Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods[*]

Alexander Ludwig[†]   Matthias Schön[‡]

April 29, 2015

Abstract

This paper investigates extensions of the method of endogenous gridpoints (ENDGM) introduced by Carroll (2006) to higher dimensions with more than one continuous endogenous state variable. We compare three different categories of algorithms: (i) the conventional method with exogenous grids (EXOGM), (ii) the pure method of endogenous gridpoints (ENDGM) and (iii) a hybrid method (HYBGM). ENDGM comes along with Delaunay interpolation on irregular grids. Comparison of methods is done by evaluating speed and accuracy. We find that HYBGM and ENDGM both dominate EXOGM. In an infinite horizon model, ENDGM also always dominates HYBGM. In a finite horizon model, the choice between HYBGM and ENDGM depends on the number of gridpoints in each dimension. With less than 150 gridpoints in each dimension ENDGM is faster than HYBGM, and vice versa. For a standard choice of 25 to 50 gridpoints in each dimension, ENDGM is 1.4 to 1.7 times faster than HYBGM in the finite horizon version and 2.4 to 2.5 times faster in the infinite horizon version of the model.

JEL Classification: C63, E21.

Keywords: Dynamic Models, Numerical Solution, Endogenous gridpoints Method, Delaunay Interpolation

[†]SAFE, Goethe University Frankfurt; CMR; MEA; Netspar; postal address: House of Finance; Grüneburgplatz 1; 60323 Frankfurt am Main (Germany); e-mail: ludwig@safe.uni-frankfurt.de

[‡]CMR, University of Cologne; postal address: Albertus-Magnus-Platz; 50923 Köln; Germany; e-mail: m.schoen@wiso.uni-koeln.de

1 Introduction

Dynamic models of equilibrium in discrete time are workhorse models in Economics. However, most of these models do not have an analytic closed form solution and equilibria have to be approximated numerically. To this purpose, numerous procedures have been developed in the literature, cf. Judd (1998), Miranda and Fackler (2004). If the problem is differentiable, a popular approach is to use first-order methods, i.e., to iterate on first-order conditions. An important contribution to this literature is Carroll (2006) who introduces the method of endogenous gridpoints (ENDGM). In comparison to the method of exogenous gridpoints (EXOGM), ENDGM greatly enhances computational speed because part of the problem can be computed in closed form.

This paper investigates extensions of Carroll's ENDGM to dynamic problems with more than one continuous endogenous state variable. The key insight of ENDGM is that the choice of the variable on which to define the grid is subject to the user in any dynamic problem. A smart choice may then lead to closed form solutions of first-order conditions, greatly enhancing speed of computations. We here introduce this general idea by first considering the standard implementation of ENDGM in a one-dimensional problem, i.e., in a setup with one endogenous state variable. To this purpose, we introduce some minimal notation, otherwise keeping the presentation as informal as possible. A more in depth treatment is contained in Section 3.

We base the exposition on a consumption-savings problem, as in our application. In a standard exogenous grid method (EXOGM), one solves in each time period (or iteration) for each grid point on grid $\mathcal{G}^a$ of today's state variable a (=assets) some non-linear problem. The solution is given by the associated control variable c (=consumption) and next period's endogenous state variable assets, a'. Solution of this equation also requires interpolation on some function(s) f on a' because generally $a' \notin \mathcal{G}^a$—e.g., f could be the derivative of the value function or, depending on the nature of the problem, the value function itself. Given a, c, a', the additional control savings, s, can be computed. To summarize, the mapping in EXOGM is $a \to (c, a') \to s$ where the mapping $a \to (c, a')$ requires, among other numerical operations, solving a non-linear equation and interpolation. Also observe that, for some regular grid $\mathcal{G}^a$—think, for simplicity, of equally spaced grid points—the "endogenous" grid of a' is generally irregular because the spacing between grid points is a result of the entire mathematical operation.

The trick of ENDGM in such a setup is to revert the mapping, i.e., $s \to (a', c) \to a$. Instead of working on an exogenous grid for a, this is achieved by defining a grid on savings, $\mathcal{G}^s$. Depending on the nature of the problem it is then possible to solve for c (and a') analytically. This is the crucial step: The speed advantage of ENDGM relative to EXOGM is achieved because the mapping $s \to (a', c)$ has a closed form solution. For given contemporaneous variables s, c, and next period's a' one can then endogenously

compute today's endogenous state a. Again observe that, for some regular grid $\mathcal{G}^s$, the "endogenous" grid of a is generally irregular. In subsequent iterations, it is necessary to interpolate on such an irregular grid. In one dimension this does not cause any specific problems.

In this paper we highlight, however, that this irregularity of endogenous grids is the source of a problem specific to ENDGM in higher dimensions. We emphasize that this drawback is not related to the solution of the system of equations per se but results from the endogenously computed states. As we show, the resulting state grid is generally not rectangular, i.e., gridpoints are irregularly distributed in the space. In consequence, even linear interpolation is much more costly than for conventional rectangular grids.

This is easiest to understand again by example. Consider two endogenous state variables a and h, where h is human capital, as in our application. Accordingly, (a', h') are next period's endogenous state variables. Control variables are consumption c, as before, as well as investment in human capital, i. In addition, consider the endogenous controls s (=savings, as before) and current period gross holdings of human capital, z, where z is some function of the human capital stock, h, and the flow investment into human capital, i. Corresponding to the one-dimensional problem the mapping in EX-OGM is $(a, h) \rightarrow (c, i, a', h') \rightarrow (s, z)$ where the mapping $(a, h) \rightarrow (c, i, a', h')$ requires solution of a system of two non-linear equations. In ENDGM, the mapping is again reversed, i.e., $(s, z) \rightarrow (a', h', c, i) \rightarrow (a, h)$. Depending on the nature of the problem, the mapping $(s, z) \rightarrow (a', h', c, i)$ has a closed form solution. As in the application in one dimension, the endogenous grid formed of a, h is irregular. In subsequent iterations one has to interpolate on such an irregular grid. While such an interpolation is unproblematic in one dimension, this irregularity severely complicates location of points for interpolation in higher dimensions.

This exposition clarifies that there exists a fundamental trade-off between EXOGM and ENDGM in higher dimensions. On the one hand, EXOGM requires the use of numerical routines throughout whereas ENDGM computes solutions to first-order conditions in closed form. On the other hand, interpolation in EXOGM is on regular grids and therefore simple. Interpolation in ENDGM on irregular grids is much more complex.

We solve this complex interpolation by Delaunay triangulation (Delaunay 1934). Delaunay interpolation, originally coming from the field of geometry. It was only recently introduced to the field by Brumm and Grill (2014). Broer, Kapicka, and Klein (2013) is the only other (unpublished) paper in Economics we are aware of that applies the method. Our contribution is to investigate its performance in combination with ENDGM.

In addition to EXOGM and ENDGM, we consider a third algorithm, a hybrid method of exogenous gridpoints in one dimension and endogenous gridpoints in the other (HYBGM).[1] Consequently, the endogenously computed grid is only irregular in one dimension.

---

[1]This is similar to the approach of Hintermaier and Koeniger (2010), also see below.

This is a so-called rectilinear grid. Interpolation on a rectilinear grid is easy, just as in the one-dimensional problem. The trade-off between HYBGM and ENDGM is therefore between numerically more costly routines, e.g. Broyden's method, in some dimensions vis-à-vis analytical solutions in all dimensions but a more complex interpolation.

To analyze and to compare these methods we use a simple human capital model. As we already discussed above, this model features two endogenous state variables, financial assets and human capital. Evaluation of methods in this two dimensional setup is done by comparing speed and accuracy of the different approaches.

Our main finding is that HYBGM and ENDGM both dominate EXOGM. They are both substantially faster. In our infinite horizon application, ENDGM also dominates HYBGM. In our finite horizon application, the choice between HYBGM and ENDGM depends on the number of gridpoints in each dimension. For a relatively low number of gridpoints, ENDGM is advantageous and vice versa for HYBGM. We also discuss limitations of ENDGM and HYBGM which are both only applicable to specific problems at hand.

To the best of our knowledge ENDGM in higher dimensions is not yet fully understood. Our paper is an important contribution to fill this gap. Related work by Krueger and Ludwig (2007) and Barillas and Fernandez-Villaverde (2007) extends ENDGM to problems with two control variables but just one endogenous state variable. Hintermaier and Koeniger (2010) use ENDGM in a durable goods model with two endogenous state variables. The main difference of their approach to ours is that ENDGM is only applied in one dimension. Their method still requires solving a nonlinear equation and is thereby very similar to our HYBGM.[2] Our contribution is to implement ENDGM in two dimensions.

Other related literature extends ENDGM to a class of dynamic programming problems with both discrete and continuous choices in which the value function is non-smooth and non-concave, cf. Fella (2014) and Iskhakov et al. (2014).

Our analysis proceeds as follows. Section 2 presents the simple human capital setting on which we base the evaluation of methods. Section 3 introduces the main features of the methods under evaluation, the method of exogenous gridpoints, the pure method of endogenous gridpoints and the hybrid method. Section 4 presents results according to speed and accuracy of all three methods. Section 5 concludes. Additional material is contained in an appendix.

---

[2]One difference to our version of HYBGM is that we solve this non-linear equation with a univariate solver whereas Hintermaier and Koeniger (2010) use interpolation techniques that are generally less accurate. This is essentially analogous to applying a bisection method for one iteration only.

2   General Framework

We develop a consumption and savings model which allows us to illustrate and to compare three approaches to solve dynamic models with two endogenous states using first-order methods. In addition to assets there is a second endogenous state variable, a human or health capital stock (we will use both interpretations interchangeably). Human capital can be accumulated over time and is produced with a nonlinear production function. For expositional purposes we keep the model very simple. For example, despite the degenerate risk of survival, we ignore any stochasticity to the effect that, e.g., wage processes are fully deterministic. Of course, the underlying trade-off between solution methods will also hold in more complex problems.

2.1   A Simple Human Capital Model

A risk averse agent with maximum time horizon T, $T = \infty$ possible, derives utility from consumption, c_t, in each period, with standard additive separable life time utility

$$U = \sum_{t=1}^{T} \beta^{t-1} s\left(h_t\right) u\left(c_t\right),$$

where $\beta \in (0, 1)$ is the discount factor. The instantaneous utility function $u\left(c_t\right)$ as well as the probability to survive to the next period $s\left(h_t\right)$ are assumed to be strictly increasing and concave in their respective arguments. Income of the agent, y_t, consists of labor income which depends on the amount of accumulated human capital, h_t, hence

$$y_t = wh_t,$$

where w is the wage rate.

In each period the household faces the decision to consume, c_t, to invest savings, s_t, in a risk-free financial asset, a_t, which earns (gross) interest R and to invest an amount i_t into human capital, h_t. Human capital depreciates at constant rate δ and is produced by the production function $f\left(i\right)$. We assume that $f_i > 0$, $f_{ii} < 0$ and that the Inada conditions are satisfied, i.e., $\lim_{i_t \to 0} f_i = \infty$ and $\lim_{i_t \to \infty} f_i = 0$.[3] The human capital accumulation equation is accordingly given by

$$h_{t+1} = (1 - \delta)\left(h_t + f\left(i_t\right)\right), \tag{1}$$

where h_0 is given.

---

[3]These conditions are crucial because otherwise it could turn out to be optimal to invest in only one asset. The other asset would be redundant and our problem would collapse to a problem in one dimension.

