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

Alexander Ludwig[†]   Matthias Schön[‡]

September 5, 2013

Abstract

This paper investigates extensions of the method of endogenous grid-points (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 (EXGM), (ii) the pure method of endogenous grid-points (ENDGM) and (iii) a hybrid method (HEGM). ENDGM comes along with Delaunay interpolation on irregular grids. Comparison of methods is done by evaluating speed and accuracy. We find that HEGM and ENDGM both dominate EXGM. The choice between HEGM and ENDGM depends on the number of dimensions and the number of grid-points in each dimension. With less than 150 grid-points in each dimension ENDGM is faster than HEGM, and vice versa. For a standard choice of 20 to 40 grid-points in each dimension, ENDGM is 1.6 to 1.8 times faster than HEGM.

JEL Classification: C63, E21.

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

*We thank Johannes Brumm, Christopher Carroll, Michael Reiter and seminar participants at University of Cologne, the 2012 CEF and the Cologne Macroeconomic Workshop 2012 for helpful comments.

[†] CMR & FiFo, University of Cologne; MEA; Netspar; Albertus-Magnus-Platz; 50923 Köln; Germany; E-mail: ludwig@wiso.uni-koeln.de

[‡] CMR, University of Cologne; 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. 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 in this literature is Carroll (2006) who introduces the method of endogenous grid points (ENDGM). In comparison to the method of exogenous grid-points (EXGM), ENDGM greatly enhances computational speed because part of the problem can be computed in closed form.

This paper investigates extensions of Carroll's method to dynamic problems with more than one continuous endogenous state variable. We highlight an important drawback of ENDGM in higher dimensions which is due to the endogenously computed states. The resulting state grid is not rectangular, i.e., grid-points are irregularly distributed in the space. In consequence, even linear interpolation is much more costly than for conventional rectangular grids. Hence, there exists a fundamental trade-off between EXGM and ENDGM in higher dimensions. On the one hand, EXGM requires the use of numerical routines throughout whereas ENDGM computes solutions to first-order conditions in closed form. On the other hand, interpolation in EXGM 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 interpolation originally coming from the field of geometry is not commonly adopted in Economics. It was only recently introduced to the field by Brumm and Grill (2010).

In addition to EXGM and ENDGM, we consider a third algorithm, a hybrid method of exogenous and endogenous grid-points (HEGM). HEGM uses endogenous grids in some but not all dimensions. The trade-off between HEGM and ENDGM is therefore between costly routines 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. 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 HEGM and ENDGM both dominate EXGM. They are both substantially faster. The choice between HEGM and ENDGM depends on the number of grid-points in each dimension. For a relatively low number of grid-points, ENDGM is advantageous and vice versa for HEGM. We discuss limitations of ENDGM and HEGM which are both only applicable for specific problems at hand. Furthermore, we argue that the relative advantage of HEGM decreases in higher dimensions (higher than two).

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) incorporate two endogenous state variables in a durable goods model and apply a version of HEGM. Their algorithm exploits the fact that durable goods consumption is zero at the end of life. In a human capital model such as ours this is not possible because the human capital stock is neither zero nor known at the end of life. Hence our version of HEGM differs from the Hintermaier and Koeniger (2010) approach.

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 grid-points, the pure method of endogenous grid-points and the hybrid method of exogenous and endogenous grid-points. Section 4 presents results according to speed and accuracy of all three methods. Section 5 concludes. Additional material is contained in an appendix.

2 General Framework

We develop a human capital model which allows us to illustrate and compare three approaches to solve dynamic models with two endogenous states using first-order methods. In addition to financial assets there is another endogenous state, 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 simple. Of course, the underlying trade-off between solution methods will also hold in more complex problems.

2.1 A Simple Human Capital Model

We consider a setup in which the only risk is the risk to survive to the next period. 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 $u\left(c_t\right)$ is assumed to be strictly concave, and the probability to survive to the next period $s\left(h_t\right)$ is assumed to be non-linear in the accumulated human (or health) capital stock h_t. Income of the agent, y_t, consists of labor income which depends on the amount of accumulated human capital h_t,

$$y_t = wh_t,$$

where w is the wage rate.

In each period the household faces the decision to consume c_t, to invest in a risk-free financial asset, a_t, which earns (gross) interest R and to invest an amount i_t in 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 the Inada conditions are satisfied, i.e., $\lim_{i_t \to 0} f_i = \infty$ and $\lim_{i_t \to \infty} f_i = 0$. 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. The human capital accumulation equation is accordingly given by

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

where h_0 is given.

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

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

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

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

subject to the constraints

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

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 θ:

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

The human capital production function is

$$f\left(i_t\right) = \gamma \frac{i_t^{1-\alpha} - 1}{1-\alpha}$$

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

$$s\left(h_t\right) = 1 - \phi \frac{1}{1+h_t}.$$

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 these conditions may not hold in human capital models because endogenous human capital formation may lead to non-convexities (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 methods developed in this paper 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.

