MAX-PLANCK-INSTITUT FÜR SOZIALRECHT UND SOZIALPOLITIK
MAX PLANCK INSTITUTE FOR SOCIAL LAW AND SOCIAL POLICY

mea Munich Center for the Economics of Aging

Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods

Alexander Ludwig, Matthias Schön

09-2013

MEA DISCUSSION PAPERS

mea – Amalienstr. 33_D-80799 Munich_Phone +49 89 38602-355_Fax +49 89 38602-390_**www.mea.mpisoc.mpg.de**

Alte Nummerierung: 274-13

Updated version; first version: October 6, 2014

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

Alexander Ludwig† Matthias Schön‡

April 18, 2016

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 by using a specific model with two endogenous state variables. 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, Method of Endogenous Gridpoints, Delaunay Interpolation

---

*We thank Johannes Brumm, Christopher Carroll, Thomas Jørgensen, Michael Reiter and seminar participants at University of Cologne, the 2012 CEF and the Cologne Macroeconomic Workshop 2012 for helpful comments. Alex Ludwig gratefully acknowledges research support from the Research Center SAFE, funded by the State of Hessen initiative for research LOEWE and financial support by the German National Research Foundation under SPP 1578. Matthias Schön gratefully acknowledges financial support by the State of North Rhine-Westfalia.

†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 in discrete time are workhorse models in Economics. However, most of these models do not have an analytic closed form solution and therefore have to be solved 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 based on an exogenous grid of state variables (EXOGM). An important contribution to this literature is Carroll (2006) who introduces the method of endogenous gridpoints (ENDGM). In comparison to 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. We present and evaluate two alternatives to EXOGM by use of a specific economic model with two endogenous state variables. The first alternative is a full-blown ENDGM which avoids rootfinding procedures throughout but requires a rather complex interpolation method. As a second method we investigate a hybrid method (HYBGM) which stands in between EXOGM and ENDGM by using rootfinding procedures in one dimension combined with standard fast interpolation methods.

To understand how the tradeoff between these alternatives arises in higher dimensions, first focus on a simple consumption-savings problem in on one dimension as in Carroll (2006). In a standard exogenous grid method (EXOGM) one solves in each 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 policy function. To summarize, the mapping in EXOGM is $a \to c \to a'$ whereby this mapping requires, among other numerical operations, solving a non-linear equation and interpolation. Also observe that, for some regular grid $\mathcal{G}^a$, 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 is to reverse the mapping, i.e., $a' \to c \to a$. Instead of working on an exogenous grid for a, this is achieved by defining a grid on next period's assets, $\mathcal{G}^{a'}$. Depending on the nature of the problem it is then possible to solve for c analytically. This is the crucial step: The speed advantage of ENDGM relative to EXOGM is achieved because the mapping $a' \to c$ has a closed form solution. For given contemporaneous variables c and next period's a' one can endogenously compute today's endogenous state a. Given the regular grid $\mathcal{G}^{a'}$, 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. The resulting state grid is generally not rectangular because 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. Corresponding to the one-dimensional problem the mapping in EXOGM is $(a, h) \rightarrow (c, i) \rightarrow (a', h')$ which requires solution of a system of two non-linear equations. In ENDGM, the mapping is reversed, i.e., $(a', h') \rightarrow (c, i) \rightarrow (a, h)$. As for the one dimensional problem, this mapping may have a closed form solution but the endogenous grid formed of a, h is irregular. In subsequent iterations one has to interpolate on such an irregular grid. This irregularity severely complicates location of points for interpolation in higher dimensions.

Hence, 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) which originates from the field of Geometry and was only recently introduced to Economics by Brumm and Grill (2014).[1] Our contribution is to investigate its performance in combination with ENDGM.

Our in-between method HYBGM uses exogenous gridpoints in one dimension and endogenous gridpoints in the other.[2] Consequently, the endogenously computed grid is only irregular in one dimension whereas it is regular in the other, 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 in some dimensions vis-à-vis analytical solutions in all dimensions but a more complex interpolation. Another aspect is that HYBGM is easier to implement.

To analyze and to compare these methods we develop a simple consumption savings model with endogenous human capital. Our specification is such that the 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.

[1] For another application of the method in Economics see Broer, Kapicka, and Klein (2013).

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

Our main finding is that HYBGM and ENDGM both dominate EXOGM: they are substantially faster. The relative speed advantage of ENDGM decreases in the number of grid points because the complex interpolation becomes increasingly costly whereby the speed of solvers used in EXOGM and HYBGM improves when the density of the grid increases due to improved initializations. In light of the positive speed results of ENDGM reported for one-dimensional problems in Jørgensen (2013) and several others, the finding that HYBGM dominates EXOGM is not surprising because it preserves the complexity of EXOGM in the one dimension and the speed advantage of the endogenous grid method in the other. That ENDGM dominates EXOGM is a quantitative finding because the resolution of the trade-off between the more complex interpolation method and the simpler solution of the system of non-linear equations is a priori not clear. Furthermore, we also find that ENDGM dominates HYBGM in our infinite horizon application. There we use an "Approximate Delaunay" method which spares out a large part of the computational burden of the interpolation in ENDGM. In our finite horizon application, which uses "Pure Delaunay", 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.[3]

Related work by Krueger and Ludwig (2007) in heterogenous agent models and Barillas and Fernandez-Villaverde (2007) in the neoclassical growth model extends ENDGM to problems with two control variables but just one endogenous state variable. Hintermaier and Koeniger (2010) numerically solve a durable goods model with two endogenous state variables by applying a hybrid method similar to our HYBGM. The key difference to our version of HYBGM is that we solve the non-linear equation with a univariate solver whereas Hintermaier and Koeniger (2010) use interpolation techniques that are generically less accurate. Other related literature evaluates the performance of ENDGM in the context of estimating structural models (Jørgensen 2013) and/or 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. (2015).

Most closely related to our contribution is White (2015) who builds on the earlier working paper version of our paper (Ludwig and Schön 2013). White (2015) applies a slight modification of our economic model to evaluate the performance of an alternative interpolation method to Delaunay interpolation which superimposes more structure. To understand this, notice that Delaunay interpolation comes in three steps. The first is a triangulation of the state space. The second is the location of the triangle in which a specific interpolation point is located which is a simple search. The third is the in-

[3]We also discuss limitations of ENDGM and HYBGM which are both only applicable to specific problems at hand. A thorough treatment of the formal conditions that make ENDGM applicable is contained in Iskhakov (2015) and White (2015). We provide simple examples for situations where such conditions to not hold.

terpolation itself. The costly step in the overall procedure is the triangulation. White (2015)'s method circumvents this step by preserving the ordering of the exogenous and endogenous grid points so that no new objects have to be created for the purpose of interpolation. The state space is subdivided into irregular quadrilateral sectors on which standard bilinear interpolation is possible. An important question is how the Delaunay method compares with White (2015)'s approach. Given that White (2015) avoids the costly triangulation step, the method is faster than Delaunay interpolation for a given number of gridpoints. However, because triangles cover smaller areas in the state space than quadrilateral sectors, we conjecture that accuracy is lower and that accuracy decreases more strongly in the irregularity of the endogenously constructed grid. For strong irregularity standard methods to locate a grid point may also not work.[4] We leave an investigation of these aspects for future research.