Financial markets are imperfect and households are not allowed to hold negative financial assets. The dynamic budget constraint writes as

$$a_{t+1} = R(a_t + wh_t - c_t - i_t) \geq 0,$$

where a_0 is given.

Recursive Formulation of the Household Problem   The recursive formulation of the household problem is as follows:

$$V_t(a_t, h_t) = \max_{c_t, i_t, a_{t+1}, h_{t+1}} \{u(c_t) + \beta s(h_{t+1}) V_{t+1}(a_{t+1}, h_{t+1})\}$$

subject to the constraints

$$a_{t+1} = R(a_t + wh_t - c_t - i_t)$$
$$h_{t+1} = (1 - \delta)(h_t + f(i_t))$$
$$a_{t+1} \geq 0$$
$$h_{t+1} > 0. \tag{2}$$

Assumptions on Functional Forms   For our numerical approach we assume that instantaneous utility has the CRRA property with coefficient of relative risk aversion denoted by $\theta > 0$:

$$u(c_t) = \frac{c_t^{1-\theta} - 1}{1 - \theta}.$$

The human capital production function is

$$f(i_t) = \frac{1}{\alpha} i_t^{\alpha}$$

for curvature parameter $\alpha \in (0, 1)$. As to the functional form of the per-period survival probability we follow Hall and Jones (2007) and assume that

$$s(h_t) = 1 - \phi \frac{1}{1 + h_t},$$

for $\phi \in (0, 1]$.

We assume that the value function is strictly concave and unique maximizers are continuous policy functions, cf. Stokey and Lucas (1989). It is well-known that strict concavity of the value function may be violated in models with endogenous human capital formation (value functions may have concave and convex regions). Hence, first-order conditions are generally necessary but not sufficient. In applications, one way to accommodate this is to use first-order methods at the calibration stage of the model (where

speed is an issue). Upon convergence, one can then test for uniqueness by checking for alternative solutions by use of global methods. To focus our analysis we do not further address these aspects here.[4]

Solution The optimal solution is fully characterized by the following set of first-order conditions and constraints:

$$c_t^{-\theta} = \beta R \left(1 - \phi \frac{1}{1 + h_{t+1}}\right) V_{t+1_a}(a_{t+1}, h_{t+1}) \tag{3a}$$

$$\gamma i_t^{-(1-\alpha)} = \frac{R}{(1-\delta)} \frac{V_{t+1_a}(a_{t+1}, h_{t+1})}{\frac{\phi}{(1+h_{t+1}-\phi)(1+h_{t+1})} V_{t+1}(a_{t+1}, h_{t+1}) + V_{t+1_h}(a_{t+1}, h_{t+1})} \tag{3b}$$

$$a_{t+1} = R(a_t + wh_t - c_t - i_t) \tag{3c}$$

$$h_{t+1} = (1-\delta)(h_t + f(i_t)) \tag{3d}$$

$$a_{t+1} \geq 0. \tag{3e}$$

V_{t_a} and V_{t_h} are derivatives of the value function with respect to financial assets and human capital, respectively. The first equation relates today's consumption to consumption of tomorrow, whereas the second equation relates costs and gains of investing in human capital. Notice that constraint (2) can be dropped because of the lower Inada condition of the human capital investment function $f(i)$. Searching for the solution of this model amounts to finding the four optimal policies for consumption, $c_t(\cdot, \cdot)$, investment in human capital, $i_t(\cdot, \cdot)$, next period's financial assets, $a_{t+1}(\cdot, \cdot)$, and next period's human capital, $h_{t+1}(\cdot, \cdot)$, as functions of the two endogenous state variables, financial assets, a_t, and human capital, h_t, that solve equation system (3).

The envelope conditions are:

$$V_{t_a}(a_t, h_t) = u_c = c_t^{-\theta} \tag{4a}$$

$$V_{t_h}(a_t, h_t) = \left(w + \frac{1}{f_i}\right) u_c = \left(w + \frac{1}{\gamma i_t^{-(1-\alpha)}}\right) c_t^{-\theta}. \tag{4b}$$

Using (3a) together with (4a) gives the standard Euler equation of consumption.[5]

2.2 Calibration

We choose the same parametrization of the model for all solution methods described in Section 3. The coefficient of relative risk aversion is set to $\theta = 0.5$ to assure a positive

---

[4]We checked ex-post if value functions are globally concave which they are for the parameter space considered here. A crucial parameter is α as it governs the curvature of the human capital production function. If we were to choose a higher degree of curvature (lower α) than non-concavities may arise. These results are available upon request.

[5]For derivation of (3) and the Envelope conditions see Appendix A.

value of life. We set the time preference rate to $\rho = 0.04$. In order to provide sufficient incentives to save in the finite horizon setting without introducing risk we set an interest rate of $R - 1 = 0.05$. In the infinite horizon setting we set an interest rate of $R - 1 = 0.03$ which is smaller than ρ in order to assure that financial assets are bounded. For the depreciation rate of human capital we take $\delta = 0.05$. The curvature parameter of the human capital production function is $\alpha = 0.35$. The wage rate w is set to 0.1. The survival rate parameter is $\phi = 0.5$.

3 Solution Methods

The main idea of all methods is to exploit the FOCs (3a) and (3b) to compute optimal policies at discrete points that constitute a mesh in the state space. All three methods use the recursive nature of the problem. Correspondingly, in the finite horizon version, the model is solved backwards from the last to the first period ($t = T, T - 1, \ldots, 0$). In the infinite horizon implementation the iteration continues until convergence on policy functions.

Differences between methods arise because of different solution procedures to the multi-dimensional nonlinear equation system (3) and different interpolation methods, respectively. To provide a preview: The first algorithm (EXOGM) applies a multi-dimensional Quasi-Newton method. Standard interpolation methods are used. The second algorithm (ENDGM) uses the method of endogenous gridpoints and thereby solves the system of equations (3) analytically. It is accompanied by Delaunay interpolation. The third algorithm (HYBGM) combines the former two, i.e., it applies the method of endogenous gridpoints (and closed form solutions) in one dimension and uses a one-dimensional Quasi-Newton method in the other dimension. As EXOGM, HYBGM comes along with a standard interpolation procedure.

3.1 Multi-Dimensional Root-Finding with Regular Interpolation (EXOGM)

The most direct approach to solve (3) is to insert the constraints into the FOCs and to rely on a numerical multi-dimensional root-finding routine. Multi-dimensional solvers are necessary because c and i show up on both sides of the respective non-linear equations in (3). In our application we use a Quasi-Newton method, more specifically Broyden's method, cf. Press et al. (1996).

The implementation steps of EXOGM are as follows:

1. To initialize EXOGM predefine two grids, one for financial assets a, $\mathcal{G}^a = \left\{ a^1, a^2, ..., a^K \right\}$ and one for human capital h, $\mathcal{G}^h = \left\{ h^1, h^2, ..., h^J \right\}$ and construct $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$.

2. In period T, savings and investment in human capital are zero as both assets are useless in period $T + 1$[6] and income is completely consumed for all $\left(a^k, h^j\right) \in \mathcal{G}^{a,h}$:

$$c_T\left(\cdot, \cdot\right) = a_T^k + wh_T^j$$
$$i_T\left(\cdot, \cdot\right) = 0.$$

Using the above in equations (4a) and (4b) the value function and its derivatives with respect to a and h in T are

$$V_T\left(a_T^k, h_T^j\right) = \frac{1}{1-\theta}\left(c_T^{k,j}\right)^{1-\theta}$$
$$V_{T_a}\left(a_T^k, h_T^j\right) = \left(c_T^{k,j}\right)^{-\theta}$$
$$V_{T_h}\left(a_T^k, h_T^j\right) = \left(w + \frac{1}{\gamma}\left(i_T^{k,j}\right)^{1-\alpha}\right)\left(c_T^{k,j}\right)^{-\theta} = w\left(c_T^{k,j}\right)^{-\theta}.$$

3. Iterate backwards on $t = T - 1, \ldots, 0$. In each t for each $\left(a_t^k, h_t^j\right) \in \mathcal{G}^{a,h}$:

   (a) Given (suitably interpolated values of) V_{t+1}, V_{t+1_a} and V_{t+1_h}, solve the two-dimensional equation system

$$\left(c_t^{k,j}\right)^{-\theta} = \beta R \left(1 - \phi\frac{1}{1 + (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}\left(i_t^{k,j}\right)^\alpha\right)}\right)$$

$$V_{t+1_a}\left(\overbrace{R\left(a_t^k + wh_t^j - c_t^{k,j} - i_t^{k,j}\right)}^{a_{t+1}^{k,j}}, \overbrace{(1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}\left(i_t^{k,j}\right)^\alpha\right)}^{h_{t+1}^{k,j}}\right)$$

$$\gamma\left(i_t^{k,j}\right)^{-(1-\alpha)} = \frac{R}{(1-\delta)}\frac{V_{t+1_a}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right)}{\frac{\phi}{\left(1+h_{t+1}^{k,j}-\phi\right)\left(1+h_{t+1}^{k,j}\right)}V_{t+1}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right) + V_{t+1_h}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right)}$$

   for $c_t^{k,j}$ and $i_t^{k,j}$ using Broyden's method. If $c_t^{k,j} + i_t^{k,j} > a_t^k + wh_t^j$ (binding

---

[6]This rationale does not imply that h must be zero in period $T + 1$ because human capital is—in contrast to financial assets—inalienable.

borrowing constraint) recompute $i_t^{k,j}$ by solving

$$\left(a_t^k + wh_t^j - i_t^{k,j}\right)^{-\theta} - $$
$$\frac{1}{\left(\left(1-\delta\right)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right)^2} V_{t+1_a}\left(0, (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right)\beta\left(1-\delta\right)\gamma(i_t^{k,j})^{-(1-\alpha)}$$
$$- \left(1 - \frac{1}{\left(\left(1-\delta\right)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right)}\right)V_{t+1_h}\left(0, (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right)\cdot$$
$$\beta\left(1-\delta\right)\gamma(i_t^{k,j})^{-(1-\alpha)} = 0$$

for $i_t^{k,j}$. Next, re-compute $c_t^{k,j} = a_t^k + wh_t^j - i_t^{k,j}$.