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

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

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

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

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

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

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, $c_t\left(\cdot,\cdot\right)$, investment in human capital, $i_t\left(\cdot,\cdot\right)$, next period's financial assets, $a_{t+1}\left(\cdot,\cdot\right)$, and next period's human capital, $h_{t+1}\left(\cdot,\cdot\right)$, 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:

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

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

Using (3a) together with (4a) gives the standard Euler equation of consumption.[1]

2.2 Calibration

We choose the same parameterization of the model for all solution methods described in section 3. The coefficient of relative risk aversion parameter is set to $\theta = 0.5$ to assure a positive value of life. We set the time preference rate to $\rho = 0.04$. In order to achieve incentives to save in the finite horizon setting without introducing uncertainty we set an interest rate of $R - 1 = 0.05$. In the infinite horizon setting we set an interest rate of $R - 1 = 0.035$ smaller than ρ to assure that financial assets are bounded. For the depreciation rate of human capital we take $\delta = 0.05$. The parameters of the human capital production function are $\alpha = 0.65$ and $\gamma = 1.0$, respectively. 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 ($t = T, ..., 0$). 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. The first algorithm (EXGM) applies a multi-dimensional Quasi-Newton method. Standard interpolation methods are used. The second algorithm (ENDGM) uses the method of endogenous grid-points and thereby solves the system of equations (3) analytically. It is accompanied by Delaunay interpolation. The third algorithm (HEGM) combines the former two, i.e., it applies the method of endogenous grid-points (and closed form solutions) in one dimension and uses a one-dimensional Quasi-Newton method in the other dimension. As EXGM, HEGM comes along with a standard interpolation procedure.

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

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

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 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 EXGM are as follows:

1. To initialize EXGM 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 $T + 1$[2] and income is completely consumed for all $\left(a^k, h^j\right) \in \mathcal{G}^{a,h}$:

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

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

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

3. Given functions V_{t+1}, V_{t+1_a} and V_{t+1_h} from the previous step iterate backwards on $t = T - 1, ..., 0$. For each $\left(a_t^k, h_t^j\right) \in \mathcal{G}^{a,h}$:

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

(a) Solve the two-dimensional equation system

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

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

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

$$\begin{aligned}
V_t\left(a_t^k,h_t^j\right) &= \frac{1}{1-\theta}\left[c_t^{k,j}\right]^{1-\theta}+\beta\left(1-\phi\frac{1}{1+h_{t+1}^{k,j}}\right)V_{t+1}(a_{t+1}^{k,j},h_{t+1}^{k,j}) \\
V_{t+1_a}\left(a_t^k,h_t^j\right) &= \left[c_t^{k,j}\right]^{-\theta} \\
V_{t+1_h}\left(a_t^k,h_t^j\right) &= \left[w+\left(i_T^{k,j}\right)^{\alpha}\right]\left[c_t^{k,j}\right]^{-\theta}.
\end{aligned}$$

Since EXGM requires to apply the solver for each point in $\mathcal{G}^{a,h}$, this procedure entails solving the multidimensional equation system $[K * J]$ times. 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.[3] An additional shortcoming of EXGM compared to ENDGM and HEGM is that it cannot exactly determine the region in which the borrowing constraint is binding. Policy functions are imprecise at the kink. This may also cause convergence problems. Furthermore, numerical methods often require fine tuning so that stability of the solver is ascertained. For some parameter constellations we in fact encountered instability problems in EXGM.

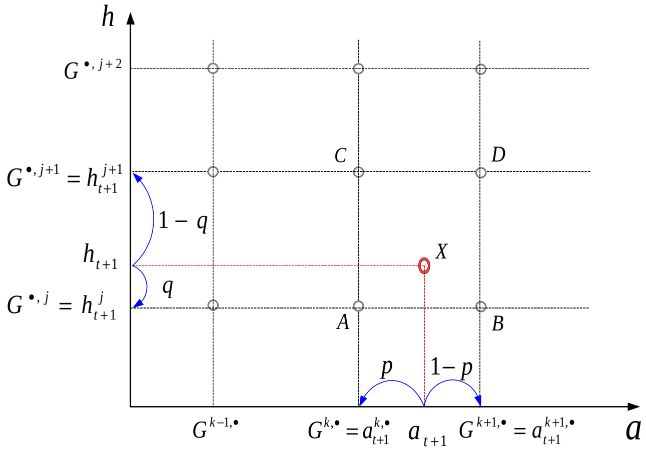
Interpolation on a Rectilinear Grid Step 3a requires evaluation of both the value function V_{t+1} and its derivatives , V_{t+1_a} and V_{t+1_h}. We apply piecewise linear interpolation. We determine interpolation nodes by the concept "grid square", cf. Press et al. (1996). In order to apply this procedure it is necessary to have a rectilinear grid, i.e., the state space has to be tessellated by rectangles.[4] In this case all grid-points in row $\mathcal{G}^{\bullet,j}$ have

[3] Similar to secant methods in one dimension Broyden's algorithm only converges superlineraly at $O(n\log n)$ in the neighborhood of the root, cf. Press et al. (1992).