Our analysis proceeds as follows. Section 2 presents the simple life-cycle model with endogenous assets and human capital 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 The Model

We develop a consumption and savings model which allows us to illustrate and to compare three approaches to solve dynamic models with two endogenous state variables 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, stripping it such that it is a non-generate problem in two continuous state variables which makes the endogenous grid method applicable. This requires specific assumptions on functional forms and arguments of the respective functions which we make explicit in various places. Although many applications of dynamic structural models are cast in stochastic environments, we focus attention on the main trade-off by ignoring any stochasticity beyond the degenerate risk of survival. Of course, the underlying trade-off between solution methods will also hold in more complex problems. However, the exact resolution of this trade-off depends on the specific application at hand.

[4]The standard method that we and White (2015) apply is the so-called visibility walk. It is known that it may fall into a cycle in non-Delaunay triangulations, cf. Devillers et al. (2001).

2.1 A Simple Human Capital Model

A risk averse agent with maximum time horizon T, $T = \infty$ possible, derives utility from consumption, $c_t \geq 0$ in each period, with standard additive separable life time utility $U = \sum_{t=1}^{T} \beta^{t-1} \pi_t u(c_t)$, where $\beta \in (0,1)$ is the discount factor and $\pi_t = \prod_{i=2}^{t} \psi(h_i)$, for $t > 1$, $\pi_1 = 1$ is the unconditional probability to survive from period 1 to t.[5] Dependency of survival on health is crucial to avoid that the decision problem collapses to one in a single state variable, see Remark 1 below.

The instantaneous utility function $u(c_t)$ as well as the probability to survive to the next period $\psi(h_t)$ are assumed to be strictly increasing and concave in their respective arguments ($u_c > 0, u_{cc} < 0, \psi_h > 0, \psi_{hh} < 0$). We also assume that the utility function satisfies the Inada conditions, i.e., $\lim_{c_t \to 0} u_c(c_t) = \infty$ and $\lim_{c_t \to \infty} u_c(c_t) = 0$ so that $c_t > 0$ will be optimal in all t and there is no satiation. 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 $R = 1 + r$ 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 $f(i)$. We assume that $f_i > 0$, $f_{ii} < 0$ and that $f(i)$ satisfies the Inada conditions, i.e., $\lim_{i_t \to 0} f_i = \infty$ and $\lim_{i_t \to \infty} f_i = 0$. These conditions insure that $i_t > 0$ and, together with the assumptions on the utility function, it will always be optimal to invest in both assets so that the decision problem never collapses to a one dimensional one. The human capital accumulation equation is accordingly given by

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

where h_0 is given.

Financial markets are imperfect (there is no annuitization of wealth) and households are not allowed to hold negative financial assets. The dynamic budget constraint writes as

$$a_{t+1} = R(a_t + wh_t - c_t - i_t) \geq 0, \qquad (2)$$

where a_0 is given.

Remark 1 *Assume that human (or health) capital does not affect the probability to survive so that human capital only enters the budget constraints (1) and (2). Then the setup*

[5] The standard formulation of contingency of the survival rate on health capital—which depreciates over time, see below—as we use it here has recently been criticized by Dalgaard and Strulik (2014) who point out that a more appropriate way of modeling decay is the notion of appreciating deficits rather than decreasing health capital.

collapses to a situation with a single endogenous state variable. This can be seen by combining the resource constraints to get

$$a_{t+1} + h_{t+1} = a_t R + h_t(1 + w - \delta) + g(i_t) - c_t$$

where $g(i_t) \equiv (1-\delta) f(i_t) - R i_t$*. Next define total wealth as* $\mathrm{w}_t \equiv a_t + w_t$ *and let* $\alpha_t = \frac{a_t}{\mathrm{w}_t}$ *be the period* t *share of assets in total wealth. Using these definitions rewrite the resource constraint further as*

$$\mathrm{w}_{t+1} = \mathrm{w}_t \left(1 + w - \delta + \alpha_t \left(r - (w - \delta)\right)\right) + g(i_t) - c_t$$

so that the model reduces to a standard portfolio choice problem in the endogenous continuous state variable w_t *and the three continuous control variables* α_t, c_t, i_t*.*

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

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

subject to the constraints

$$\begin{aligned} a_{t+1} &= R\left(a_t + w h_t - c_t - i_t\right) \\ h_{t+1} &= (1-\delta)\left(h_t + f\left(i_t\right)\right) \\ a_{t+1} &\geq 0 \\ h_{t+1} &> 0. \end{aligned} \tag{3}$$

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 $\theta > 0$:

$$u\left(c_t\right) = \begin{cases} \frac{c_t^{1-\theta}}{1-\theta} & \text{if } \theta \neq 1 \\ \ln(c_t) & \text{if } \theta = 1. \end{cases}$$

The human capital production function is $f\left(i_t\right) = \frac{1}{\xi} i_t^{\xi}$ for curvature parameter $\xi \in (0,1)$. As to the functional form of the per-period survival probability we follow Hall and Jones (2007) and assume that $\psi\left(h_t\right) = 1 - \phi \frac{1}{1+h_t}$, 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.[6]

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

$$c_t^{-\theta} = \beta R \left(1 - \phi \frac{1}{1+h_{t+1}}\right) V_{t+1_a}\left(a_{t+1}, h_{t+1}\right) \tag{4a}$$

$$i_t^{-(1-\xi)} = \frac{R}{(1-\delta)} \frac{V_{t+1_a}\left(a_{t+1}, h_{t+1}\right)}{\frac{\phi}{(1+h_{t+1}-\phi)(1+h_{t+1})} V_{t+1}\left(a_{t+1}, h_{t+1}\right) + V_{t+1_h}\left(a_{t+1}, h_{t+1}\right)} \tag{4b}$$

$$a_{t+1} = R\left(a_t + w h_t - c_t - i_t\right) \tag{4c}$$

$$h_{t+1} = (1-\delta)\left(h_t + f\left(i_t\right)\right) \tag{4d}$$

$$a_{t+1} \geq 0. \tag{4e}$$

V_{t_a} and V_{t_h} are derivatives of the value function with respect to financial assets and human capital, respectively. Notice that constraint (3) can be dropped because of the lower Inada condition of the human capital investment function $f(i)$. The envelope conditions are:

$$V_{t_a}\left(a_t, h_t\right) = u_c = c_t^{-\theta} \tag{5a}$$

$$V_{t_h}\left(a_t, h_t\right) = \left(w + \frac{1}{f_i}\right) u_c = \left(w + \frac{1}{i_t^{-(1-\xi)}}\right) c_t^{-\theta}. \tag{5b}$$

Using (4a) together with (5a) gives the standard Euler equation of consumption.[7]

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 (4) using (5).