(b) Save/Update both the value function and its derivatives

$$V_t\left(a_t^k, h_t^j\right) = \frac{1}{1-\theta}\left(c_t^{k,j}\right)^{1-\theta} + \beta\left(1 - \phi\frac{1}{1+h_{t+1}^{k,j}}\right)V_{t+1}(a_{t+1}^{k,j}, h_{t+1}^{k,j})$$

$$V_{t+1_a}\left(a_t^k, h_t^j\right) = \left(c_t^{k,j}\right)^{-\theta}$$

$$V_{t+1_h}\left(a_t^k, h_t^j\right) = \left(w + \frac{1}{\gamma}\left(i_t^{k,j}\right)^{1-\alpha}\right)\left(c_t^{k,j}\right)^{-\theta}.$$

Since EXOGM requires to apply the solver for each point in $\mathcal{G}^{a,h}$, this procedure entails solving the multidimensional equation system $[K \cdot J]$ times in each $t = T - 1, \ldots, 0$. Depending on the stopping criterion in the numerical routine this could be either quite costly in terms of computing time or the computed solutions suffer under low accuracy. An additional shortcoming of EXOGM compared to ENDGM and HYBGM is that the region where the borrowing constraint is binding is not determined.[7] In consequence, policy functions are imprecise at the kink. This may also cause convergence problems. Furthermore, numerical methods often require fine tuning so that stability of numerical routines is ascertained. We initially encountered several such instability problems which we managed to fix by setting options of the solver accordingly.[8]
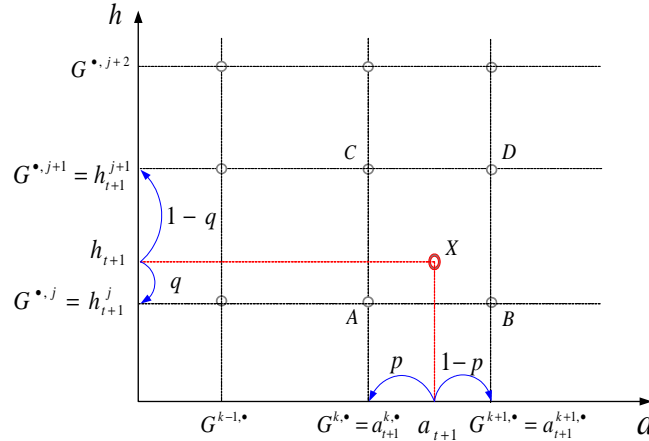
Interpolation on a Rectilinear Grid   Step 3a requires evaluation of both the value function V_{t+1} and its derivatives, V_{t+1_a} and V_{t+1_h}. As, in general, $(a_{t+1}^{k,j}, h_{t+1}^{k,j}) \notin \mathcal{G}^{a,h}$ we have to interpolate these functions. We apply bilinear interpolation. Precisely, we determine interpolation nodes by the concept "grid square", cf. Press et al. (1996). In

---

[7]In principle, this could be accommodated by an additional rootfinder to detect the kink—i.e., the a, h-combination at which the borrowing constraint just becomes unbinding—and to add in additional grid points there. We do not extend the method along this dimension. A naive extension along these lines would further slow down EXOGM. However, see Brumm and Grill (2014) for a sophisticated application.

[8]An alternative would be to avoid multivariate solvers and to instead use fixed point iterations with nested univariate solvers. However, this would further slow down EXOGM.

order to apply this procedure it is necessary to have a rectilinear grid, i.e., the state space has to be tessellated by rectangles.[9] In this case all gridpoints in row $\mathcal{G}^{\bullet,j}$ have the same value of h^j, and all gridpoints in column $\mathcal{G}^{k,\bullet}$ have the same value of a^k. The problem of locating a point in a multi-dimensional grid is split up into several problems of locating the point in one dimension. Within each dimension and a total number of N points in the point set closest neighbors in the grid are identified in about $\log_2 N$ trials using bisection methods. Figure 1 shows the location of interpolation nodes $[A; B; C; D]$ for point X in a two-dimensional rectilinear grid.

Figure 1: Rectilinear Grid

Notes: Interpolation on rectilinear grids. In any row locate the two columns ($G^{k,\bullet}$ and $G^{k+1,\bullet}$) that form the most narrow bracket of a_{t+1}. In any column locate the two rows ($G^{\bullet,j}$ and $G^{\bullet,j+1}$) that form the most narrow bracket of h_{t+1}. Interpolation nodes: $(k, j); (k + 1, j); (k, j + 1); (k + 1, j + 1)$.

In EXOGM, $\mathcal{G}^a \otimes \mathcal{G}^h$ is predetermined as a rectilinear grid (in every iteration). After locating the nodes, bi-linear interpolation of any function of F—in our case the value function in t as well as its first derivatives with respect to a and h—at point X requires computing $F(X) = \varphi_A F(A) + \varphi_B F(B) + \varphi_C F(C) + \varphi_D F(D)$ with the four basis functions φ where $\varphi_A = p \cdot q$, $\varphi_B = (1 - p) \cdot q$, $\varphi_C = p \cdot (1 - q)$ and $\varphi_D = (1 - p) \cdot (1 - q)$ with $p = \frac{a_X - a_A}{a_B - a_A}$ and $q = \frac{h_X - h_A}{h_C - h_A}$, cf. Judd (1998).

3.2 Analytical Solution with Delaunay Interpolation (ENDGM)

The above setting has a straightforward economic interpretation. Given an exogenous state today (a_t, h_t) compute the endogenous state variables (a_{t+1}, h_{t+1}). The main idea of ENDGM is to redefine exogenous and endogenous objects in the numerical solution: the grid of contemporaneous control variables is taken as exogenous whereas the grid of today's state variables is determined endogenously.

---

[9]Notice that these rectangles do not necessarily have to be congruent to each other.

In our two-dimension setup, implementation of the method requires definition of two endogenous control variables on which to base the exogenous grids. To this purpose define by

$$s_t \equiv a_t + wh_t - c_t - i_t = \frac{a_{t+1}}{R} \tag{5a}$$

$$z_t \equiv h_t + f(i_t) = \frac{h_{t+1}}{1 - \delta} \tag{5b}$$

return adjusted stock of physical and human capital, respectively. Our implementation of the method defines grids on (s_t, z_t) and maps from (s_t, z_t) to (a_{t+1}, h_{t+1}) by $a_{t+1} = Rs_t$ and $h_{t+1} = (1 - \delta) z_t$.[10] Next, the system of FOCs can be solved analytically to determine the corresponding set of contemporaneous controls, (c_t, i_t). Finally, we use the budget constraint and the law of motion for human capital to get the corresponding endogenous state variables, (a_t, h_t). Precisely, the implementation steps are as follows:

1. To initialize ENDGM predefine two grids, one for gross savings s, $\mathcal{G}^s \equiv \left\{ s^n, s^{n+1}, ..., s^K \right\}$ and one for gross investment in human capital z, $\mathcal{G}^z \equiv \left\{ z^1, z^2, ..., z^J \right\}$ as defined in (5) and form $\mathcal{G}^{s,z} = \mathcal{G}^s \otimes \mathcal{G}^z$.

2. Define $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$ for T. Compared to $\mathcal{G}^s$, the grid $\mathcal{G}^a$ includes n additional gridpoints. These gridpoints represent the region in which the borrowing constraint is binding (see step 3e). In period T, as in EXOGM,

$$c_T(\cdot, \cdot) = a_T^{k,j} + wh_T^{k,j}$$
$$i_T(\cdot, \cdot) = 0$$

for all $\left(a^{k,j}, h^{k,j} \right) \in \mathcal{G}^{a,h}$ and

$$V_T \left(a_T^{k,j}, h_T^{k,j} \right) = \frac{1}{1 - \theta} \left(c_T^{k,j} \right)^{1-\theta}$$
$$V_{T_a} \left(a_T^{k,j}, h_T^{k,j} \right) = \left(c_T^{k,j} \right)^{-\theta}$$
$$V_{T_h} \left(a_T^{k,j}, h_T^{k,j} \right) = \left(w + \frac{1}{\gamma} \left(i_T^{k,j} \right)^{1-\alpha} \right) \left(c_T^{k,j} \right)^{-\theta} = w \left(c_T^{k,j} \right)^{-\theta}.$$

3. Iterate backwards from $t = T - 1, ..., 0$. In each t, for each $\left(s^k, z^j \right) \in \mathcal{G}^{s,z}$:

---

[10]In deterministic model such as ours, this mapping is of course deterministic. We could therefore directly work on a grid of (a_{t+1}, h_{t+1}). However, this would generally not be possible in a stochastic model because the realizations of (a_{t+1}, h_{t+1}) depend on the realizations of shocks in period $t+1$. For sake of generality, we therefore define the grid on (s_t, z_t).

(a) Compute a_{t+1}^k and h_{t+1}^j:

$$a_{t+1}^k = Rs^k,$$
$$h_{t+1}^j = (1-\delta)\, z^j.$$

(b) Given V_{t+1}, V_{t+1_a} and V_{t+1_h} interpolate the value function and its derivatives at $\left(a_{t+1}^k, h_{t+1}^j\right)$ to get (interpolated values of) $V_{t+1}\left(a_{t+1}^k, h_{t+1}^j\right)$, $V_{t+1_a}\left(a_{t+1}^k, h_{t+1}^j\right)$ and $V_{t+1_h}\left(a_{t+1}^k, h_{t+1}^j\right)$ using Delaunay interpolation (see below).

(c) Compute $c_t^{k,j}$ and $i_t^{k,j}$:

$$c_t^{k,j} = \left(\beta R \left(1 - \phi \frac{1}{1+(1-\delta)\,z^j}\right) V_{t+1_a} \left(\overbrace{Rs^k}^{a_{t+1}^k}, \overbrace{(1-\delta)\,z^j}^{h_{t+1}^j} \right) \right)^{-\frac{1}{\theta}},$$

$$i_t^{k,j} = \frac{1}{\gamma} \left(\frac{R}{(1-\delta)} \frac{V_{t+1_a}\left(a_{t+1}^k, h_{t+1}^j\right)}{\frac{\phi}{\left(1+h_{t+1}^j-\phi\right)\left(1+h_{t+1}^j\right)} V_{t+1}\left(a_{t+1}^k, h_{t+1}^j\right) + V_{t+1_h}\left(a_{t+1}^k, h_{t+1}^j\right)} \right)^{-\frac{1}{1-\alpha}}.$$

(d) Compute $a_t^{k,j}$ and $h_t^{k,j}$:

$$h_t^{k,j} = z^j - \frac{\gamma}{\alpha}\left(i_t^{k,j}\right)^\alpha$$
$$a_t^{k,j} = s^k - w h_t^{k,j} + c_t^{k,j} + i_t^{k,j}.$$

(e) If $a_t^{n+1,j} > 0$, define for each j an auxiliary grid $\mathcal{G}^{aux} \equiv \{a^1, a^2, ..., a^n\}$ between 0 and $a_t^{n+1,j}$.[11] In this region the borrowing constraint is binding. Compute $i_t^{k,j}$ by solving

$$\left(a_t^k + w \left(\frac{h_{t+1}^j}{1-\delta} - \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha \right) - i_t^{k,j} \right)^{-\theta}$$
$$- \frac{1}{\left(h_{t+1}^j\right)^2} V_{t+1_a}\left(0, h_{t+1}^j\right) \beta \left(1-\delta\right) \gamma (i_t^{k,j})^{-(1-\alpha)}$$
$$- \left(1 - \frac{1}{h_{t+1}^j}\right) V_{t+1_h}\left(0, h_{t+1}^j\right) \beta \left(1-\delta\right) \gamma (i_t^{k,j})^{-(1-\alpha)} = 0$$

using a non-linear solver. Then compute $c_t^{k,j} = a_t^k + w \left(\frac{h_{t+1}^j}{1-\delta} - \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha \right) - i_t^{k,j}$.

---

[11]If $a_t^{n+1,j} \leq 0$, the borrowing constraint is not binding and we add in some artificial numbers for the solution here. Observe that the method can be further improved by working with t- (or iteration-) dependent grids, an approach we do not adopt here.