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

the same value of h^j, and all grid-points 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 closest neighbors in the grid are identified in about $\log_2 N$ tries 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


h
$G^{\bullet,j+2}$
C
D
$G^{\bullet,j+1} = h_{t+1}^{j+1}$
$1-q$
h_{t+1}
X
q
$G^{\bullet,j} = h_{t+1}^{j}$
A
B
p
$1-p$
$G^{k-1,\bullet}$
$G^{k,\bullet} = a_{t+1}^{k,\bullet}$
a_{t+1}
$G^{k+1,\bullet} = a_{t+1}^{k+1,\bullet}$
a

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 EXGM, $\mathcal{G}^a \otimes \mathcal{G}^h$ is predetermined as a rectilinear grid (in every iteration). After locating the nodes bilinear 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

$$\begin{aligned}
\varphi_A &= p * q \\
\varphi_B &= (1-p) * q \\
\varphi_C &= p * (1-q) \\
\varphi_D &= (1-p) * (1-q)
\end{aligned}$$

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 control variables (a_{t+1}, h_{t+1}). The main idea of ENDGM is to redefine exogenous and endogenous objects in the numerical solution: the control variable is taken as exogenous whereas today's state is taken endogenous. In this sense, "Endogenous Grid Method" is a misnomer because computation is still based on an exogenous grid, not for the state but for the control variables.[5]

Computationally we first condition on a set of future endogenous state variables, (a_{t+1}, h_{t+1}),—which are control variables from the perspective of the current period—and exploit the system of FOCs to determine the corresponding set of contemporaneous controls, (c_t, i_t). Second, we use the budget constraint and the law of motion for human capital to get the according current set of 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, $G^s \equiv \left\{s^1, s^2, ..., s^K\right\}$ which is defined as $s = a_t + wh_t - c_t - i_t = \frac{a_{t+1}}{R}$ and one for human capital z, $G^z \equiv \left\{z^1, z^2, ..., z^J\right\}$ which is defined as $z = h_t + f\left(i_t\right) = \frac{h_{t+1}}{1-\delta}$ 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. In period T, as in EXGM,

$$\begin{aligned} c_T\left(\cdot,\cdot\right) &= a_T^{k,j} + wh_T^{k,j} \\ i_T\left(\cdot,\cdot\right) &= 0 \end{aligned}$$

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

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

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

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

$$\begin{aligned} a_{t+1}^k &= Rs^k, \\ h_{t+1}^j &= \left(1-\delta\right)z^j. \end{aligned}$$

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

[5] A more appropriate name might be "Endogenous State Grid Method".

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

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

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

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

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

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

$$\begin{aligned} V_t\left[a_t^{k,j}, h_t^{k,j}\right] &= \frac{1}{1-\theta}\left[c_t^{k,j}\right]^{1-\theta} + \beta\left(1 - \phi \frac{1}{1+h_{t+1}^j}\right) V_{t+1}\left[a_{t+1}^k, h_{t+1}^j\right] \\ V_{t_a}\left[a_t^{k,j}, h_t^{k,j}\right] &= \left[c_t^{k,j}\right]^{-\theta} \\ V_{t_h}\left[a_t^{k,j}, h_t^{k,j}\right] &= \left[w + \left(i_T^{k,j}\right)^{\alpha}\right]\left[c_t^{k,j}\right]^{-\theta}. \end{aligned}$$

The clear advantage of ENDGM compared to EXGM becomes obvious in step 3c. Because of the redefinition of a_{t+1} and h_{t+1} 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 definition, an exact determination of the range of the borrowing constraint and produces higher accuracy of the solution than EXGM in this region. However, in contrast to the standard one-dimensional problem, the policy function itself in the range where the borrowing constraint is binding does not have a closed form solution in ENDGM.[6] In a multi-dimensional setting, the consumption and human capital investment policy functions are not necessarily linear functions of current cash on hand. To accommodate this we set exogenous grid-points in the region of the borrowing constraint and use a one-dimensional root-finder in order to compute the policy function. For more details see the appendix.

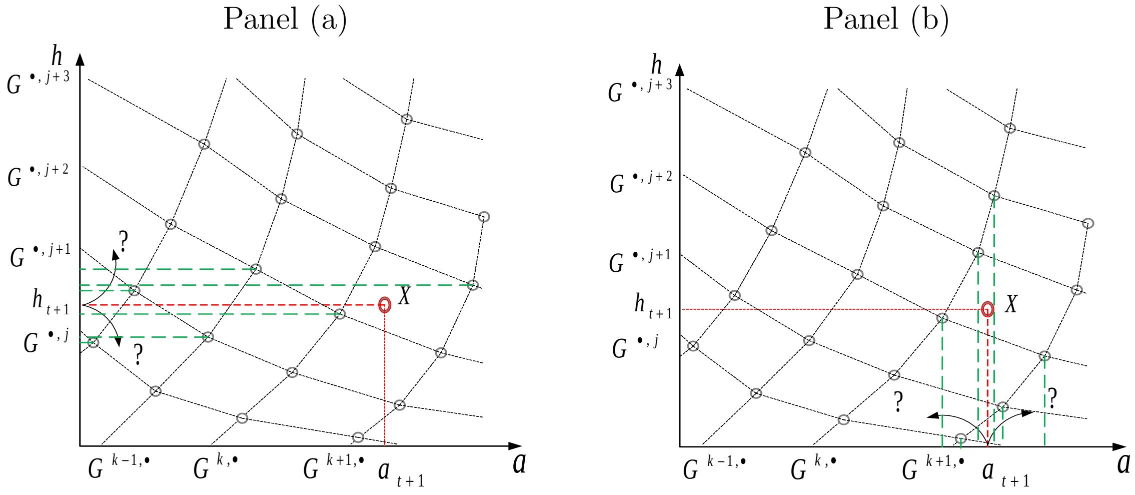
Remark 1 *In contrast to EXGM, ENDGM is not a general method. Suppose we were to adopt a general Ben-Porath human capital function, cf. Ben-Porath (1967), in which*

[6] 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).