[6] We checked ex-post if value functions are globally concave which they are for the parameter space considered here. A crucial parameter is ξ as it governs the curvature of the human capital production function. If we were to choose a higher degree of curvature (lower ξ) than non-concavities may arise. These results are available upon request. Also see Fella (2014) and Iskhakov et al. (2015) for sophisticated methods to deal with non-convexities.

[7] For derivation of (4) and the Envelope conditions see Appendix A.

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 value of life. We set the time preference rate to $\rho = 0.04$ so that the discount factor $\beta \equiv \frac{1}{1+\rho}$ is approximately 0.96. 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 $\xi = 0.35$. 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 (4a) and (4b) 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 ($t = T, T-1, \dots, 0$). In the infinite horizon implementation the iteration continues until convergence on policy functions (=time iteration).

Differences between methods arise because of different solution procedures to the multi-dimensional nonlinear equation system (4) 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) solves the system of equations (4) analytically. It is accompanied by Delaunay interpolation. The third algorithm (HYBGM) combines the former two, i.e., it applies the method of endogenous gridpoints 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 (4) 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 (4). In our application we use a Quasi-Newton method, more specifically Broyden (1965)'s method, cf. Press et al. (1996).

The implementation steps of EXOGM are as follows:[8]

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 $T+1$[9] and income is completely consumed for all $\left(a^k, h^j\right) \in \mathcal{G}^{a,h}$:

$$
\begin{aligned}
c_T(\cdot,\cdot) &= a_T^k + w h_T^j \\
i_T(\cdot,\cdot) &= 0.
\end{aligned}
$$

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) Solve (using interpolation methods, see below) the two-dimensional equation system

$$
\begin{aligned}
\left(c_t^{k,j}\right)^{-\theta} &= \beta R \left(1 - \phi \frac{1}{1 + (1-\delta)\left(h_t^j + \frac{1}{\xi}\left(i_t^{k,j}\right)^{\xi}\right)}\right) \\
& V_{t+1_a}\left(\overbrace{R\left(a_t^k + w h_t^j - c_t^{k,j} - i_t^{k,j}\right)}^{a_{t+1}^{k,j}}, \overbrace{(1-\delta)\left(h_t^j + \frac{1}{\xi}\left(i_t^{k,j}\right)^{\xi}\right)}^{h_{t+1}^{k,j}}\right) \\
\left(i_t^{k,j}\right)^{-(1-\xi)} &= \frac{R}{(1-\delta)} \frac{V_{t+1_a}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right)}{\frac{\phi}{\left(1+h_{t+1}^{k,j}-\phi\right)\left(1+h_{t+1}^{k,j}\right)} V_{t+1}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right) + V_{t+1_h}\left(a_{t+1}^{k,j}, h_{t+1}^{k,j}\right)}
\end{aligned}
$$

   for $c_t^{k,j}$ and $i_t^{k,j}$ using Broyden's method.

[8] Our descriptions of all algorithms leave out the numerical characterization and storage of the value function (and its derivatives) as side products of the respective algorithm.

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

(b) If $c_t^{k,j}+i_t^{k,j} > a_t^k+wh_t^j$ (binding borrowing constraint) recompute $i_t^{k,j}$ by solving

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

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

Steps 3a and 3b require interpolation on c_{t+1} and i_{t+1}, and updating of V_{t+1_a}, V_{t+1_h} using the envelope conditions (5).[10]

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.[11] 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.[12]

Interpolation on a Rectilinear Grid Steps 3a and 3b require interpolation on c_{t+1}, i_{t+1} because, in general, $(a_{t+1}^{k,j}, h_{t+1}^{k,j}) \notin \mathcal{G}^{a,h}$. We apply standard bilinear interpolation, cf., e.g., Press et al. (1996) and Judd (1998).

[10]Since the policy functions generally feature less curvature than the derivatives of the value function, it is more efficient to interpolate on the policy functions and then to use the envelop conditions to update the derivatives of the value functions.

[11]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.

[12]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.

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.

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

$$s_t \equiv a_t + wh_t - c_t - i_t = \frac{a_{t+1}}{R} \tag{6a}$$

$$z_t \equiv h_t + f(i_t) = \frac{h_{t+1}}{1-\delta} \tag{6b}$$

the respective return adjusted stocks of physical and human capital. 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 $h_{t+1} = (1-\delta) z_t$.[13] 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, $\mathcal{G}^s \equiv \left\{s^{n+1}, s^{n+2}, ..., s^K\right\}$ (where $s^{n+1} = 0$, for $n > 0$, is the interior solution with zero savings; the borrowing constraint will be treated separately by adding $n > 0$ additional gridpoints, see step 3e below) and one for gross investment in human capital z, $\mathcal{G}^z \equiv \left\{z^1, z^2, ..., z^J\right\}$ as defined in (6) and form $\mathcal{G}^{s,z} = \mathcal{G}^s \otimes \mathcal{G}^z$.

2. For period T define $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$. Compared to $\mathcal{G}^s$, the grid $\mathcal{G}^a$ includes n additional gridpoints. These gridpoints represent the region with potentially binding borrowing constraints (see the previous step and step 3e below). In period T, as in step 2 of EXOGM, compute $c_T(\cdot,\cdot)$ and $i_T(\cdot,\cdot)$.

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

    (a) Construct a Delaunay triangulation of (a_{t+1}, h_{t+1}) to prepare interpolation on c_{t+1}, i_{t+1}.

[13] In a 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}) may depend on the realizations of shocks in period $t+1$, e.g., if there are shocks to the interest rate. For sake of generality, we therefore define the grid on (s_t, z_t).

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

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

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

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

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

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

$$h_t^{k,j} = z^j - \frac{1}{\xi}\left(i_t^{k,j}\right)^{\xi}$$
$$a_t^{k,j} = s^k - w h_t^{k,j} + c_t^{k,j} + i_t^{k,j}.$$

(e) At (s^{n+1}, z^j) if the endogenously computed $a_t^{n+1,j} > 0$, define an auxiliary grid $\mathcal{G}^{aux} \equiv \{a^1, a^2, ..., a^n\}$ where $a^1 = 0$ and $a^n < a_t^{n+1,j}$ to represent the region with the binding borrowing constraint. Compute $i_t^{k,j}$ by solving

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

using a non-linear solver. Then compute $c_t^{k,j} = a_t^k + w\left(\frac{h_{t+1}^j}{1-\delta} - \frac{1}{\xi}(i_t^{k,j})^{\xi}\right) - i_t^{k,j}$. If $a_t^{n+1,j} \leq 0$, then the borrowing constraint is not binding at (s^{n+1}, z^j) and we break the loop.[14]

As in EXOGM, steps 3c and 3e require interpolation on c_{t+1}, i_{t+1} and updating of V_{t+1_a}, V_{t+1_h} using the envelope conditions (5).