(f) Save/Update both the value function and its derivatives

$$V_t\left(a_t^{k,j}, h_t^{k,j}\right) = \frac{1}{1-\theta}\left(c_t^{k,j}\right)^{1-\theta} + \beta\left(1 - \phi\frac{1}{1+h_{t+1}^j}\right)V_{t+1}\left(a_{t+1}^k, h_{t+1}^j\right)$$

$$V_{t_a}\left(a_t^{k,j}, h_t^{k,j}\right) = \left(c_t^{k,j}\right)^{-\theta}$$

$$V_{t_h}\left(a_t^{k,j}, h_t^{k,j}\right) = \left(w + \frac{1}{\gamma}\left(i_T^{k,j}\right)^{1-\alpha}\right)\left(c_t^{k,j}\right)^{-\theta}.$$

The clear advantage of ENDGM compared to EXOGM becomes obvious in step 3c. By conditioning on the grid of s_t and z_t the system of FOCs can be solved for c_t and i_t analytically and hence no numerical root-finder is needed. Furthermore, ENDGM provides, by construction, an exact determination of the range of the borrowing constraint and produces higher accuracy of the solution than EXOGM in this region. However, in contrast to the standard one-dimensional problem considered by Carroll (2006), the policy function itself does not have a closed form solution in this range, see step 3e.[12]

Remark 1 *In contrast to EXOGM, ENDGM is not a general method. Suppose we were to adopt a general Ben-Porath human capital function, cf. Ben-Porath (1967), in which the level of human capital directly affects the productivity of human capital investments, i.e., we replace $f(i)$ in equation (1) with $f(h,i)$. ENDGM is no longer applicable in such a formulation. This exemplifies that an application of ENDGM often requires specific modeling assumptions.*

Delaunay Interpolation   In EXOGM the grid is rectilinear by construction whereas in ENDGM the endogenously computed grid $\mathcal{G}^{a,h}$ is not. This constitutes the main drawback of ENDGM because location of interpolation nodes is not obvious. As illustrated in Figure 2, separating the multi-dimensional problem into several one-dimensional problems is not possible. In each row not just the value of a changes but also the value of h so that the concept of bi-linear interpolation in a square grid is not applicable. ENDGM hence generates a situation where neighboring points in the state space do not need to be neighboring elements in the grid matrix.

The most common approach adopted in other scientific fields such as geometry or geography to locate neighboring points in an irregular grid is the concept of Delaunay triangulation and its related geometric construct, the Voronoi diagram. We explain the geometric construction of the Voronoi diagram by use of Figure 3. The Voronoi diagram (polygon)—shown in Panel (a) of Figure 3—is the region of the state space consisting of all points closer to gridpoint P_1 than to any other gridpoint. The Voronoi diagram

---

[12]In a standard consumption-savings model with only one endogenous continuous state variable the policy function is computed by linearly interpolating between the policy at zero saving and the origin, cf. Carroll (2006).

Figure 2: Irregular Grid

Panel (a)   Panel (b)
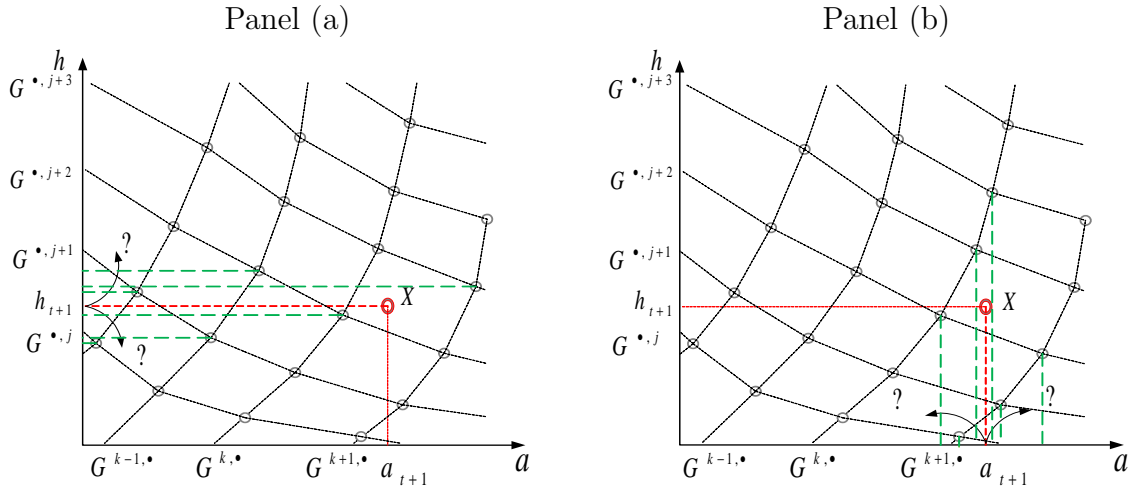
Notes: Interpolation on irregular grids. Multidimensional interpolation cannot be separated into several one-dimensional interpolations as the values of a and h change in each column or row.

is obtained from the perpendicular bisectors of the lines connecting neighboring points. Voronoi diagrams for all points form a tessellation of the space, cf. Panel (a). Edges of the Voronoi diagram are all the points in the plane that are equidistant to the two nearest gridpoints, cf. Panel (b). The Voronoi vertices are the points equidistant to three gridpoints, i.e., they are the center of circumcircles including the three neighboring gridpoints, cf. Panel (c). Connecting these gridpoints constitutes the unique triangulation known as the Delaunay triangulation as displayed in Panel (d), cf. Baker (1999). The vertices of a triangle are the nearest neighbors of all points contained in that triangle. These concepts can also be generalized to more than two dimensions.

The computational implementation of a Delaunay triangulation is done by the so-called randomized incremental algorithm, illustrated in Figure 4. It is incremental in the sense that it adds points to the triangulation one at a time to maintain a Delaunay triangulation at each stage. It is randomized in that points are added in a random order which guarantees $O(N \log N)$ expected time for the algorithm where N is the total number of points in the point set, cf. Press et al. (2007). To construct the Delaunay triangulation for a given point set we initially have to add three "fictitious" points $[\Theta_1, \Theta_2, \Theta_3]$, forming a large starting triangle which encloses all "real" points, cf. Panel (a) of Figure 4. This is necessary in order to ensure that added points lie within an existing triangle. These "fictitious" points are deleted once the triangulation is complete. In each following step of Delaunay triangulation a point from the point set is added to the existing triangulation and connected to the vertices of the enclosing triangle. We illustrate this step in Panel (b) of the figure. Consider the existing triangle P_1, P_2, P_3 and a new point from the point set, P_5, which is not yet connected to other points. Connecting P_5 to P_1, P_2 and P_3, respectively, gives rise to three new triangles. Next, it is checked whether the

Figure 3: The Voronoi Diagram

Panel (a)

Panel (b)
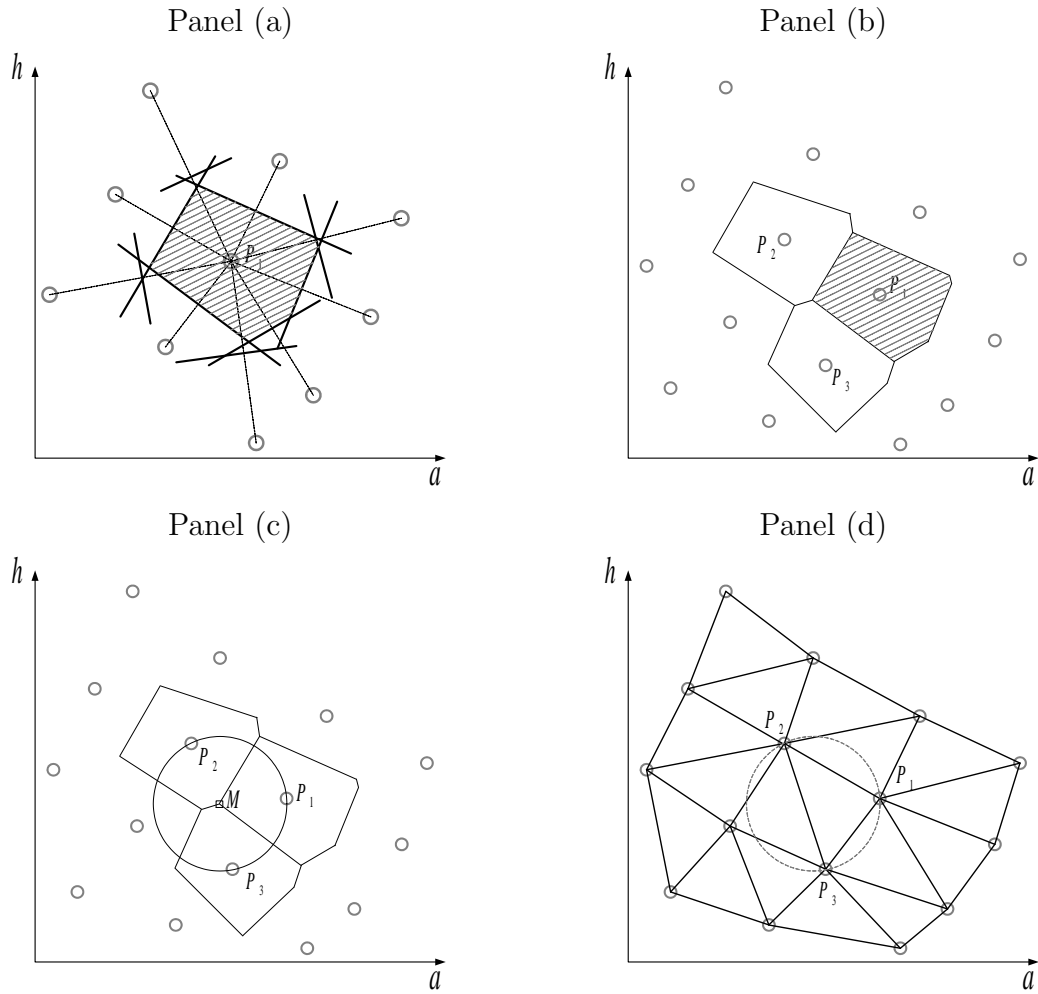
Panel (c)

Panel (d)

Notes: Panel (a): Generating the Voronoi polygon: Edges are perpendicular bisectors of lines connecting neighboring points. Panel (b): Several Voronoi tiles in mesh grid. Panel (c): Circle with center at vertex includes three closest points. Panel (d): Delaunay Triangulation: Vertices are nearest neighbors of all points within triangle.

newly created triangles are "legal", i.e., whether the circumcircle of any triangle does not contain any other point of the point set.[13] In our example, we first visit triangle P_2, P_3, P_5 in Panel (c). As shown in the figure, the circumcircle contains point P_4. Hence, triangle P_2, P_3, P_5 is not legal. Therefore, flip the edge opposite of P_5 connecting P_5 with P_4. This operation creates two new triangles, P_3, P_4, P_5 and P_2, P_4, P_5, cf. Panel (d) of the figure, which must be checked for legality. In our example, triangle P_3, P_4, P_5 is legal because the circumcircle does not contain other existing points from the point set. The process is recursive and never wanders away from any point P (point P_5 in our example). The only edges that can be made illegal by inserting a point P are edges opposite P (in triangles with P as a vertex).[14]

At interpolation, to locate a (query) point X in a given planar triangular mesh we adopt a procedure referred to as visibility walk, illustrated in Figure 5. The search starts from an initial guess of a triangle, Δ_1. Then, it is tested if the line supporting the first edge e separates Δ_1 from the query point X which reduces to a single operation test. If this is the case, the next triangle being visited is the neighbor of Δ_1 through e, Δ_2. Otherwise the second edge is tested in the same way. In case the test for the second edge also fails then the third edge is tested. The failure of this third test means that the goal has been reached. In Figure 5, this would be the case at triangle Δ_X which contains X.[15] Devillers et al. (2001) find that performance of the visibility walk is better than other possible algorithms. The location step for the visibility walk takes only $O \log (N)$ operations, cf. Press et al. (2007). The starting triangle may be arbitrary. However, an informed choice may radically shorten the length of the walk. We accommodate this by initializing the search with our solutions to gridpoints visited previously.