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.

Delaunay Interpolation In EXGM 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 bilinear 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.

Figure 2: Irregular Grid



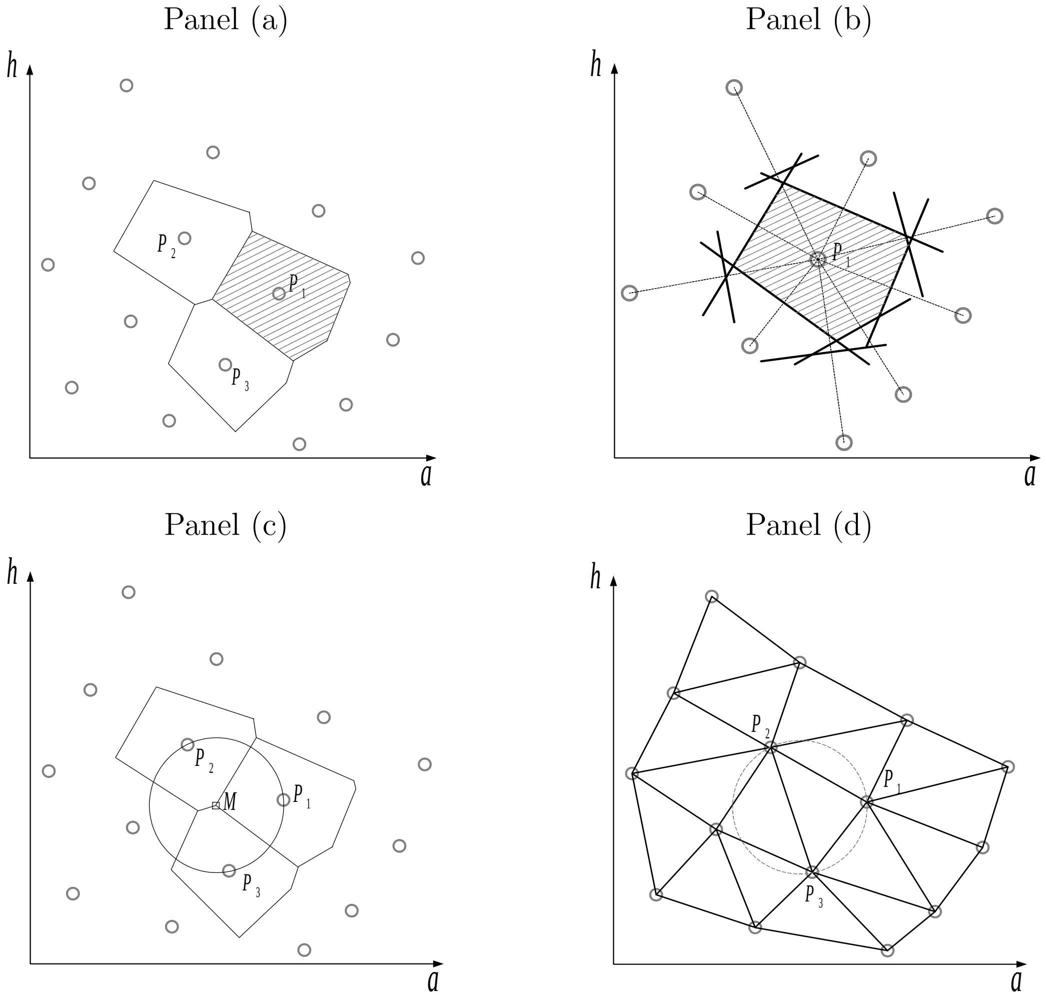
Interpolation on irregular grids. Multidimensional interpolation cannot be separated into several one-dimensional interpolations as the values of a and h change in each column or row.

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.[7] 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 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 grid-points, cf. Panel (b). The Voronoi vertices are the points equidistant to three

[7] Recently, these methods have been introduced to Economics, cf., e.g., Brumm and Grill (2010)

grid-points, i.e., they are the center of circumcircles including the three neighboring grid-points, cf. Panel (c). Connecting these grid-points 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.

Figure 3: The Voronoi Diagram



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.

The computational implementation of a Delaunay triangulation is done by the so-called randomized incremental algorithm, which we illustrate 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, 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 newly created triangles are "legal", i.e., whether the circumcircle of any triangle does not contain any other point of the point set.[8] 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).[9]

At interpolation, to locate a (query) point X in a given planar triangular mesh we adopt a procedure referred to as visibility walk which is 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.[10] 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 grid-points visited previously.

