t, y_t)}{\partial \widehat{y}}, \tag{22}$$

$$\widetilde{V'}_{t+1}(w_t, z_t) = \int \frac{\partial V_{t+1}(\Delta_t^w(w_t, s), \chi_t(w_t, z_t, s))}{\partial x} \cdot \frac{\partial \chi_t(w_t, z_t, s)}{\partial z} dF_{t+1}(s).$$

Although x_t and y_t appear in the definition of $\widehat{V'}_{t+1}(w_t, z_t)$, by the construction of $\widehat{y}$ there is no actual dependence on the endogenous state or control variables. The functions in (21) and (22) can be combined to form the objects of interest:

$$g_t(w_t, x_t, y_t) = (\widehat{g}_t(w_t, x_t, y_t), \widetilde{g}_t(w_t, x_t, y_t)), \quad \overline{V'}_{t+1}(w_t, z_t) = (\widehat{V'}_{t+1}(w_t, z_t), \widetilde{V'}_{t+1}(w_t, z_t)). \tag{23}$$

By construction, ENDG can be used to solve a dynamic stochastic optimization problem that satisfies Conditions 1–3. Moreover, the policy and value functions can be approximated with non-linear grid interpolation on the endogenous state space points when the "weak monotonicity" condition is met:

Condition 4. At policy solution $\psi_t(w_t, x_t)$, $\Xi_t(\cdot)$ is monotone up in x_t:

$$\frac{d\Xi_t(w_t, x_t, \psi(w_t, x_t))}{dx_t} > 0.$$

Condition 4 is not strong and should be satisfied by nearly every model of interest. Violations of the condition would be analogous to a marginal propensity to spend from wealth greater than one or the optimal values of a capital stock (e.g. health capital or a durable good) being negatively correlated across periods.

5.3. Method of endogenous gridpoints defined

The main insight of the method of endogenous gridpoints is that the original first order conditions (18), in which both sides depend on the control variables, can be rearranged into a system of equations with endogenous variable dependence on only one side:

$$g_t(w_t, x_t, y_t) = \beta_t \overline{V'}_{t+1}(w_t, z_t). \tag{24}$$

The key difference between the traditional solution method and the method of endogenous gridpoints is that ENDG fixes the mesh of gridpoints $\mathcal{J}_t$ in the post-decision state space $W_t \times Z_t$ rather than the pre-decision state space $W_t \times X_t$. From any post-decision state (w_t^j, z_t^j), ENDG asks what action must have *just been taken* in order for ending the period in this way to be optimal. That is, for each gridpoint in $\mathcal{J}_t$, the method solves for both the action y_t^j and the endogenous pre-decision state x_t^j from which it was taken. This is represented by a system of $m+n$ equations with $m+n$ unknowns:

$$\underbrace{g_t(w_t^j, x_t^j, y_t^j) = \beta_t \overline{V'}_{t+1}(w_t^j, z_t^j)}_{m \text{ equations}}, \quad \underbrace{\Xi_t(w_t^j, x_t^j, y_t^j) = z_t^j}_{n \text{ equations}}. \tag{25}$$

For any post-decision gridpoint $(w_t^j, z_t^j) \in \mathcal{J}_t$, the RHS of both parts of (25) are *constant*. In this way, ENDG transforms the optimization problem at a single gridpoint from an m-dimensional rootfinding operation of a non-analytic system with an ℓ-dimensional integral at each guess of y_t^j into an $(m+n)$-dimensional rootfinding operation of a (possibly analytic) system with the integral evaluated *exactly once*.

After defining the post-decision state space and the decomposition of the transition function (and verifying that the model meets Conditions 1–3), ENDG follows a four step procedure in each period t, recursively solving backward until $t=0$.