[14] Observe that the procedure of dealing with the borrowing constraint can be further improved by working with state and iteration dependent saving grids (cf., e.g., Krueger and Ludwig (2016)), an approach we do not adopt here.

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.[15]

The key disadvantage is the construction of the Delaunay triangulation in step 3a to prepare for Delaunay interpolation on c_{t+1}, i_{t+1}. We provide a detailed description of these methods in the next paragraph.

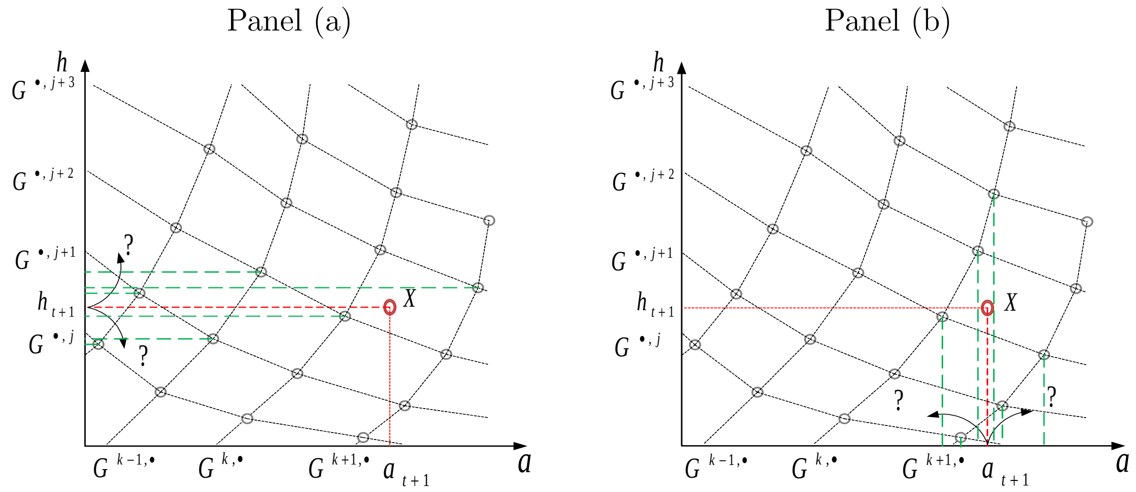
Remark 2 *In contrast to EXOGM, ENDGM is not a general method. One specification for which ENDGM does not work is a general Ben-Porath human capital function, cf. Ben-Porath (1967), in which the level of human capital directly affects the productivity of human capital investments, i.e., we replace $f(i)$ in equation (1) with $f(h, i)$. Another specification would be to let the human capital stock enter the per period utility function directly, i.e., $u(c_t, h_t)$. This exemplifies that an application of ENDGM often requires specific modeling assumptions. For a thorough formal treatment of the conditions under which ENDGM is applicable see Iskhakov (2015) and White (2015).*

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 1, 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, required in Step 3a of the algorithm, and its related geometric construct, the Voronoi diagram. We explain the geometric construction of the Voronoi diagram by use of Figure 2. The Voronoi diagram (polygon)—shown in Panel (a) of Figure 2—is the region of the state space consisting of all points closer to gridpoint P_1 than to any other gridpoint. The Voronoi diagram 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

[15]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 1: Irregular Grid

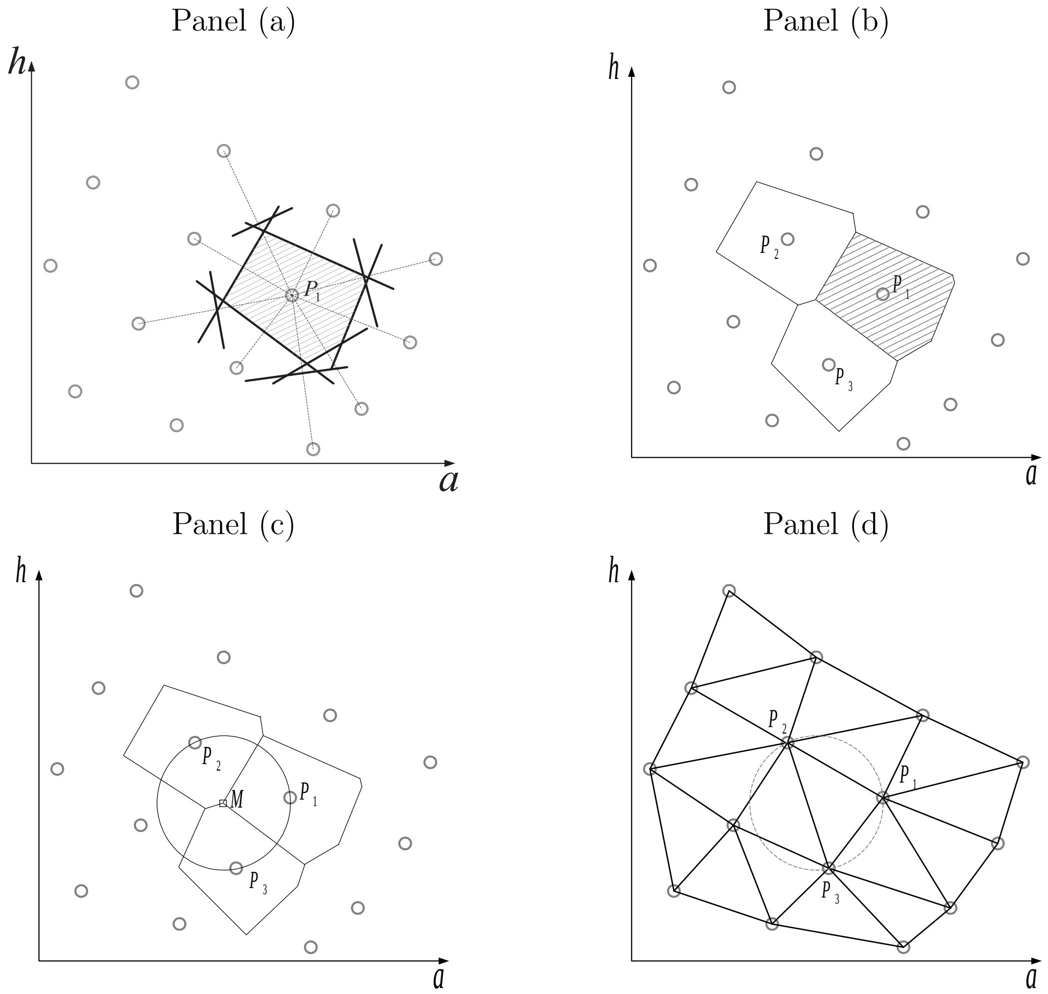


Notes: Interpolation on irregular grids. Multidimensional interpolation cannot be separated into several one-dimensional interpolations as the values of a and h change in each column or row.