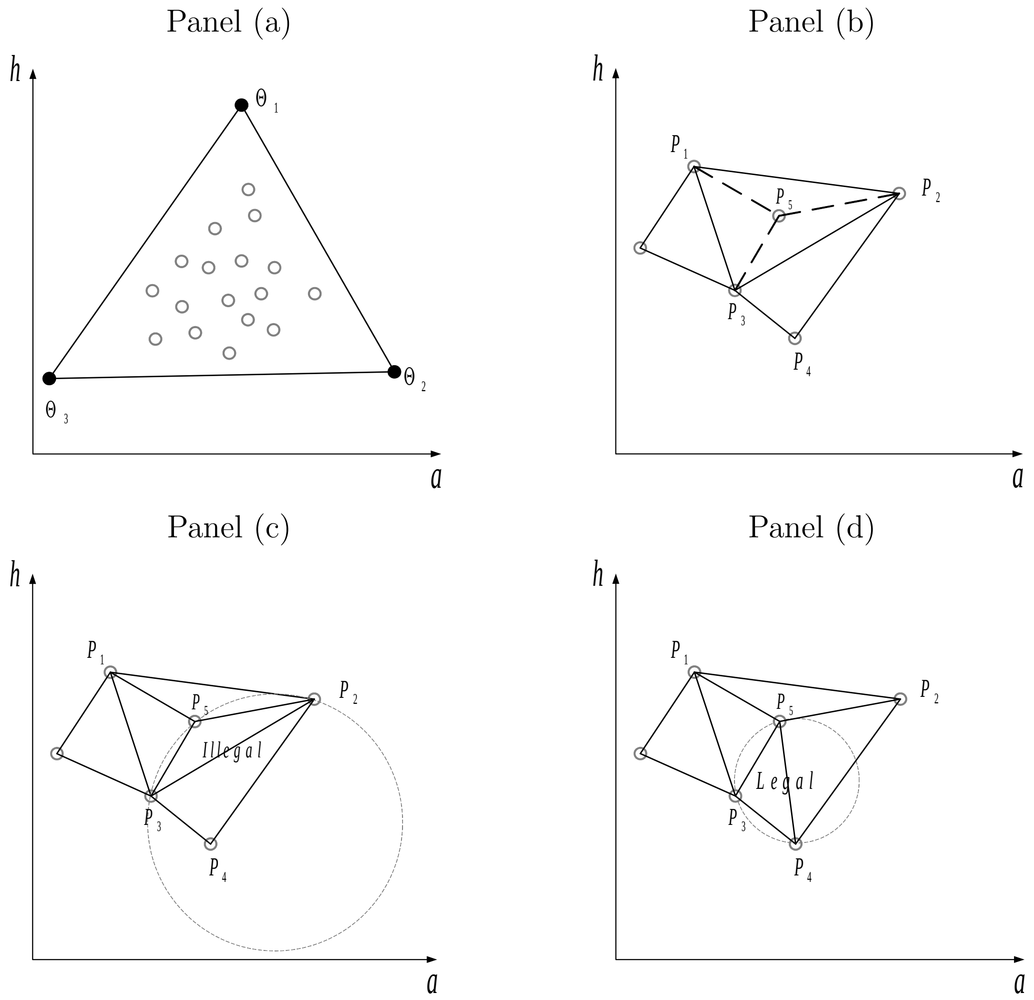
After locating the triangle we compute the normalized barycentric coordinates (weights)

[8] 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).

[9] 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.

[10] 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



Panel (a)

Panel (b)

Panel (c)

Panel (d)

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.

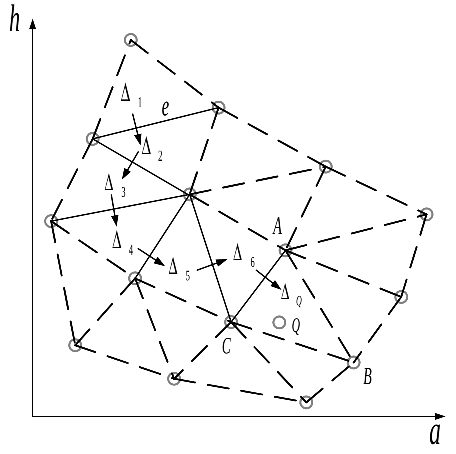
of the query point X with respect to the vertices (A, B, C) of the triangle Δ_X

$$\begin{aligned}\varphi_A &= \frac{(a_X - a_C)(h_B - h_C) + (a_C - a_B)(h_X - h_C)}{(a_A - a_C)(h_B - h_C) + (a_C - a_B)(h_A - h_C)}\\ \varphi_B &= \frac{(a_X - a_C)(h_C - h_A) + (a_A - a_C)(h_X - h_C)}{(a_A - a_C)(h_B - h_C) + (a_C - a_B)(h_A - h_C)}\\ \varphi_C &= 1 - \varphi_A - \varphi_B.\end{aligned}$$

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,

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

Figure 5: Visibility Walk



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.