1. For each post-decision gridpoint $(w_t^j, z_t^j) \in \mathcal{J}_t \subset W_t \times Z_t$, compute $\overline{V'}_{t+1}(w_t^j, z_t^j)$ via the integral in (22).
2. Find the unique solution in $X_t \times Y_t$ to the system of $m+n$ equations in (25), yielding pre-decision state (w_t^j, x_t^j) and the optimal action y_t^j at that state, so that $\psi_t(w_t^j, x_t^j) = y_t^j$.
3. Calculate the level of the value function $V_t(w_t^j, x_t^j)$ at each of the just-solved state points using (16), as well as the marginal value $\partial V_t / \partial x$ via the envelope conditions.
4. Approximate the policy function and value function (and its derivatives) by interpolating over the endogenously determined state points.

5.4. Discussion

The class of models described in Section 5.1 is very broad, incorporating exogenous state variables, risky endogenous states, more controls than endogenous states, state-dependent utility, and time-varying primitives. This general structure emphasizes the wide array of models for which ENDG can be used and fits all previously published applications into the framework. Maliar and Maliar (2013) present a solution method with a parallel structure to ENDG, achieving effectively identical results in both speed and accuracy. Rather than iterating backward on the first order conditions, their method (ECM) iterates forward using the envelope conditions. While not necessarily included in their example model, all but one of the complicating features listed above are compatible with ECM; however, as the method "forward solves", it can only be used for infinite horizon, time-invariant models. In contrast, ENDG can be applied to both finite (e.g. life cycle) and infinite horizon models.

The computational gains of the method of endogenous gridpoints take two forms. First, in *every* model for which it is applicable, ENDG avoids repeated evaluations of numeric integrals when solving for the optimal action. Second, *some* applicable models have an analytic solution to the ENDG first order conditions in (25), eliminating all root-finding from the solution method; other models have a partially analytic solution that reduces the dimensionality of the root-find to less than $m+n$. The existence of an analytic solution depends on the form of the utility function and intraperiod transition function; in very recent work, Iskhakov (2015) characterizes the conditions for avoiding all root-finding on a subclass of models. To the extent that the solution is analytic, ENDG also benefits from improved accuracy relative to a search for a satisfactory solution to the first order conditions.

While the functional forms used in the benchmark model of Section 2 allow ENDG to be applied[20] and permit a fully algebraic solution, small adjustments to the model can break these results. For example, suppose that next period's health was instead given by $h_{t+1} = (1-\delta_{t+1})h_t + f(i_t)$, so that depreciation only affects prior capital. ENDG cannot be used to solve

[20] The model fits neatly into the theoretical framework presented above: the post-decision state variables $a_t = m_t - c_t - i_t$ and $H_t = h_t + f(i_t)$ define the intraperiod transition function $\Xi_t(m_t, h_t, c_t, i_t)$, and the interperiod transition $\chi_t(a_t, H_t, \omega_{t+1}, \delta_{t+1})$ is given by (3).

this model because there is no way to decompose the transition function so that the post-decision state space has exactly $n=2$ dimensions – the distribution of future states depends on h_t, i_t, and a_t independently, with no 2D sufficient statistic available.[21]

However, the applicability of ENDG would be restored if there were no depreciation rate risk, $\delta_{t+1}=\overline{\delta}$. In that case, the post-decision state for health can be taken as next period's health itself: $H_t=(1-\overline{\delta})h_t+f(i_t)=h_{t+1}$. More broadly, any model with no endogenous state risk satisfies Condition 1 with a trivial decomposition: $z_t=\Xi_t(w_t,x_t,y_t)=\Delta_t^x(w_t,x_t,y_t,s_{t+1})$ and $x_{t+1}=\chi_t(w_t,z_t,s_{t+1})=z_t$. This result demonstrates how methods that solve a model by constructing a grid on next period's state variables, including Wright and Williams (1984), are nested in the ENDG framework as a special case.