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 3. 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 3. 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 newly created triangles are "legal", i.e., whether the circumcircle of any triangle does not

Figure 2: The Voronoi Diagram



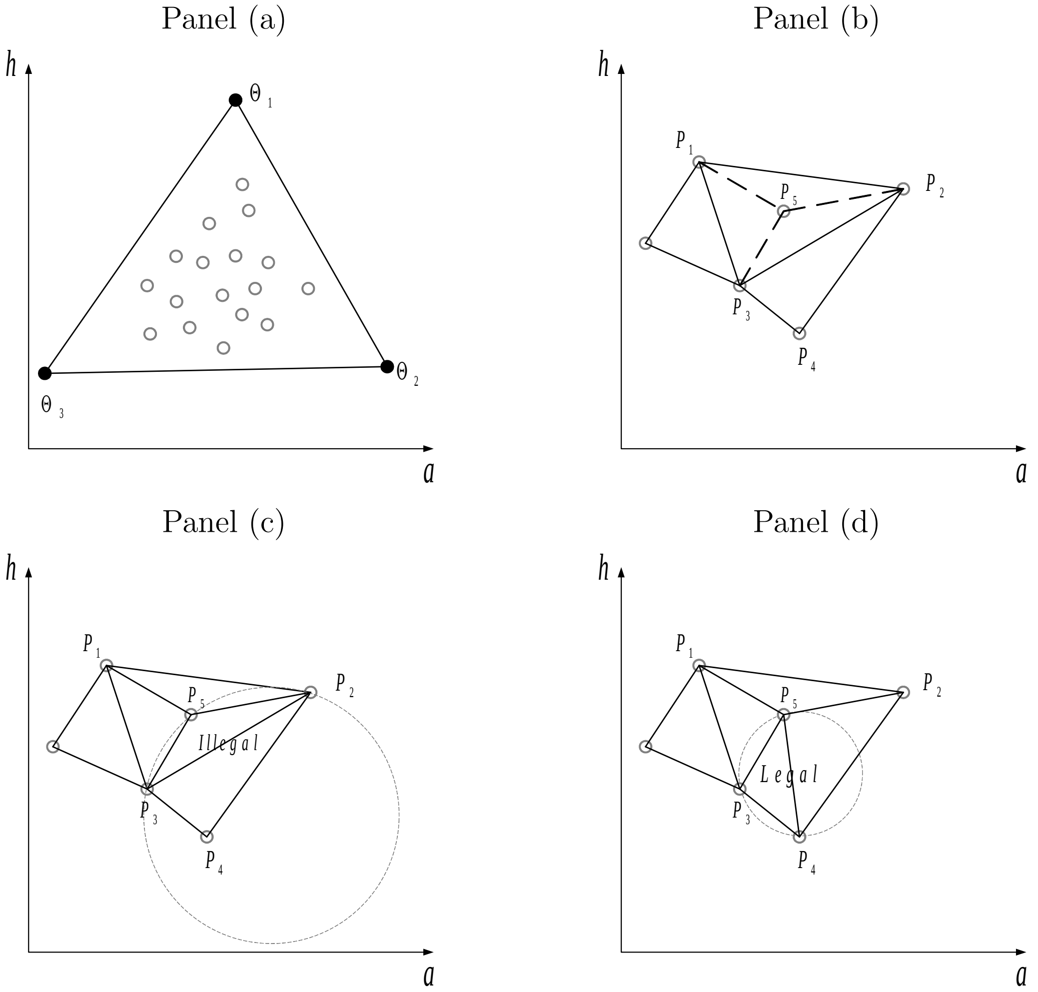
Notes: Panel (a): Generating the Voronoi polygon: Edges are perpendicular bisectors of lines connecting neighboring points. Panel (b): Several Voronoi tiles in a mesh grid. Panel (c): Circle with center at vertex includes three closest points. Panel (d): Delaunay Triangulation: Vertices are nearest neighbors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

contain any other point of the point set.[16] 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).[17]

[16]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).

[17]This procedure is described in Press et al. (2007). We use the numerical package geompack3 based

Figure 3: 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.

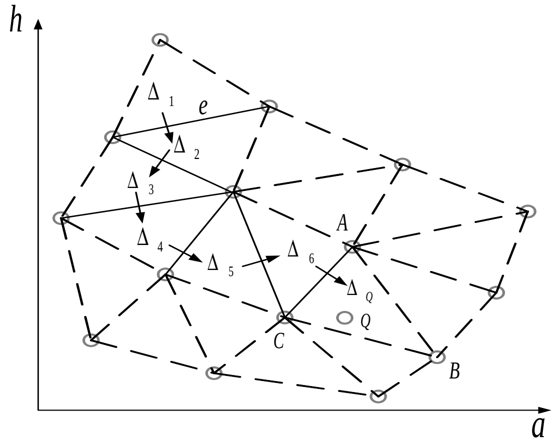
At interpolation, required in steps 3c and 3e, to locate a (query) point X in a given planar triangular mesh we adopt a procedure referred to as visibility walk, illustrated in Figure 4. The search starts from an initial guess of a triangle, Δ_1. Then, it is tested if the line supporting the first edge 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 4, this would be the case at triangle Δ_X which contains X.[18] Devillers et al. (2001) find that performance of the visibility

on Joe (1991) for both the Delaunay triangulation and the "visibility walk", described next.

[18]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).

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.

Figure 4: 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.

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 average of the respective function values at the vertices[19],

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

[19] 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.

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.[20] 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 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, $\mathcal{G}^s \equiv \left\{s^{n+1}, s^{n+2}, ..., s^K\right\}$ as in ENDGM 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. For period T, define $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$. As in ENDGM $\mathcal{G}^a$ includes n additional gridpoints compared to $\mathcal{G}^s$ representing the region of potentially binding borrowing constraints. In period T, as in step 2, of EXOGM compute $c_T(\cdot,\cdot)$, and $i_T(\cdot,\cdot)$.

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 $a^k_{t+1} = Rs^k$.

   (b) Solve the one-dimensional equation for $i^{k,j}_t$

$$i^{k,j}_t = \left(\frac{R}{(1-\delta)} \frac{V_{t+1_a}\left(\overbrace{Rs^k}^{a^k_{t+1}}, \overbrace{(1-\delta)\left(h^j_t + \frac{1}{\xi}\left(i^{k,j}_t\right)^{\xi}\right)}^{h^{k,j}_{t+1}}\right)}{\frac{\phi}{\left(1+h^{k,j}_{t+1}-\phi\right)\left(1+h^{k,j}_{t+1}\right)} V_{t+1}\left(a^k_{t+1}, h^{k,j}_{t+1}\right) + V_{t+1_h}\left(a^k_{t+1}, h^{k,j}_{t+1}\right)} \right)^{-\frac{1}{1-\xi}}$$

[20] Using Brent's method instead—which would be the standard choice in univariate problems—turns out to slow down speed of HYBGM.

using Broyden's method. This includes multiple computations of $h_{t+1}^{k,j} = (1-\delta)\left(h_t^j + \frac{1}{\xi}\left(i_t^{k,j}\right)^{\xi}\right)$.