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

We next consider a hybrid method (HEGM) which combines EXGM 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 a_{t+1} and current state h_t, we exploit one of the two FOCs to derive one policy function—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.[11] In the second step, policy function i_t is used to get the second endogenous state variable, h_{t+1}. Exploiting the second FOC we get the second policy function, c_t. In the third step, we compute the corresponding state variable a_t from the budget constraint. The implementation steps are as follows:

1. To initialize HEGM 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 $\mathcal{G}^{a,h} = \mathcal{G}^a \otimes \mathcal{G}^h$ and compute

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

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

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

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

3. Given functions V_{t+1}, V_{t+1_a} and V_{t+1_a} from the previous step iterate backwards on $t = T-1, ..., 0$. For each $\left(s^k, h^j\right) \in \mathcal{G}^{s,h}$:

   (a) Solve the one-dimensional equation system for $i_t^{k,j}$

$$i_t^{k,j} = \left[\frac{R}{(1-\delta)} \frac{V_{t+1_a}\left[\overbrace{Rs^k}^{a^k_{t+1}}, \overbrace{(1-\delta)\left(h_t^j + \frac{1}{1-\alpha}\left(i_t^{k,j}\right)^{1-\alpha}\right)}^{h^{k,j}_{t+1}}\right]}{\frac{\phi}{\left(1+h^{k,j}_{t+1}-\phi\right)\left(1+h^{k,j}_{t+1}\right)} V_{t+1}\left[a^k_{t+1}, h^{k,j}_{t+1}\right] + V_{t+1_h}\left[a^k_{t+1}, h^{k,j}_{t+1}\right]}\right]^{-\frac{1}{\alpha}}$$

   using Broyden's method. This includes several computations of a^k_{t+1} and $h^{k,j}_{t+1}$ and hybrid interpolations—described below—on V_{t+1}, $V_{t+1_a}$and V_{t+1_h}.

   (b) Use $i_t^{k,j}$ to compute

$$h^{k,j}_{t+1} = (1-\delta)\left(h_t^j + \frac{1}{1-\alpha}\left(i_t^{k,j}\right)^{1-\alpha}\right)$$

   and next compute $c_t^{k,j}$ as

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

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

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

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

$$\begin{aligned}
V_t\left(a_t^{k,j}, h_t^j\right) &= \frac{1}{1-\theta}\left[c_t^{k,j}\right]^{1-\theta} + \beta\left(1-\phi\frac{1}{1+h_{t+1}^{k,j}}\right)V_{t+1}(a_{t+1}^k, h_{t+1}^{k,j}) \\
V_{t_a}\left(a_t^{k,j}, h_t^j\right) &= \left[c_t^{k,j}\right]^{-\theta} \\
V_{t_h}\left(a_t^{k,j}, h_t^j\right) &= \left[w + \left(i_T^{k,j}\right)^{\alpha}\right]\left[c_t^{k,j}\right]^{-\theta}.
\end{aligned}$$

As EXGM, HEGM requires to run a numerical solver $[K * J]$ times. 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, HEGM is not a general method. Suppose that consumption has an additional effect on human capital. 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 HEGM inapplicable.*[12]

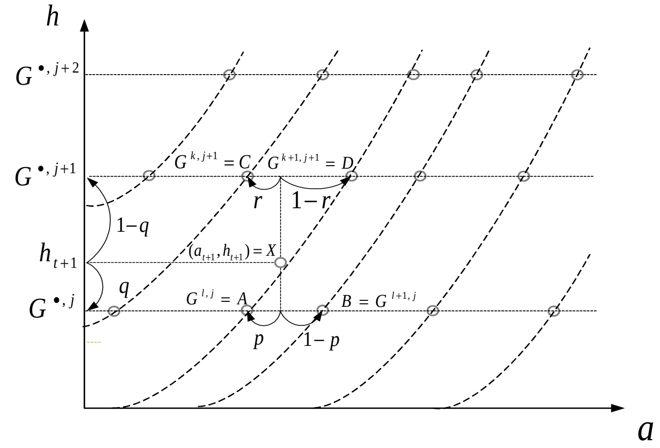
Hybrid Interpolation Hybrid interpolation, illustrated in Figure 6, is defined on a curvilinear grid where one dimension is being held constant. To locate any query point X hybrid interpolation proceeds in three steps. First, in the dimension of the exogenous grid (current state h_t) find the most narrow bracket of h_{t+1} and compute the weights according to the relative distance to these grid-points. Second, in both rows, find those grid-points 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

$$\begin{aligned}
\varphi_A &= p * q \\
\varphi_B &= (1-p) * q \\
\varphi_C &= r * (1-q) \\
\varphi_D &= (1-r) * (1-q)
\end{aligned}$$

[12] Hintermaier and Koeniger (2010) show a potential way to solve this specific problem by a different kind of HEGM. Hintermaier and Koeniger replace the numerical solver in step 3a with an additional outer loop over a guess for a future endogenous state. In finite horizon models, this procedure requires an exact knowledge of the state in the last period. In a durable goods model, Hintermaier and Koeniger set both states to zero in the last period T+1. In a human capital model such as ours such an assumption is however invalid. It can only be assumed that optimal investment in human capital is zero in the last period but not the human capital stock itself.

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, HEGM reduces complexity of the problem without involving advanced interpolation procedures.

Figure 6: Hybrid Interpolation



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 grid-points 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. Throughout, we use triple exponential grids for a, h, s, z, respectively. We set the range of grid G_s to $[0, 300]$ and the range of G_z to $[1, 300]$. The according grids G_a and G_h are adjusted to cover the corresponding range of the state space.