Consider instead a version of the benchmark model in which the health production function depends on both investment and current health, as in Ben-Porath (1967): $f(h_t,i_t)$. The method of endogenous gridpoints can be used to solve this model, and the algebraic solutions for c_t and m_t are unaffected. However, obtaining h_t and i_t for a post-decision gridpoint (a_t,H_t) requires solving a system of two non-linear equations of the form:

$$f^i(h_t,i_t)=K, \quad h_t+f(h_t,i_t)=H_t \quad \text{(for some constant } K\text{)}. \tag{26}$$

In this version, both ENDG and traditional EXOG require root-finding on a non-linear 2D system, but EXOG must compute numeric integrals at each iteration of the root-finding operation while ENDG computes them only once. Even when the model primitives do not generate a fully analytic solution, ENDG still offers a significant computational acceleration.

6. Conclusion

This paper presents a technique for interpolating on the irregular collection of pre-decision gridpoints generated by the method of endogenous gridpoints in multi-dimensional models. Non-linear grid interpolation maps irregular quadrilaterals into the unit square, acting as an intuitive extension of bilinear interpolation on a rectilinear grid; it extends to three or more dimensions with only minor modifications, translating non-polyhedral shapes into the unit cube. Unlike other methods for interpolating on an irregular collection of points, non-linear grid interpolation exploits known relationships among the gridpoints and involves neither a lengthy construction step nor an evaluation step. Critically, it allows the computational acceleration of ENDG to erode very slowly as the density of gridpoints increases.

Further, I generalize the method of endogenous gridpoints in a framework for dynamic stochastic optimization problems and specify conditions on the class of models for which ENDG can be applied. When these conditions are satisfied, the first order conditions can be rewritten in a way that isolates the control variables from the expectation of future marginal value. This reformulation eliminates the need for numeric integrals to be computed many times in the search for the optimal control, instead evaluating them exactly once. Moreover, the resulting system of equations often has a fully analytic solution; even when it does not, the root-finding operation will be reduced in both dimensionality and computational burden.

In combination, these theoretical and applied methods greatly expand the range of models for which the method of endogenous gridpoints can be used, representing a step toward managing (but not eliminating) the well known curse of dimensionality.

Acknowledgments

I thank editor Thomas Lubik and anonymous referees at the JEDC for their comments and suggestions that significantly improved and streamlined the exposition of this work.

Appendix A. Higher dimensional interpolation on non-linear grids

In models with three or more endogenous state dimensions,[22] both the sector search and calculation of relative coordinates must be modified from the two dimensional algorithms. While these methods are slower than their 2D counterparts, the additional computational burden is significantly less than the lengthy construction step of Delaunay interpolation in higher dimensions (nor does the required memory to store the interpolation grow exponentially with the number of dimensions) or the $O(J)$ evaluation procedure of inverse distance weighting and related methods.

In two dimensions, we can evaluate whether a point is in a sector by explicitly evaluating the four linear boundaries, as in (8). There is no analogous procedure in a three dimensional state space: the six sides of an irregular grid sector are not

[21] The same logic applies to models with portfolio allocation between a risky and riskless asset, so that ENDG cannot be used with these models at all.

[22] If a state dimension evolves exogenously of the control variables, then non-linear grid interpolation need not (and should not) be used in that dimension; linear interpolation can be used instead.

defined by planes, as the four vertices of a side do not align on a single plane. Instead, we first check whether it is plausible for x to be in the sector based on the "bounding box": the smallest rectangular prism superset of the sector. Defining $\aleph$ as the set of 2^n sector vertices, x must satisfy

$$(\min_i x_k^i \in \aleph) \le x_k \le (\max_i x_k^i \in \aleph) \text{ for } k = 1, \ldots, n. \tag{27}$$

If any of the $2n$ inequalities are violated, then x cannot be in the current sector; the indices of the next guess can be incremented based on these violations, as in Section 3.3. If all of the inequalities are satisfied, then we know that x is either in the current sector or fairly close to it. We then calculate the relative coordinates of x in the current sector using the procedure in the paragraphs below, under the assumption that it is in this sector. If that process returns coordinates in $[0,1]^n$ then the sector has been correctly identified and the search terminates. If invalid coordinates are returned, then the sector guess can be updated based on the violations; e.g. if $\beta > 1$, then the next sector guess should be incremented one to the "right". It is rare that the evaluation of one point will require calculating the relative coordinates in more than two sectors.