After locating the triangle we compute the normalized barycentric coordinates (weights) of the query point X with respect to the vertices (A, B, C) of the triangle Δ_X,

$$\varphi_A = \frac{(a_X - a_C)(h_B - h_C) + (a_C - a_B)(h_X - h_C)}{(a_A - a_C)(h_B - h_C) + (a_C - a_B)(h_A - h_C)}$$

$$\varphi_B = \frac{(a_X - a_C)(h_C - h_A) + (a_A - a_C)(h_X - h_C)}{(a_A - a_C)(h_B - h_C) + (a_C - a_B)(h_A - h_C)}$$

$$\varphi_C = 1 - \varphi_A - \varphi_B.$$

Finally, the interpolated value of any function F at point X is given as the weighted

---

[13]This principle is derived from the definition that a triangulation fulfills the Delaunay property if and only if the circumcircle of any triangle does not contain a point in its interior, cf. de Berg et al. (2008).

[14]This procedure is described in Press et al. (2007). We use the numerical package geompack3 based on Joe (1991) for both the Delaunay triangulation and the "visibility walk", described next.
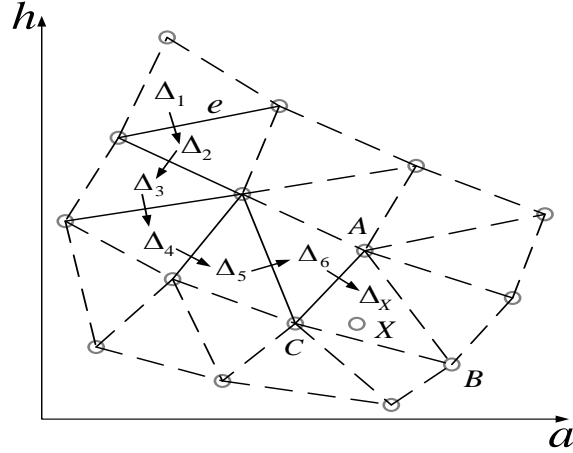
[15]In non-Delaunay triangulations, the visibility walk may fall into a cycle, whereas in Delaunay triangulations the visibility walk always terminates, cf. Devillers et al. (2001).

Figure 4: Incremental Algorithm
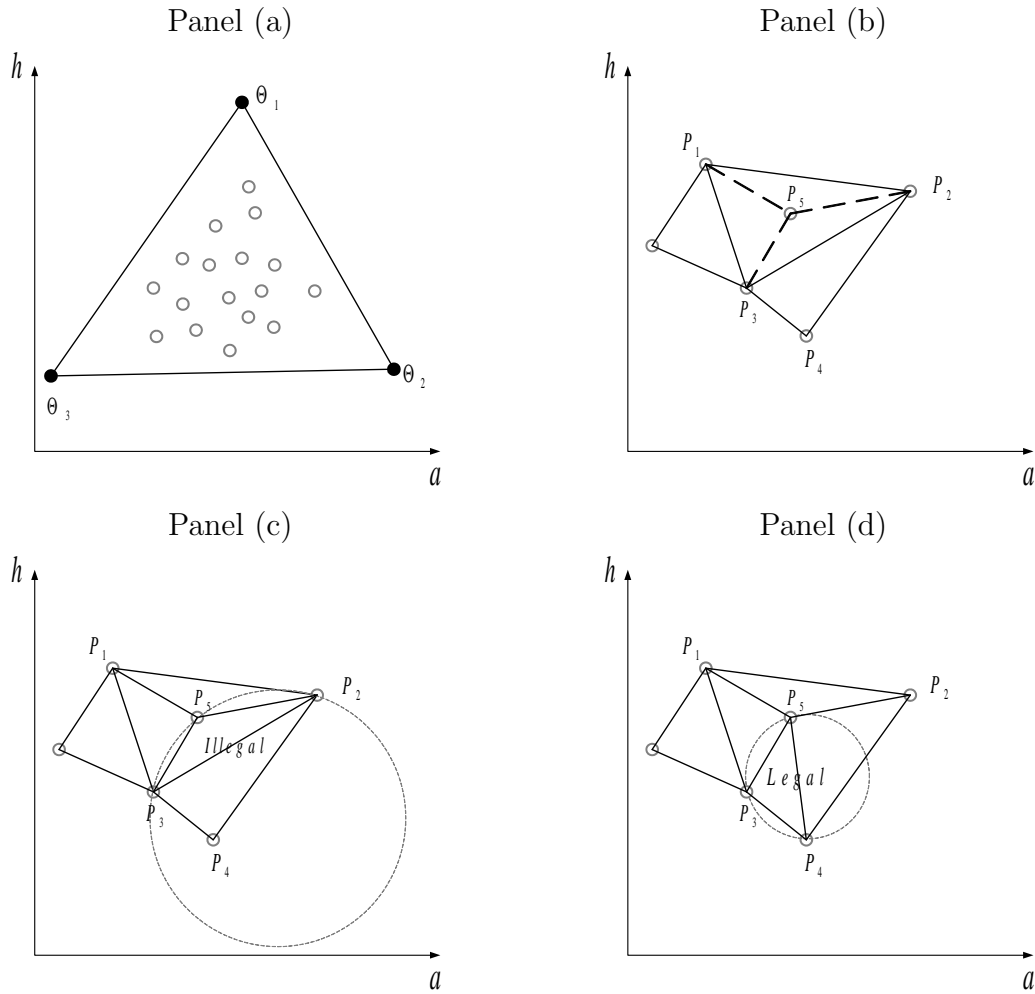
Notes: Panel (a): Three "fictional" points added to constitute the first triangle which includes all "real" points of the point set. Panel (b): Point added to existing Delaunay Triangulation and connected to vertices of enclosing triangle. Panel (c): Circumcircle contains a point. and is therefore illegal triangle. Panel (d): Circumcircle does not contain any point and is therefore legal.

Figure 5: Visibility Walk

Notes: Visibility walk in Delaunay triangulation - Locate triangle Δ_X containing X with initial guess Δ_1. If the line supporting e separates Δ from X, which reduces to a single orientation test, then the next visited triangle is the neighbor of Δ through e.

average of the respective function values at the vertices,

$$F(X) = \varphi_A F(A) + \varphi_B F(B) + \varphi_C F(C).$$

In our code we also incorporate the option of a multi-linear interpolation used by Broer et al. (2013). This alternative interpolation method is very useful in applications in which existing triangles are visited frequently. In our specific applications, this is, however, not the case so that the method does not have an advantage over the simple interpolation method we use. We therefore do not apply it when generating our results below.[16]

---

[16]The basic idea of multilinear interpolation is as follows: We can write

$$\begin{bmatrix} a_X \\ h_X \end{bmatrix} = \begin{bmatrix} a_A \\ h_A \end{bmatrix} + s \begin{bmatrix} a_B - a_A \\ h_B - h_A \end{bmatrix} + t \begin{bmatrix} a_C - a_B \\ h_C - h_B \end{bmatrix} = \begin{bmatrix} a_A \\ h_A \end{bmatrix} + A \begin{bmatrix} s \\ t \end{bmatrix}, \text{ for } A = \begin{bmatrix} a_B - a_A & a_C - a_B \\ h_B - h_A & h_C - h_B \end{bmatrix}$$

and some scalars s and t. Given (a_X, h_X), the solution for (s, t) is accordingly given by

$$\begin{bmatrix} s \\ t \end{bmatrix} = A^{-1} \begin{bmatrix} a_X - a_A \\ h_X - h_A \end{bmatrix} = A^{-1} \begin{bmatrix} a_X \\ h_X \end{bmatrix} - b, \text{ where } b = A^{-1} \begin{bmatrix} a_A \\ h_A \end{bmatrix}. \tag{6}$$

The value of function $F(X)$ then follows as

$$F(X) = F(A) + s\left(F(B) - F(A)\right) + t\left(F(C) - F(B)\right). \tag{7}$$

Matrix A^{-1} and vector b must only be computed once when triangle Δ_X is visited for the first time and can accordingly be stored. In subsequent visits of Δ_X one can compute, for any point $\tilde{X} \in \Delta_X$, the scalars $(\tilde{s}, \tilde{t})$ directly from equation (6) and the interpolated value from equation (7).

3.3 One-Dimensional Root-Finding with Hybrid Interpolation (HYBGM)

We next consider a hybrid method (HYBGM) which combines EXOGM and ENDGM. Specifically, we use ENDGM in one dimension of the problem only. Hence, we define one of the two state variables on an "endogenous" grid, whereas the other is on an "exogenous" grid. The algorithm proceeds in three steps. In the first step, conditioning on control variable s_t and period t endogenous state h_t, we compute next period's endogenous state variable a_{t+1} and exploit one of the two FOCs to derive the value of one period t control variable—in this setup investment in human capital, i_t. In this step a one-dimensional solver is required. To preserve comparability with the previously described methods we choose Broyden's method.[17] In the second step, control i_t is used to get the value of the second period $t+1$ endogenous state variable, h_{t+1} from the budget constraint. Exploiting the second FOC we can then compute the second control variable, c_t. In the third step, we compute the corresponding endogenous state variable a_t from the budget constraint. The implementation steps are as follows:

1. To initialize HYBGM predefine two grids, one for gross savings s, $\mathcal{G}^s \equiv \left\{ s^1, s^2, ..., s^K \right\}$ and one for human capital h, $\mathcal{G}^h \equiv \left\{ h^1, h^2, ..., h^J \right\}$ and form $\mathcal{G}^{s,h} = \mathcal{G}^s \otimes \mathcal{G}^h$

2. In period T, define an initial guess for $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$. $\mathcal{G}^a$ includes n additional gridpoints compared to $\mathcal{G}^s$. These gridpoints represent the region in which the borrowing constraint is binding (see step 3d). Compute

$$c_T \left(\cdot, \cdot \right) = a_T^{k,j} + w h_T^j$$
$$i_T \left(\cdot, \cdot \right) = 0$$

for all $\left(a_T^{k,j}, h_T^j \right) \in \mathcal{G}^{a,h}$ and

$$V_T \left(a_T^{k,j}, h_T^j \right) = \frac{1}{1 - \theta} \left(c_T^{k,j} \right)^{1 - \theta}$$
$$V_{T_a} \left(a_T^{k,j}, h_T^j \right) = \left(c_T^{k,j} \right)^{-\theta}$$
$$V_{T_h} \left(a_T^{k,j}, h_T^j \right) = \left(w + \frac{1}{\gamma} \left(i_T^{k,j} \right)^{1 - \alpha} \right) \left(c_T^{k,j} \right)^{-\theta}.$$

3. Iterate backwards on $t = T - 1, ..., 0$. In each t, for each $\left(s^k, h^j \right) \in \mathcal{G}^{s,h}$:

    (a) Compute $a_{t+1}^k = R s^k$.

---

[17]Using Brent's method instead turns out to slow down speed of HYBGM.