(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) As in ENDGM, compute $a_t^{k,j}$ from the budget constraint.

(e) At (s^{n+1}, h^j) if the endogenously computed $a_t^{n+1,j} > 0$ define an auxiliary grid $\mathcal{G}^{aux} \equiv \{a^1, a^2, ..., a^n\}$ as in ENDGM. Compute $i_t^{k,j}$ by solving

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

for $i_t^{k,j}$. Next, compute $c_t^{k,j}$ as in ENDGM.

If $a_t^{n+1,j} \leq 0$, then the borrowing constraint is not binding at (s^{n+1}, h^j) and we break the loop.

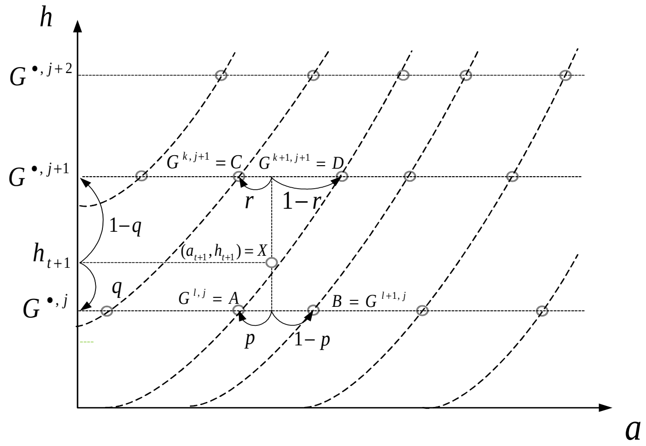
As in EXOGM, steps 3b and 3e require interpolation on c_{t+1}, i_{t+1} and updating of V_{t+1_a}, V_{t+1_h} using the envelope conditions (5).

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 in steps 3b and 3e of the method.

Remark 3 *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 which, for sake of simplicity, we let be $g(c_t)$, e.g., it could be a constant fraction of total consumption. 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 5, 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 $p = \frac{a_B - a_X}{a_B - a_A}$, $r = \frac{a_D - a_X}{a_D - a_C}$ and $q = \frac{h_C - h_X}{h_C - h_A}$. Thus, HYBGM reduces complexity of the problem without involving advanced interpolation procedures.

Figure 5: 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); $(k,j+1)$; $(l,j+1)$; $(l+1,j+1)$.

4 Results

We present results separately for the finite and infinite horizon versions of our model.[21] 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 cover the corresponding range of the state space.[22]

[21] We implement the solution in Fortran using the Intel Visual Fortran Compiler 11.1. The computation is done on a desktop computer with a consumer grade processor (Intel Core Duo E8500).

[22] Also observe, by construction, that 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.

4.1 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:

$$e_{1,t} = 1 - \frac{\left(R\psi(h_{t+1})\beta\left(c_{t+1}\right)^{-\theta}\right)^{-\frac{1}{\theta}}}{c_t}, \tag{7a}$$

$$e_{2,t} = 1 - \frac{\left(\frac{R}{(1-\delta)}\left(\frac{\psi_h(h_{t+1})V_{t+1}}{\psi(h_{t+1})(c_{t+1})^{-\theta}} + w + i_{t+1}^{1-\xi}\right)^{-1}\right)^{-\frac{1}{\xi}}}{i_t}. \tag{7b}$$

These errors are dimension free quantities. Equation (7a) 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.

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-cycle 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 (7).[23] 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.[24] 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 general equilibrium applications or in structural estimation. With regard to this

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

[24]Throughout, we have five points in the region of the binding borrowing constraint. The relative time spent on the numerical solution in this region is slightly below 10% for $N = 25^2$ and decreases to about 2% for $N = 200^2$.

relative comparison, observe from Panel (a) of Figure 6 that EXOGM is outperformed by both ENDGM and HYBGM. As the size of the grid increases, the absolute speed advantage of ENDGM over EXOGM stays roughly constant so that the relative speed advantage decreases in the size of the grid.

Table 1: Finite Horizon Model: Performance Results

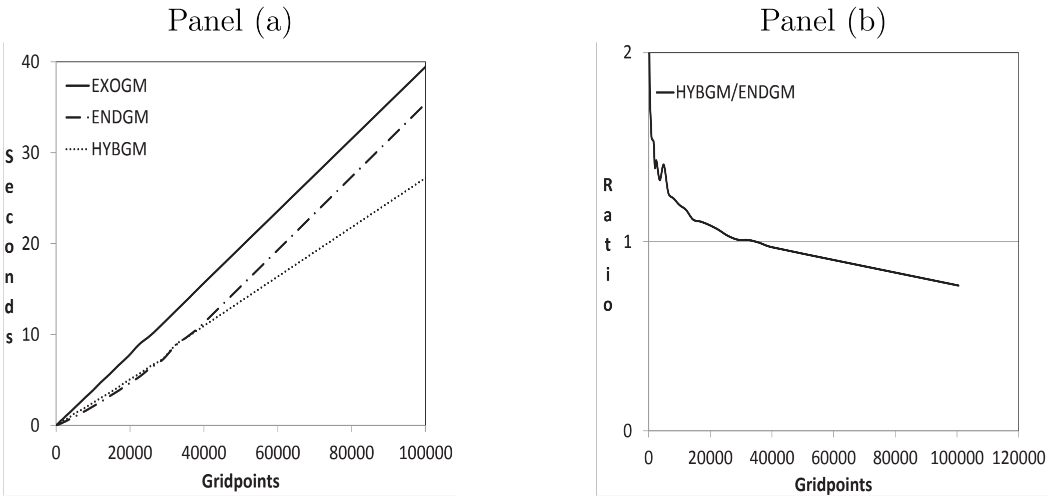
	Speed		Euler Equation Error	
Number of Gridpoints for (a, h)	Seconds	Relative to ENDGM	Maximum for c ; i	Average for c ; i
ENDGM				
$(25, 25)$	0.094	-	-2.56; -2.17	-3.70; -2.94
$(50, 50)$	0.437	-	-2.92; -2.60	-4.36; -3.53
$(100, 100)$	2.090	-	-3.37; -3.07	-4.91; -4.05
$(200, 200)$	11.278	-	-3.84; -3.47	-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.

Panel (b) of Figure 6 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 $N = 25^2$, ENDGM is about 1.7 times faster than HYBGM. For solving the model with a higher number of gridpoints, however, HYBGM is advantageous. In our setting the break-even point between both algorithms is at a number of $N = 180^2$ 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 about 1.4 to 1.7 times faster than HYBGM and about 2.3 to 2.5 times faster than EXOGM.

The reason for these findings is that the construction of the triangulation in ENDGM, cf. step 3a in the description of the algorithm, is $O(N \, log \, N)$ so that the algorithm eventually becomes worth than HYBGM—and does no longer improve over EXOGM in terms of absolute speed advantage—as the size of the grid (N) increases. Table 2 shows how the share of time spend on the triangulation increases in the number of gridpoints from

Figure 6: 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).

roughly 56% for $N = 25^2$ gridpoints to almost 80% for $N = 200^2$ gridpoints. An additional force at work in favor of EXOGM and HYBGM is that numerical solvers become faster as N increases because of the increasing density of the grid since we initialize the non-linear solvers in EXOGM and HYBGM by using solutions to the respective previous gridpoint as starting values.