Calculating the relative coordinates of a point within a sector can be achieved algebraically in a two dimensional setting through judicious use of the quadratic formula; this method cannot be used in three dimensions or higher.[23] Instead, we can use Newton's method to find these coordinates. In three dimensions, the mapping from the unit cube into an irregular sector[24] consists of a triple trilinear system:

$$\begin{aligned} x_k = {} & (1-\alpha)(1-\beta)(1-\gamma)x_k^A + \alpha(1-\beta)(1-\gamma)x_k^B + (1-\alpha)\beta(1-\gamma)x_k^C + \alpha\beta(1-\gamma)x_k^D \\ & + (1-\alpha)(1-\beta)\gamma x_k^E + \alpha(1-\beta)\gamma x_k^F + (1-\alpha)\beta\gamma x_k^G + \alpha\beta\gamma x_k^H \text{ for } k = 1,2,3. \end{aligned} \tag{28}$$

Each derivative of this system is a bilinear function of the other coordinates. For example:

$$\frac{\partial x_k}{\partial \alpha} = (1-\beta)(1-\gamma)\left(x_k^B - x_k^A\right) + \beta(1-\gamma)\left(x_k^D - x_k^C\right) + (1-\beta)\gamma\left(x_k^F - x_k^E\right) + \beta\gamma\left(x_k^H - x_k^G\right). \tag{29}$$

Starting from an initial guess of $(\alpha_0, \beta_0, \gamma_0) = (0.5, 0.5, 0.5)$ and defining x_i as the point yielded when $(\alpha_i, \beta_i, \gamma_i)$ are used in (28), we can then iteratively update the guess of relative coordinates until the process converges to a given tolerance by doing:

$$\begin{bmatrix} \alpha_{i+1} \\ \beta_{i+1} \\ \gamma_{i+1} \end{bmatrix} = \begin{bmatrix} \alpha_i \\ \beta_i \\ \gamma_i \end{bmatrix} + \begin{bmatrix} \frac{\partial x_1}{\partial \alpha} & \frac{\partial x_1}{\partial \beta} & \frac{\partial x_1}{\partial \gamma} \\ \frac{\partial x_2}{\partial \alpha} & \frac{\partial x_2}{\partial \beta} & \frac{\partial x_2}{\partial \gamma} \\ \frac{\partial x_3}{\partial \alpha} & \frac{\partial x_3}{\partial \beta} & \frac{\partial x_3}{\partial \gamma} \end{bmatrix}^{-1} \begin{bmatrix} x_1 - x_1^i \\ x_2 - x_2^i \\ x_3 - x_3^i \end{bmatrix}. \tag{30}$$

Because Newton's method converges quadratically, the function implied by (28) is roughly linear, and the initial guess is reasonably close to the true relative coordinates, this procedure converges to double precision (approximately 10^{-16}) in fewer than six iterations on average.

Both the sector search method and the Newton search for relative coordinates apply in four (or more) dimensional state spaces in the same way as in 3D, though the corresponding partial derivatives in (29) require more arithmetic operations, and the inversion of the Jacobian in (30) is less trivial. Note that finding the maximal and minimal vertices in each dimension for a sector (as in (27)) need only be performed once, when the irregular grid is created. The indices of the extrema for each sector can be stored, requiring only $2(n^2 - n)$ *bits* per sector; even in a model with six endogenous dimensions, this information can be stored in a long integer. In contrast, the storage needs of Delaunay interpolation grow exponentially in the number of dimensions.