(b) Given (suitably interpolated values of) V_{t+1}, V_{t+1_a} and V_{t+1_a}, solve the one-dimensional equation system for $i_t^{k,j}$

$$i_t^{k,j} = \frac{1}{\gamma} \left(\frac{R}{(1-\delta)} \frac{V_{t+1_a} \left(\overbrace{Rs^k}^{a_{t+1}^k}, \overbrace{(1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}\left(i_t^{k,j}\right)^\alpha\right)}^{h_{t+1}^{k,j}} \right)}{\frac{\phi}{\left(1+h_{t+1}^{k,j}-\phi\right)\left(1+h_{t+1}^{k,j}\right)} V_{t+1}\left(a_{t+1}^k, h_{t+1}^{k,j}\right) + V_{t+1_h}\left(a_{t+1}^k, h_{t+1}^{k,j}\right)} \right)^{-\frac{1}{1-\alpha}}$$

using Broyden's method. This includes several computations of $h_{t+1}^{k,j} = (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}\left(i_t^{k,j}\right)^\alpha\right)$ and hybrid interpolations —described below— on V_{t+1}, V_{t+1_a} and V_{t+1_h}.

(c) Compute $c_t^{k,j}$ as

$$c_t^{k,j} = \left(\beta R \left(1 - \phi\frac{1}{1+h_{t+1}^{k,j}} \right) V_{t+1_a}\left(Rs^k, h_{t+1}^{k,j}\right) \right)^{-\frac{1}{\theta}}.$$

(d) If $a_t^{n+1,j} > 0$, define for each j an auxiliary grid $\mathcal{G}^{aux} \equiv \{a^1, a^2, ..., a^n\}$ between 0 and $a_t^{n+1,j}$. In this region the borrowing constraint is binding. Compute $i_t^{k,j}$ by solving

$$\left(a_t^k + wh_t^j - i_t^{k,j}\right)^{-\theta} -$$

$$\frac{1}{\left((1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right)^2} V_{t+1_a}\left(0, (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right) \beta(1-\delta)\gamma(i_t^{k,j})^{-(1-\alpha)}$$

$$- \left(1 - \frac{1}{(1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)}\right) V_{t+1_h}\left(0, (1-\delta)\left(h_t^j + \frac{\gamma}{\alpha}(i_t^{k,j})^\alpha\right)\right) \cdot$$

$$\beta(1-\delta)\gamma(i_t^{k,j})^{-(1-\alpha)} = 0$$

for $i_t^{k,j}$. Next, compute $c_t^{k,j} = a_t^k + wh_t^j - i_t^{k,j}$.

(e) Compute $a_t^{k,j}$ from the budget constraint, hence

$$a_t^{k,j} = s^k - wh_t^j + c_t^{k,j} + i_t^{k,j}.$$

(f) Save/Update both the value function and its derivatives

$$V_t\left(a_t^{k,j}, h_t^j\right) = \frac{1}{1-\theta}\left(c_t^{k,j}\right)^{1-\theta} + \beta\left(1 - \phi\frac{1}{1+h_{t+1}^{k,j}}\right)V_{t+1}(a_{t+1}^k, h_{t+1}^{k,j})$$

$$V_{t_a}\left(a_t^{k,j}, h_t^j\right) = \left(c_t^{k,j}\right)^{-\theta}$$

$$V_{t_h}\left(a_t^{k,j}, h_t^j\right) = \left(w + \frac{1}{\gamma}\left(i_T^{k,j}\right)^{1-\alpha}\right)\left(c_t^{k,j}\right)^{-\theta}.$$

As EXOGM, HYBGM requires to run a numerical solver $[K \cdot J]$ times in each $t = T-1, \ldots, 0$. However, computational burden is alleviated by reducing complexity of the equation system. Furthermore, as in ENDGM, it is possible to exactly determine the range of the borrowing constraint. In contrast to ENDGM in two dimensions, there is no need for a complex interpolation method.

Remark 2 *As ENDGM, HYBGM is not a general method. Suppose that consumption has an additional effect on human (or health capital). Consider for example an application where health capital is negatively affected by the consumption of junk food. Correspondingly rewrite (1) to $h_{t+1} = (1 - \delta)(h_t + f(i_t) - g(c_t))$ to the effect that both controls c_t and i_t appear on both sides of the equation system even after applying the reformulation of endogenous states. This renders HYBGM inapplicable.*

Hybrid Interpolation Hybrid interpolation, illustrated in Figure 6, is defined on a curvilinear grid where one dimension is being held constant. To locate any query point X hybrid interpolation proceeds in three steps. First, in the dimension of the exogenous grid (current state h_t) find the most narrow bracket of h_{t+1} and compute the weights according to the relative distance to these gridpoints. Second, in both rows, find those gridpoints that form the most narrow bracket of a_{t+1} and compute the according weights. Third, interpolation of any function of F at point X requires computing $F(X) = \varphi_A F(A) + \varphi_B F(B) + \varphi_C F(C) + \varphi_D F(D)$ with the four basis functions φ where $\varphi_A = p \cdot q$, $\varphi_B = (1-p) \cdot q$, $\varphi_C = r \cdot (1-q)$ and $\varphi_D = (1-r) \cdot (1-q)$ with $p = \frac{a_X - a_A}{a_B - a_A}$, $r = \frac{a_X - a_C}{a_D - a_C}$ and $q = \frac{h_X - h_C}{h_C - h_A}$. Thus, HYBGM reduces complexity of the problem without involving advanced interpolation procedures.

4 Results

We present results separately for the finite and infinite horizon versions of our model. Throughout, we use triple exponential grids for a, h, s, z, respectively. We set the range of grid $\mathcal{G}_s$ to $[0, 500]$ and of $\mathcal{G}_z$ to $[1, 500]$. The according grids $\mathcal{G}_a$ and $\mathcal{G}_h$ are adjusted to

Figure 6: Hybrid Interpolation
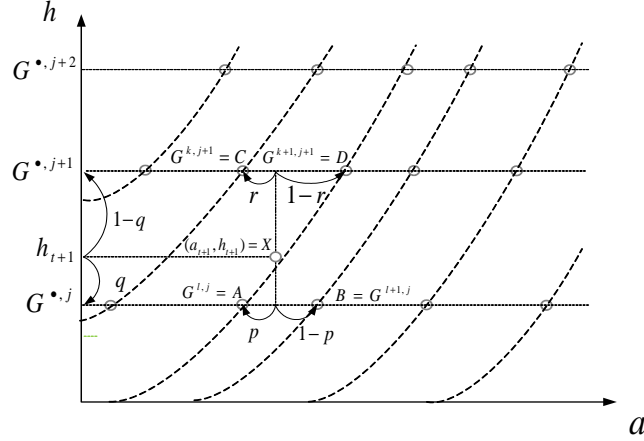
Notes: Hybrid Interpolation. First, in the exogenous dimension, locate the two rows $G^{\bullet,j}$ and $G^{\bullet,j+1}$ that form the most narrow bracket of h_{t+1}. Second, locate in these two rows the gridpoints that form the most narrow bracket of a_{t+1}. Interpolation nodes: (k,j); $(k,j+1)$; $(l,j+1)$; $(l+1,j+1)$.

cover the corresponding range of the state space.[18]

4.1 Error Evaluation

In both the finite and the infinite horizon version of the model, evaluation of accuracy of the solution is done by applying normalized Euler equation errors, cf. Judd (1992), as has become standard in the literature, cf., e.g., Santos (2000) and Barillas and Fernandez-Villaverde (2007). In our approach we get the Euler equation errors e_1 and e_2 by using the respective envelope conditions and combine them with the FOCs to get:

$$e_{1,t} = 1 - \frac{\left(Rs(h_{t+1})\beta\left(c_{t+1}\right)^{-\theta}\right)^{-\frac{1}{\theta}}}{c_t}, \tag{8a}$$

$$e_{2,t} = 1 - \frac{\left(\frac{R}{(1-\delta)}\left(\frac{s_h(h_{t+1})V_{t+1}}{s(h_{t+1})(c_{t+1})^{-\theta}} + w + \frac{1}{\gamma}i_{t+1}^{1-\alpha}\right)^{-1}\right)^{-\frac{1}{\alpha}}}{i_t}. \tag{8b}$$

These errors are dimension free quantities. Equation (8a) expresses the optimization error as a fraction of current consumption. An error of $e_{1,t} = 10^{-3}$, for instance, means that the household makes a \$1 mistake for each \$1000 spent, cf. Aruoba, Fernandez-Villaverde, and Rubio-Ramirez (2006). These errors are expressed in units of base 10-logarithm which means that -4 is an error of 0.0001.

---

[18]Also observe, by construction, there is only one occasionally binding constraint in our model. This would be different in a situation with durable consumption goods as in Hintermaier and Koeniger (2010). As ENDGM is a very efficient way in dealing with occasionally binding constraints such an alternative model may further improve the relative performance of ENDGM.

4.2 Finite Horizon

We iterate over $T = 100$ time periods. Computational speed of the respective algorithms is measured in seconds. To compare all three methods in terms of accuracy we simulate 100 life-cycles profiles and evaluate Euler equation errors accordingly. Initial assets a_0 are set in the range $[10, 100]$ whereas initial human capital h_0 is drawn from the range $[50, 100]$. For each simulation and each age we compute $e_{1,t}$ and $e_{2,t}$ from equation (8).[19] We next compute average and maximum errors across all simulations and ages. These are provided in Table 1. Both are of similar magnitudes across algorithms. To evaluate the relative performance of the different algorithms, we can therefore further concentrate on comparison of speed only.

Table 1 shows computing times for EXOGM, ENDGM and HYBGM for different numbers of gridpoints. We report absolute computing time as well as relative speed, i.e., relative to the ENDGM method. As our model is (on purpose) very stylized, absolute computing times are low across all models. However, relative speed is the relevant measuring rod because absolute speed scales up in the complexity of the model's specification, e.g., in fully stochastic models, applications in general equilibrium or estimation of models with structural methods. With regard to this relative comparison, observe from Panel (a) of Figure 7 that EXOGM is outperformed by both ENDGM and HYBGM.

Panel (b) of Figure 7 shows that ENDGM has a relative advantage in comparison to HYBGM in solving the model with a relatively small number of gridpoints. At a grid size of 25^2, ENDGM is about 1.7 times faster than HYBGM. For solving the model with a higher number of gridpoints, however, HYBGM is advantageous. At a grid size of 300^2, HYBGM is about 1.3 times faster than ENDGM. In our setting the break-even point between both algorithms is at a number of 180^2 gridpoints and a computing time of $8.8s$. As can be seen from Table 1, for a standard choice of 25 to 50 gridpoints in each dimension, ENDGM is $\frac{0.624}{0.437} \approx 1.4$ to $\frac{0.156}{0.094} \approx 1.7$ times faster than HYBGM and $\frac{0.982}{0.437} \approx 2.3$ to $\frac{0.234}{0.094} \approx 2.5$ times faster than EXOGM.

4.3 Infinite horizon

To compare the algorithms in the infinite horizon setting, we make the same initial guesses for derivatives V_{0_a} and V_{0_h} and iterate until convergence on policy functions subject to convergence criterion $\varepsilon = 10^{-6}$ in terms of the maximum absolute distance of policy functions. In the infinite horizon setting, speed of ENDGM can be increased if the Delaunay Triangulation is not constructed every iteration. Instead, we hold the triangulation pattern fixed after a certain number of iterations—50 in our case. We call this modification of the algorithm "Approximate Delaunay". Figure 8 illustrates this. Panel (a) of the figure shows how endogenous grid-points move in the (a, h) space from

---

[19]Euler equation errors are not computed if the borrowing constraint is binding.