4.1 Finite Horizon

We iterate over $T = 100$ time periods. Computational speed of the respective algorithms is measured in seconds. 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:

$$e_1 = 1 - \frac{\left[Rs(h_{t+1})\beta \left(c_{t+1}\right)^{-\theta} \right]^{-\frac{1}{\theta}}}{c_t},$$

$$e_2 = 1 - \frac{\left[\frac{R}{(1-\delta)} \left(\frac{s_h(h_{t+1})V_{t+1}}{s(h_{t+1})(c_{t+1})^{-\theta}} + w + \frac{1}{(i_{t+1})^{-\alpha}} \right)^{-1} \right]^{-\frac{1}{\alpha}}}{i_t}.$$

This error is a dimension free quantity expressing the optimization error as a fraction of current consumption. An error of $e_1 = 10^{-3}$, for instance, means the household would make 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. To compare these methods in terms of accuracy we simulate 80 life-cycles profiles. Initial assets are set to $a_0 = 10$ whereas initial human capital h_0 differs in the range $\{50, 70\}$. For each age we compute e_1 and e_2.

Averages and maximum errors 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 speed only.

Table 1 shows computing times for EXGM, ENDGM and HEGM for different numbers of grid-points. As can further be seen in Panel (a) of Figure 7 EXGM is outperformed by both ENDGM and HEGM. Furthermore, we experience numerical stability problems with EXGM, especially for large number of grid-points.[13] Handling the applied numerical routines becomes cumbersome and finding a solution is not guaranteed. If possible, EXGM should therefore be avoided in higher dimensional problems.

Panel (b) of Figure 7 shows that ENDGM has a relative advantage in comparison to HEGM in solving the model with a relatively small number of grid-points. At a grid size of 10^2, ENDGM is almost two times faster than HEGM. For solving the model with a higher number of grid-points, however, HEGM is advantageous. At a grid size of 500^2 HEGM is two times faster than ENDGM. In our setting the break-even point between both algorithms is at a number of 150^2 grid-points and a computing time of $5.5s$. As can be seen from table 1, for a standard choice of 20 to 40 grid-points in each dimension, ENDGM is $\frac{0.4992}{0.3120} \approx 1.6$ to $\frac{0.1404}{0.078} \approx 1.8$ times faster than HEGM and $\frac{0.234}{0.078} \approx 3$ to $\frac{1.0452}{0.3120} \approx 3.4$ times faster than EXGM.

4.2 Infinite horizon

Algorithms are also comparable in the infinite horizon setting. In all approaches, 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 $\varepsilon = 10^{-6}$.[14]

To compute Euler equation errors we simulate the response to a shock to financial assets and health capital for 50 periods. We set the assets $a_0 = 350$ and health in the range of $\{50, 70\}$. We compute e_1 and e_1 for the first 50 periods. Averages and maximum errors are provided in table 2. As in the finite horizon setting, Euler equation errors are of similar magnitudes across algorithms—which we also achieve by appropriate

[13] Similar problems are documented by Hintermaier and Koeniger (2010).

[14] In our algorithms we define $\tilde{c} = \left\{ \{\tilde{c}^{k,j}\}_{k=1}^{K} \right\}_{j=1}^{J}$ and $\tilde{\imath} = \left\{ \{\tilde{\imath}^{k,j}\}_{k=1}^{K} \right\}_{j=1}^{J}$ and exit in iteration n if $\sup |\tilde{c} - c^n| \leq \varepsilon \cup |\tilde{\imath} - i^n| \leq \varepsilon$.

Table 1: Finite Horizon Model: Speed

Number of Gridpoints for (a, h)	Time (seconds)	Maximum Euler error for $(c; i)$	Average Euler error for $(c; i)$
ENDGM			
$(20, 20)$	0.078	$(-2.58;\ -1.33)$	$(-3.52;\ -3.08)$
$(40, 40)$	0.3120	$(-2.97;\ -1.32)$	$(-4.28;\ -3.22)$
$(100, 100)$	2.1372	$(-3.47;\ -1.33)$	$(-5.15;\ -3.32)$
$(200, 200)$	10.6861	$(-4.28;\ -1.33)$	$(-5.75;\ -3.35)$
HEGM			
$(20, 20)$	0.1404	$(-2.76;\ -1.33)$	$(-3.42;\ -2.76)$
$(40, 40)$	0.4992	$(-3.00;\ -1.33)$	$(-4.40;\ -3.13)$
$(100, 100)$	2.6832	$(-3.44;\ -1.33)$	$(-5.25;\ -3.31)$
$(200, 200)$	10.1245	$(-3.95;\ -1.33)$	$(-5.86;\ -3.35)$
EXGM			
$(20, 20)$	0.234	$(-2,81;\ -1,33)$	$(-3.46;\ -2.75)$
$(40, 40)$	1.0452	$(-3,00;\ -1,32)$	$(-4.40;\ -3.12)$
$(100, 100)$	7.2384	$(-3,44;\ -1,33)$	$(-5.24;\ -3.31)$
$(200, 200)$	27.7214	$(-3,94;\ -1,33)$	$(-5.86;\ -3.35)$