Table 2: Time Requirements in ENDGM: Share of Triangulation

Number of Gridpoints for (a, h)	Finite Horizon	Infinite Horizon Pure	Infinite Horizon Approximate
$(25, 25)$	56.25%	37.83%	13.04%
$(50, 50)$	62.90%	63.57%	30.43%
$(100, 100)$	74.34%	75.55%	42.70%
$(200, 200)$	79.63%	81.22%	53.00%

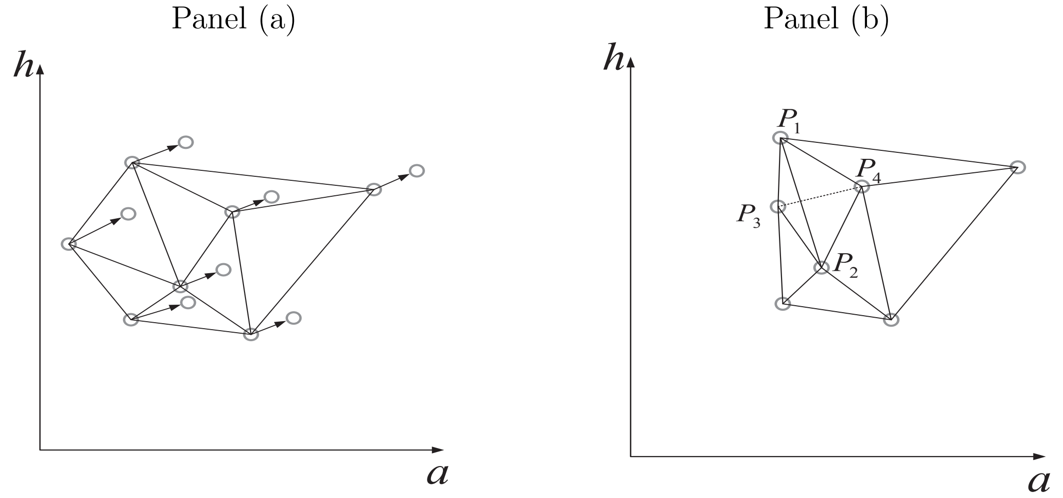
Notes: This table shows the fraction of time that is required for the Delaunay triangulation relative to total time of the ENDGM.

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 7 illustrates this. Panel (a) of the figure shows how endogenous grid-points move in the (a, h) space from 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 7 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 7: 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 (7) for the first 50 periods. Average and maximum errors are provided in Table 3.

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 again concentrate on a comparison of speed only.[25]

[25]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.

We find that ENDGM is the fastest method for all scenarios considered whereby the relative speed advantage decreases in the number of gridpoints as also documented in the finite horizon version of the model.[26] The reason for the dominance of ENDGM is the use of the variant "Approximate Delaunay" in the infinite horizon model, as described above. Table 2 shows that, relative to "Pure Delaunay" triangulation, this approximate method substantially reduces the time spend on the triangulation. 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 about 2.5 times faster than HYBGM and about 4 times faster than EXOGM, cf. Table 3.

Table 3: Infinite Horizon Model: Performance Results

	Speed		Euler Equation Error	
Number of Gridpoints for (a, h)	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

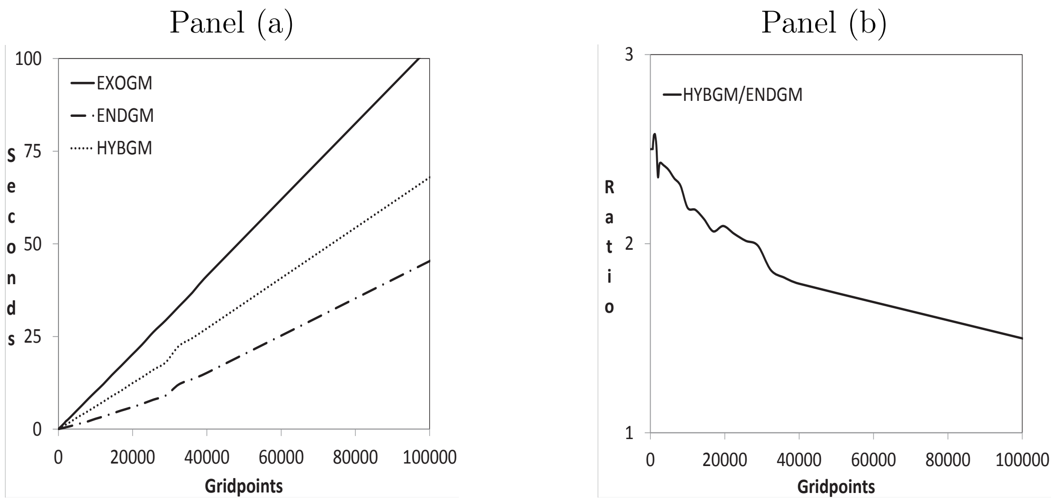
Notes: Computing time to convergence of policy functions (criterion $\varepsilon = 10^{-6}$) and resulting maximum and average Euler equation errors. Computing time is reported in seconds and absolute errors in units of base-10 logarithms.

5 Conclusion

We compare three numerical methods—the standard exogenous grid method (EXOGM), Carroll (2006)'s method of endogenous gridpoints (ENDGM), and a hybrid method

[26] Again, we have five points in the region of the binding borrowing constraint and the fraction of the overall computational time spent there varies between $1.2 - 5.3\%$.

Figure 8: Infinite Horizon Model: Speed



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).

(HYBGM)—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 (Delaunay 1934) 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, construction of triangles for 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.

For sake of simplicity, we implement this comparison in a deterministic model. Our findings therefore provide appropriate guidance to authors who use stochastic models with few shocks and few realizations per shock as in numerous general equilibrium models, cf., e.g., Krueger and Ludwig (2016) and references therein. For partial equilibrium applications with many shock realizations our results on the relative speed advantages of ENDGM are likely to indicate a lower bound. The reason is that ENDGM solves a more complex interpolation problem including the construction of the triangulation whereas EXOGM and HYBGM require more function evaluations. Complexity of this operation increases if there is risk, in particular for serial implementations. Therefore, in a relative comparison across methods, the construction step in ENDGM receives lower weight.

Three 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. Applicability requires restrictions on the model's specification and on functional forms. We provide simple examples for cases under which ENDGM and HYBGM are not applicable and refer the interested reader to White (2015) and Iskhakov (2015) for a rigorous treatment of general conditions. Second, although our code is available to users, HYBGM still has lower implementation costs than ENDGM. Third, 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. Complexity of the interpolation and storage requirements in ENDGM will also increase. As we restrict attention to two dimensional problems we cannot address how this trade-off is ultimately resolved and leave an evaluation for future research.