Table 1: Finite Horizon Model: Performance Results

Number of Gridpoints for (a,h)	Speed		Euler Equation Error	
	Seconds	Relative to ENDGM	Maximum for c ; i	Average for c ; i
ENDGM				
$(25, 25)$	0.094	-	-2.56; -2.17	-3.70; -2.94
$(50, 50)$	0.437	-	-2.92; -2.60	-4.36; -3.53
$(100, 100)$	2.090	-	-3.37; -3.07	-4.91; -4.05
$(200, 200)$	11.278	-	-3.84; -3.47	-5.44; -4.51
HYBGM				
$(25, 25)$	0.156	1.7	-2.62; -2.25	-3.88; -2.90
$(50, 50)$	0.624	1.4	-2.99; -2.71	-4.43; -3.52
$(100, 100)$	2.496	1.2	-3.43; -3.10	-5.00; -3.98
$(200, 200)$	10.218	0.9	-4.16; -3.52	-5.54; -4.45
EXOGM				
$(25, 25)$	0.234	2.5	-2.60; -2.24	-3.89; -2.90
$(50, 50)$	0.982	2.3	-2.95; -2.71	-4.42; -3.52
$(100, 100)$	3.868	1.9	-3.42; -3.10	-4.99; -3.98
$(200, 200)$	15.663	1.4	-4.18; -3.52	-5.54; -4.45

Notes: Computing time for $T = 100$ and resulting maximum and average Euler equation errors. Computing time is reported in seconds and absolute errors in units of base-10 logarithms.

Figure 7: Finite Horizon Model: Speed
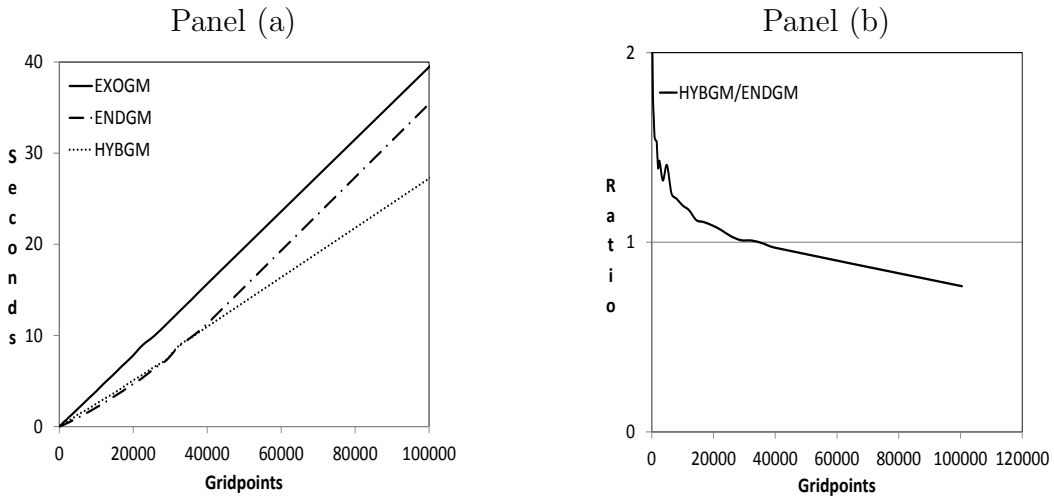
Notes: Panel (a): Computing time as a function of gridpoints in seconds (with equally many gridpoints in both dimensions). Solid line: computing time of EXOGM; dotted line: computing time of HYBGM; dashed-dotted line: computing time of ENDGM. Panel (b): Ratio of computing time of ENDGM to HYBGM as a function of gridpoints (with equally many gridpoints in both dimensions).

one iteration to the next. Panel (b) shows the new triangulation, holding constant the respective triangles from Panel (a). However, this triangulation is not Delaunay because edge P_1-P_2 becomes illegal.

In "Approximate Delaunay" it is necessary to ensure that the endogenously computed gridpoints form a convex hull. This might be violated without further adjustments. For example, in our illustration in panel (b) of Figure 8 violation of convexity would occur if point P_3 is shifted even further to the right. In such cases we redo the entire Delaunay tessellation.

Figure 8: Infinite Horizon Model: Approximate Delaunay
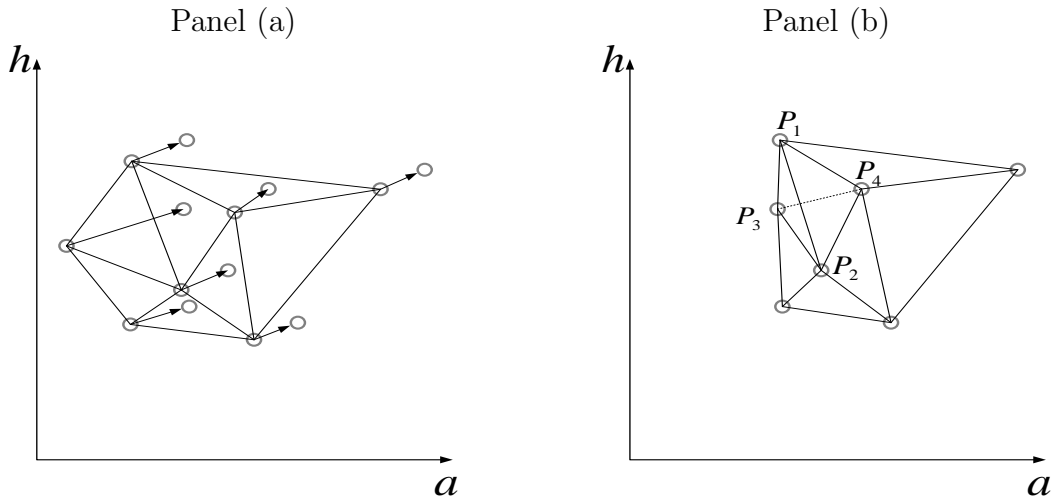
Notes: Panel (a): In each iteration of ENDGM the gridpoints are relocated. Distance and direction of this movement is different for each gridpoint. Panel (b): The resulting grid might not be Delaunay - Edge between P_1 and P_2 becomes illegal and must be flipped to P_3 and P_4. *Approximate Delaunay* keeps the old triangulation in order to save computing time, accepting a less accurate interpolation.

To compute Euler equation errors we simulate the model for various different initial conditions of financial assets and health capital over 50 periods. We set initial assets a_0 in the range of $[100, 400]$ and the health capital stock in the range of $[40, 80]$. We compute $e_{1,t}$ and $e_{2,t}$ from equation (8) for the first 50 periods. Average and maximum errors are provided in Table 2.

As in the finite horizon setting, average Euler equation errors are of similar magnitudes across algorithms—which we also achieve by appropriate settings of the respective numerical routines—so that we can again further concentrate on a comparison of speed only.[20]

We find that ENDGM is the fastest method for all numbers of gridpoints considered. In this respect our findings differ from the finite horizon version of the model in which the

---

[20]The maximum Euler equation errors are considerably higher for EXOGM. They occur in the simulations just before the depletion of all financial assets. This is due to the fact that we do not determine explicitly the region where the borrowing constraint becomes binding and accordingly have no gridpoints located there.

speed advantage of ENDGM relative to HYBGM was found to depend on the number of gridpoints. The reason for this difference is the use of the variant "Approximate Delaunay" in the infinite horizon model, as described above. As in the finite horizon model, the comparative advantage of ENDGM decreases in the number of gridpoints. Both, ENDGM and HYBGM, again clearly dominate EXOGM. For a standard choice of 25 to 50 gridpoints in each dimension, ENDGM is $\frac{1.153}{0.624} \approx 2.4$ to $\frac{0.390}{0.156} \approx 2.5$ times faster than HYBGM and $\frac{2.527}{0.624} \approx 4.0$ to $\frac{0.640}{0.156} \approx 4.1$ times faster than EXOGM, cf. Table 2.

Table 2: Infinite Horizon Model: Performance Results

Number of Gridpoints for (a, h)	Speed		Euler Equation Error	
	Seconds	Relative to ENDGM	Maximum for c ; i	Average for c ; i
ENDGM				
$(25, 25)$	0.156	-	-2.09; -2.10	-2.87; -2.87
$(50, 50)$	0.624	-	-2.37; -2.40	-3.61; -3.52
$(100, 100)$	2.792	-	-2.84; -2.91	-4.17; -4.15
$(200, 200)$	15.194	-	-3.14; -3.24	-4.80; -4.66
HYBGM				
$(25, 25)$	0.390	2.5	-2.16; -2.10	-2.92; -2.97
$(50, 50)$	1.513	2.4	-2.49; -2.58	-3.73; -3.66
$(100, 100)$	6.115	2.2	-2.91; -2.98	-4.29; -4.23
$(200, 200)$	27.175	1.8	-3.19; -3.29	-4.91; -4.80
EXOGM				
$(25, 25)$	0.640	4.1	-1.53; -1.64	-2.80; -2.87
$(50, 50)$	2.527	4.0	-1.81; -1.92	-4.17; -4.52
$(100, 100)$	10.109	3.6	-2.44; $-2, 55$	-4.17; -4.15
$(200, 200)$	41.371	2.7	-2.40; -2.51	-4.69; -4.66

Notes: Computing time to convergence of policy functions (criterion $\varepsilon = 10^{-6}$) and resulting maximum and average Euler equation errors. Computing time is reported in seconds and absolute errors in units of base-10 logarithms.

Figure 9: Infinite Horizon Model: Speed
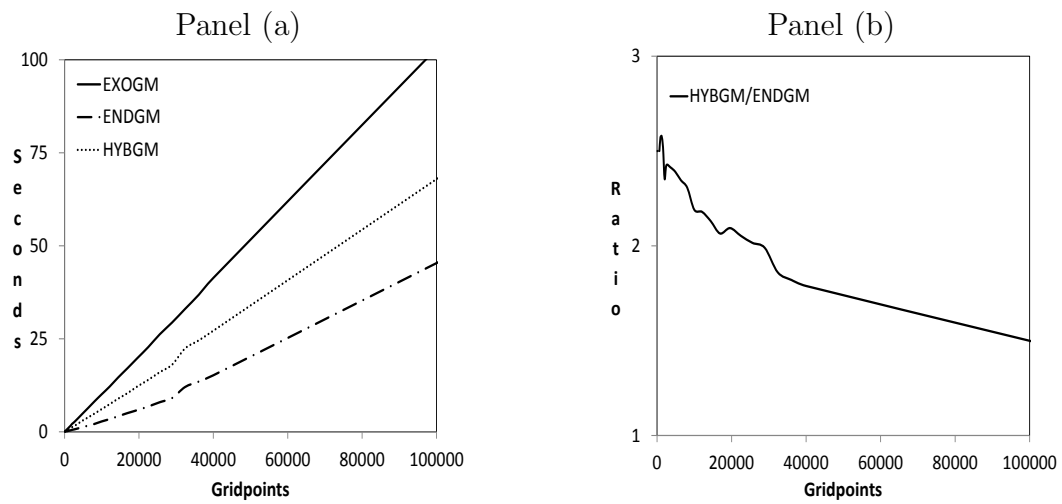
Panel (a)

Panel (b)

Notes: Panel (a): Computing time to convergence of policy functions (criterion $\varepsilon = 10^{-6}$) as a function of gridpoints (with equally many gridpoints in both dimensions). Solid line: computing time of EXOGM; dotted line: computing time of HYBGM; dashed-dotted line: computing time of ENDGM. Panel (b): Ratio of computing time to convergence of ENDGM and HYBGM as a function of gridpoints (with equally many gridpoints in both dimensions).