Appendix B. Derivations

Derivatives of the model primitives used in the ENDG solution (4) are

$$u'(c) = c^{-\rho}, \quad f'(i) = \gamma i^{\alpha - 1}, \quad s'(h) = \frac{\phi}{(h+1)^2}. \tag{31}$$

The first order condition for optimal consumption in (2) is straightforward to derive. Its counterpart for health investment is a rearrangement of the initial form, solved for $f'(i_t)$:

$$\beta \int Rs(h_{t+1}) V_{t+1}^m(m_{t+1}, h_{t+1}) + f'(i_t)(1-\delta)\left[s'(\cdot)V_{t+1}(\cdot) + s(\cdot)\left(\omega V_{t+1}^m(\cdot) + V_{t+1}^h(\cdot)\right)\right] dF(\omega, \delta) = 0. \tag{32}$$

[23] Given the considerable arithmetic and multitude of solutions to Ferrari's cubic and quartic formulas, it is both easier to program and faster to compute the Newton search described here.

[24] With points x^A, x^B, x^C, x^D defining the "low side" in the same orientation as in the right panel of Fig. 2, and points x^E, x^F, x^G, x^H their respective "high side" counterparts.

The marginal value functions in (5) are reached by taking the derivative of the Bellman equation in (1) while applying the "envelope principle". For money, this yields

$$V_t^m(m_t, h_t) = \beta R \int s(h_{t+1}) V_{t+1}^m(m_{t+1}, h_{t+1})\, dF(\omega, \delta) = u'(c_t). \tag{33}$$

Marginal value with respect to health can be found by substituting the ratio of the FOCs:

$$V_t^h(m_t, h_t) = \beta \int (1-\delta)\left[s'(\cdot)V_{t+1}(\cdot) + s(\cdot)\left(\omega V_{t+1}^m(\cdot) + V_{t+1}^h(\cdot)\right)\right] dF(\omega, \delta) = \frac{u'(c_t)}{f'(i_t)}. \tag{34}$$

References

Barillas, F., Fernández-Villaverde, J., 2007. A generalization of the endogenous grid method. J. Econ. Dyn. Control 31, 2698–2712.

Bellman, R.E., 1957. Dynamic Programming. Princeton University Press, Princeton, NJ.

Ben-Porath, Y., 1967. The production of human capital and the life cycle of earnings. J. Polit. Econ. 75 (4), 352–365.

Carroll, C.D., 2006. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. Econ. Lett. 91, 312–320.

Fella, G., 2014. A generalized endogenous grid method for non-concave problems. Rev. Econ. Dyn. 17 (2), 329–344.

Hintermaier, T., Koeniger, W., 2010. The method of endogenous gridpoints with occasionally binding constraints among endogenous variables. J. Econ. Dyn. Control 34, 2074–2088.

Iskhakov, F., 2015. Multidimensional endogenous gridpoint method: solving triangular dynamic stochastic optimization problems without root-finding operations. Econ. Lett., forthcoming.

Iskhakov, F., Rust, J., Schjerning, B., 2012. A generalized endogenous grid method for discrete-continuous choice. Working paper.

Judd, K.L., 1992. Projection methods for solving aggregate growth models. J. Econ. Theory 58 (2), 410–452.

Ludwig, A., Schön, M., 2014. Endogenous grids in higher dimensions: Delaunay interpolation and hybrid methods. Working Paper.

Maliar, L., Maliar, S., 2013. Envelope condition method vs endogenous grid method for solving dynamic programming problems. Econ. Lett. 120, 262–266.

Santos, M.S., 2000. Accuracy of numerical solutions using the euler equation residuals. Econometrica 68 (6), 1377–1402.

Stokey, N.L., Lucas, R.E., 1989. Recursive Methods in Economic Dynamics. Harvard University Press, Cambridge, MA.

Wright, B.D., Williams, J.C., 1984. The welfare effects of the introduction of storage. Q. J. Econ. 99, 169–192.