While our paper addresses an important computational problem and provides appropriate solutions, we naturally leave several other questions for future research. We solve the complex high dimensional interpolation by using the Delaunay method (Delaunay 1934) as a well established general interpolation method coming from Geometry. White (2015) suggests an alternative which is specific to situations in which the endogenously constructed grids feature substantial regularity. This alternative is faster for a given number of gridpoints but might be less accurate especially when the curvature of the higher dimensional grid increases. It would be interesting to investigate the exact circumstances under which one method dominates the other.[27] Last, our implementation of the Delaunay (1934) triangulation is implemented as a sequential procedure which we treat as an isolated subroutine thereby not exploiting information from previous iterations. How the method performs under a parallel triangulation as suggested by Blelloch, Hardwick, Miller, and Talmor (1999) and whether implementing an adaptive scheme is feasible are further viable research questions.

[27]In the same vein, the envelope condition method suggested by Maliar and Maliar (2013) for infinite horizon models might be adaptable to higher dimensions and could be tested against the methods used in this paper.

A Derivation of FOC

The dynamic version of the household problem reads as

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

subject to

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

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

$$\frac{\partial V_t\left(a_t, h_t\right)}{\partial c_t} = u_c - \beta\psi\left(h_{t+1}\right) V_{t+1_a} R - R\mu \overset{!}{=} 0 \Leftrightarrow u_c - \beta\psi\left(h_{t+1}\right) RV_{t+1_a} = R\mu, \tag{8}$$

$$\begin{aligned} \frac{\partial V_t\left(a_t, h_t\right)}{\partial i_t} &= \psi_h\left(h_{t+1}\right)(1-\delta) f_i \beta V_{t+1} + \psi\left(h_{t+1}\right)\beta\left(V_{t+1_a}(-R) + V_{t+1_h}(1-\delta) f_i\right) - R\mu \overset{!}{=} 0 \\ &\Leftrightarrow \psi_h\left(h_{t+1}\right)(1-\delta) f_i \beta V_{t+1} + \psi\left(h_{t+1}\right)\beta\left(V_{t+1_a}(-R) + V_{t+1_h}(1-\delta) f_i\right) = R\mu \end{aligned} \tag{9}$$

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

$$\begin{aligned} u_c - \beta\psi\left(h_{t+1}\right) RV_{t+1_a} &= 0 \\ \psi_h\left(h_{t+1}\right)(1-\delta) f_i \beta V_{t+1} + \psi\left(h_{t+1}\right)\beta\left(V_{t+1_a}(-R) + V_{t+1_h}(1-\delta) f_i\right) &= 0. \end{aligned}$$

Rearranging gives

$$\begin{aligned} u_c &= \beta\psi\left(h_{t+1}\right) V_{t+1_a} R \\ f_i &= \frac{R}{(1-\delta)} \frac{\psi\left(h_{t+1}\right) V_{t+1_a}}{\psi_h\left(h_{t+1}\right) V_{t+1} + \psi\left(h_{t+1}\right) V_{t+1_h}}. \end{aligned}$$

Case 2: Corner Solution—Binding Borrowing Constraint

In the second case the borrowing constraint is binding so that $a' = 0$ and $\mu > 0$. From (8) and (9) it then follows that

$$u_c = \psi_h (h_{t+1}) \beta V_{t+1} + \psi (h_{t+1}) \beta V_{t+1_h} (1 - \delta) f_i \tag{10}$$

and

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

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

Making use of our assumptions on functional forms, equation (10) reduces in EXOGM and HYBGM to

$$(a_t + wh_t - i_t)^{-\theta} - \frac{1}{\left((1-\delta)\left(h_t + i_t^{\xi}\right)\right)^2} V_{t+1} \left[0, (1-\delta)\left(h_t + i_t^{\xi}\right)\right) \beta (1-\delta) i_t^{-(1-\xi)} \\ - \left(1 - \frac{1}{(1-\delta)\left(h_t + \frac{1}{\xi} i_t^{\xi}\right)}\right) V_{t+1_h} \left(0, (1-\delta)\left(h_t + \frac{1}{\xi} i_t^{\xi}\right)\right) \beta (1-\delta) i_t^{-(1-\xi)} = 0$$

and in ENDGM to

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

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

$$\begin{aligned}
\frac{\partial V_t\left(a_t, h_t\right)}{\partial a_t} &\equiv V_{t_a} = \beta V_{t+1_a} R + R\mu = u_c \\
\frac{\partial V_t\left(a_t, h_t\right)}{\partial h_t} &\equiv V_{t_h} \\
&= \beta\psi_h\left(h_{t+1}\right) V_{t+1}(a_{t+1}, h_{t+1})\left(1-\delta\right) + \beta\psi\left(h_{t+1}\right) V_{t_a}(a_{t+1}, h_{t+1}) w R + \\
&\quad \beta\psi\left(h_{t+1}\right) V_{t+1_h}(a_{t+1}, h_{t+1})\left(1-\delta\right) + R\mu \\
&= \left(w + \frac{1}{f_i}\right) u_c.
\end{aligned}$$

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.

Blelloch, G. E., J. C. Hardwick, G. L. Miller, and D. Talmor (1999). Design and Implementation of a Practical Parallel Delaunay Algorithm. *Algorithmica 24*, 243–269.

Broer, T., M. Kapicka, and P. Klein (2013). Consumption Risk Sharing with Private Information when Earnings are Persistent. Technical report, Simon Fraser University.

Broyden, C. G. (1965). A Class of Methods for Solving Nonlinear Simultaneous Equations. *Mathematics of Computation 19*(92), 577–593.

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.

Dalgaard, C.-J. and H. Strulik (2014). Optimal Aging and Death: Understanding the Preston Curve. *Journal of the European Economic Association 12*(3), 672–701.

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

Iskhakov, F., T. H. Jørgensen, J. Rust, and B. Schjerning (2015). Estimating Discrete-Continuous Choice Models: The Endogenous Grid Method with Taste Shocks.

Joe, B. (1991). GEOMPACK: A Software Package for the Generation of Meshes Using Geometric Algorithms. *Advances in Engineering Software and Workstations 13*, 325–331.

Jørgensen, T. H. (2013). Structural estimation of continuous choice models: Evaluating EGM and MPEC. *Economics Letters 119*(3), 287–290.

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.

Krueger, D. and A. Ludwig (2016). On the Optimal Provision of Social Insurance: Progressive Taxation versus Education Subsidies in General Equilibrium. *forthcoming: Journal of Monetary Economics*.

Ludwig, A. and M. Schön (2013). Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods.

Maliar, L. and S. Maliar (2013). Envelope condition method versus endogenous grid method for solving dynamic programming problems. *Economics Letters 120*(2), 262–266.

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.

White, M. N. (2015). The method of endogenous gridpoints in theory and practice. *Journal of Economic Dynamics and Control 60*, 26–41.