5 Conclusion

We compare three numerical methods—the standard exogenous grid method (EXOGM), Carroll's method of endogenous gridpoints (ENDGM), cf. Carroll (2006), and a hybrid method (HYBGM), cf. also Hintermaier and Koeniger (2010)—to solve dynamic models with two continuous state variables and occasionally binding borrowing constraints. To illustrate and to evaluate these methods we develop a life-cycle consumption-savings model with endogenous human capital formation. Evaluation of methods is based on speed and accuracy in both a finite and an infinite horizon setting. We show that applying ENDGM gives rise to irregular grids. We emphasize that this leads to a trade-off: On the one hand, closed form solutions in ENDGM greatly simplify the problem relative to standard EXOGM. On the other hand, interpolation becomes more costly due to the irregularity of grids. We apply Delaunay methods to interpolate on these irregular grids.

Despite this more complex interpolation, we find that ENDGM outperforms EXOGM in both the finite as well as the infinite horizon version of the model. In the infinite horizon model, ENDGM also always dominates HYBGM. For a standard choice of 25 to 50 gridpoints in each dimension, ENDGM is 2.4 to 2.5 times faster than HYBGM and 4.0 to 4.1 times faster than EXOGM. As the number of gridpoints increases, interpolation on irregular grids becomes increasingly costly to the effect that the relative speed advantage of ENDGM decreases. This becomes more apparent in the finite horizon model. Here, ENDGM dominates HYBGM for small to medium sized problems whereas HYBGM dominates for a large number of gridpoints. For a standard choice of 25 to 50 gridpoints in each dimension, ENDGM is 1.4 to 1.7 times faster than HYBGM and 2.3 to 2.5 times faster than EXOGM.

Two additional remarks on ENDGM and HYBGM are in order. First, within the class of problems solvable with first-order methods, neither of the two is a general method. Both are applicable only to specific problems at hand. This requires restrictions on the model's specification and on functional forms. Second, as HYBGM uses analytical solutions in only one dimension and standard numerical methods in others, its relative advantage can be expected to decrease in the dimensionality of the problem. For example, in a three dimensional problem, as HYBGM can only use analytical solutions in one dimension, HYBGM requires to solve a two-dimensional problem numerically. On the other hand, however, complexity of interpolation in ENDGM will also increase. As we restrict attention to two dimensional problems in this paper, we cannot address how this trade-off is ultimately resolved. We leave such extensions to higher dimensions for future research.

A   Derivation of FOC

The dynamic version of the household problem reads as

$$V_t(a_t, h_t) = \max_{c_t, i_t, a_{t+1}, h_{t+1}} \{u(c_t) + \beta s(h_{t+1}) V_{t+1}(a_{t+1}, h_{t+1})\}$$

subject to

$$a_{t+1} = R(a_t + wh_t - c_t - i_t)$$
$$h_{t+1} = (1 - \delta)(h_t + f(i_t))$$
$$a_{t+1} \geq 0.$$

Assigning multiplier μ to the borrowing constraint, the two first order conditions with respect to c_t and i_t are:

$$\frac{\partial V_t(a_t, h_t)}{\partial c_t} = u_c - \beta s(h_{t+1}) V_{t+1_a} R - R\mu \overset{!}{=} 0 \Leftrightarrow u_c - \beta s(h_{t+1}) R V_{t+1_a} = R\mu, \tag{9}$$

$$\frac{\partial V_t(a_t, h_t)}{\partial i_t} = s_h(h_{t+1})(1 - \delta) f_i \beta V_{t+1} + s(h_{t+1}) \beta (V_{t+1_a}(-R) + V_{t+1_h}(1 - \delta) f_i) - R\mu \overset{!}{=} 0$$

$$\Leftrightarrow s_h(h_{t+1})(1 - \delta) f_i \beta V_{t+1} + s(h_{t+1}) \beta (V_{t+1_a}(-R) + V_{t+1_h}(1 - \delta) f_i) = R\mu \tag{10}$$

and $a_{t+1} \geq 0$, $\mu \geq 0$ and $a_{t+1}\mu = 0$.

In order to compute optimal policies we need to distinguish two cases.

Case 1: Interior Solution

In the first case the borrowing constraint is not binding so that $\mu = 0$. This reduces the system of equations to

$$u_c - \beta s(h_{t+1}) R V_{t+1_a} = 0$$
$$s_h(h_{t+1})(1 - \delta) f_i \beta V_{t+1} + s(h_{t+1}) \beta (V_{t+1_a}(-R) + V_{t+1_h}(1 - \delta) f_i) = 0.$$

Rearranging gives

$$u_c = \beta s(h_{t+1}) V_{t+1_a} R$$
$$f_i = \frac{R}{(1 - \delta)} \frac{s(h_{t+1}) V_{t+1_a}}{s_h(h_{t+1}) V_{t+1} + s(h_{t+1}) V_{t+1_h}}.$$

Case 2: Corner Solution—Binding Borrowing Constraint

In the second case the borrowing constraint is binding so that $a' = 0$ and $\mu > 0$. From (9) and (10) it then follows that

$$u_c = s_h (h_{t+1}) \beta V_{t+1} + s (h_{t+1}) \beta V_{t+1_h} (1 - \delta) f_i \qquad (11)$$

and

$$u_c = \beta (1 - \delta) f_i (s_h (h_{t+1}) V_{t+1} + s (h_{t+1}) V_{t+1_h})$$

$$a_{t+1} = 0 \Leftrightarrow c_t = a_t + w h_t - i_t.$$

Making use of our assumptions on functional forms, equation (11) reduces in EXOGM and HYBGM to

$$(a_t + w h_t - i_t)^{-\theta} - \frac{1}{\left((1 - \delta) \left(h_t + \frac{\gamma}{\alpha} i_t^\alpha\right)\right)^2} V_{t+1} \left[0, (1 - \delta) \left(h_t + \frac{\gamma}{\alpha} i_t^\alpha\right)\right] \beta (1 - \delta) \gamma i_t^{-(1-\alpha)}$$

$$- \left(1 - \frac{1}{(1 - \delta) \left(h_t + \frac{\gamma}{\alpha} i_t^\alpha\right)}\right) V_{t+1_h} \left(0, (1 - \delta) \left(h_t + \frac{\gamma}{\alpha} i_t^\alpha\right)\right) \beta (1 - \delta) \gamma i_t^{-(1-\alpha)} = 0$$

and in ENDGM to

$$\left(a_t + w \left(\frac{h_{t+1}}{1 - \delta} - \frac{1}{\gamma} i_t^{1-\alpha}\right) - i_t\right)^{-\theta} - \frac{1}{(h_{t+1})^2} \beta V_{t+1} (0, h_{t+1}) (1 - \delta) \gamma i_t^{-(1-\alpha)}$$

$$- \left(1 - \frac{1}{1 + h_{t+1}}\right) \beta V_{t+1_h} (0, h_{t+1}) (1 - \delta) \gamma i_t^{-(1-\alpha)} = 0.$$

Observe that this equation is not linear in i_t. We therefore need to use a numerical routine in the region where the borrowing constraint is binding also for ENDGM, cf. our discussion in the main text in Subsection 3.2.

In both cases—i.e., for interior solutions and for binding borrowing constraints—the envelope conditions are

$$\frac{\partial V_t (a_t, h_t)}{\partial a_t} \equiv V_{t_a} = \beta V_{t+1_a} R + R\mu = u_c$$

$$\frac{\partial V_t (a_t, h_t)}{\partial h_t} \equiv V_{t_h}$$

$$= \beta s_h (h_{t+1}) V_{t+1}(a_{t+1}, h_{t+1}) (1 - \delta) + \beta s (h_{t+1}) V_{t_a}(a_{t+1}, h_{t+1}) w R +$$

$$\beta s (h_{t+1}) V_{t+1_h}(a_{t+1}, h_{t+1}) (1 - \delta) + R\mu$$

$$= \left(w + \frac{1}{f_i}\right) u_c.$$

References

Aruoba, S. B., J. Fernandez-Villaverde, and J. F. Rubio-Ramirez (2006). Comparing solution methods for dynamic equilibrium economies. *Journal of Economic Dynamics and Control 30*(12), 2477–2508.

Baker, T. J. (1999). Delaunay - voronoi methods. In *Handbook of Grid Generation*, Chapter 16. CRC Press.

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

Ben-Porath, Y. (1967). The production of human capital and the life cycle of earnings. *Journal of Political Economy 75*, 352.

Broer, T., M. Kapicka, and P. Klein (2013). Consumption risk sharing with private information when earnings are persistent. Technical report, Simon Fraser University.

Brumm, J. and M. Grill (2014). Computing equilibria in dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control 38*, 142 – 160.

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

de Berg, M., O. Cheong, M. van Kreveld, and M. Overmars (2008, April). *Computational Geometry: Algorithms and Applications* (3rd ed.). Springer.

Delaunay, B. N. (1934). Sur la sphère vide. *Bulletin of Academy of Sciences of the USSR* (6), 793–800.

Devillers, O., S. Pion, and M. Teillaud (2001). Walking in a triangulation. In *Proceedings of the seventeenth annual symposium on Computational geometry*, SCG '01, New York, NY, USA, pp. 106–114. ACM.

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

Hall, R. E. and C. I. Jones (2007, 02). The value of life and the rise in health spending. *The Quarterly Journal of Economics 122*(1), 39–72.

Hintermaier, T. and W. Koeniger (2010). The method of endogenous gridpoints with occasionally binding constraints among endogenous variables. *Journal of Economic Dynamics and Control 34*(10), 2074–2088.

Iskhakov, F., T. Jørgensen, J. Rust, and B. Schjerning (2014, December). Estimating discrete-continuous choice models: Endogenous grid method with taste shocks. Technical report.

Joe, B. (1991). Geompack: A software package for the generation of meshes using geometric algorithms. *Advances in Engineering Software and Workstations 13*, 325–331.

Judd, K. L. (1992). Projection methods for solving aggregate growth models. *Journal of Economic Theory 58*(2), 410–452.

Judd, K. L. (1998). *Numerical Methods in Economics* (1 ed.), Volume 1. The MIT Press.

Krueger, D. and A. Ludwig (2007). On the consequences of demographic change for rates of returns to capital, and the distribution of wealth and welfare. *Journal of Monetary Economics 54*(1), 49–87.

Miranda, M. J. and P. L. Fackler (2004). *Applied Computational Economics and Finance* (1 ed.), Volume 1. The MIT Press.

Press, W., S. Teukolsky, W. Vetterling, and B. Flannery (2007). *Numerical Recipes 3rd Edition: The Art of Scientific Computing.* Numerical Recipes: The Art of Scientific Computing. Cambridge University Press.

Press, W., S. Teukolsky, W. Vetterling, B. Flannery, and M. Metcalf (1996). *Numerical Recipes in Fortran 90: Volume 2, Volume 2 of Fortran Numerical Recipes: The Art of Parallel Scientific Computing.* Fortran Numerical Recipes , Vol 2. Cambridge University Press.

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

Stokey, N. L. and R. E. Lucas (1989). *Recursive Methods in Economic Dynamics.* Harvard University Press.