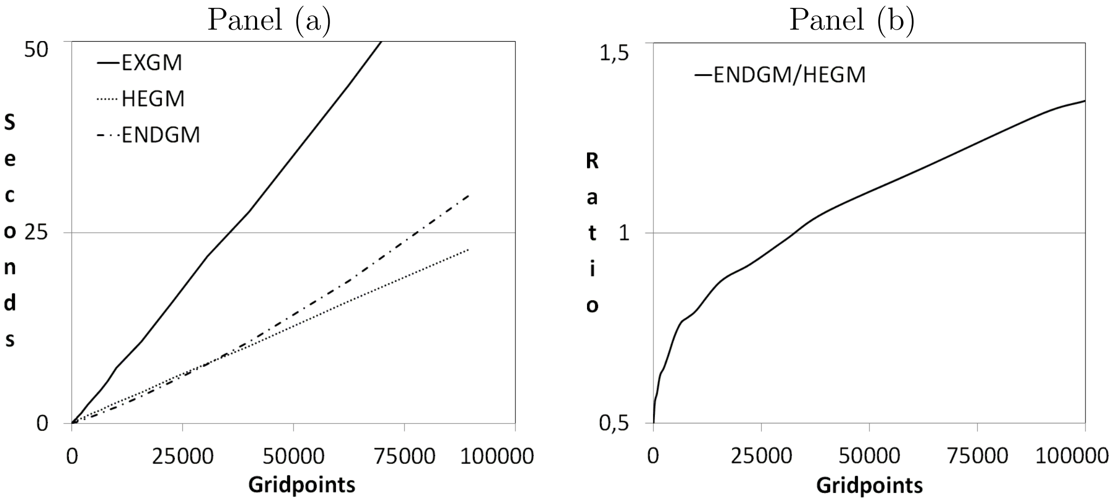
Computing time for $T = 100$ and resulting maximum and average Euler equation errors. Computing time is reported in seconds and absolute errors in units of base-10 logarithms.

settings of the respective numerical routines—so that we can again further concentrate on comparison speed only.

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—100 in our case.[15] As in the finite horizon setting, ENDGM is the fastest method for not too many grid-points and the comparative advantage decreases for a higher number of grid-points. Both methods clearly dominate EXGM in terms of speed. Similar to our findings in the finite horizon case, for a standard choice of 20 to 40 grid-points in each dimension, ENDGM is $\frac{0.41}{0.22} \approx 1.9$ to $\frac{1.39}{0.70} \approx 2.0$ times faster than HEGM and $\frac{0.69}{0.22} \approx 3.1$ to $\frac{3.03}{0.70} \approx 4.3$ times faster than EXGM, cf. table 2. As in the finite horizon setting EXGM faces severe problems in terms of stability for a high number of grid-points.

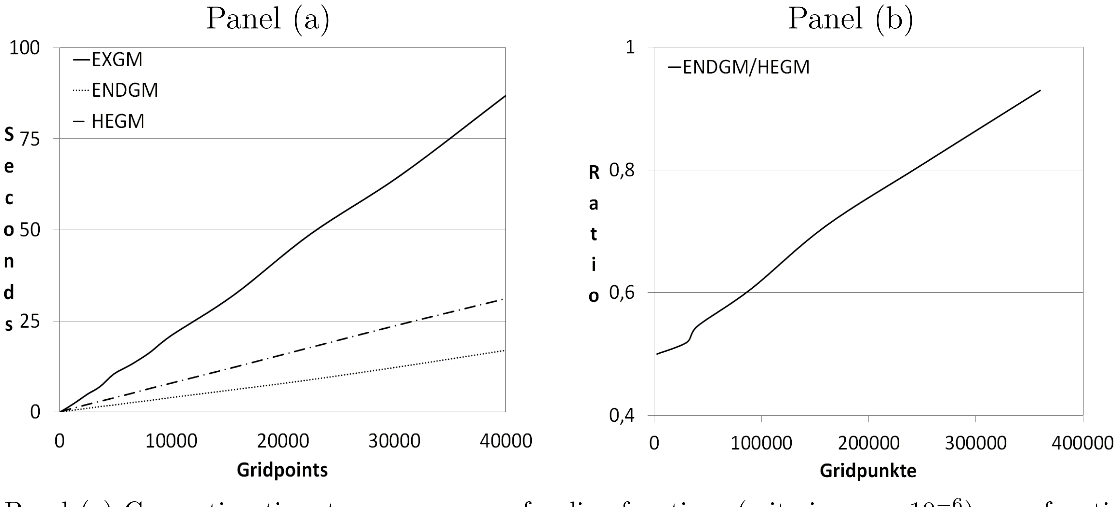
[15] It is necessary to assure that the endogenously computed grid-points form a convex hull.

Figure 7: Finite Horizon Model: Speed



Panel (a): Computing time as a function of grid-points in seconds (with equally many grid-points in both dimensions). Solid line: computing time of EXGM; dotted line: computing time of HEGM; dashed-dotted line: computing time of ENDGM. Panel (b): Ratio of computing time of ENDGM to HEGM as a function of grid-points (with equally many grid-points in both dimensions).

Figure 8: Infinite Horizon Model: Speed



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

Table 2: Infinite Horizon Model: Performance Results

Number of Gridpoints for (a, h)	Time (seconds)	Number of Iterations till convergence	Maximum Euler error for $(c; i)$	Average Euler error for $(c; i)$
ENDGM				
$(20, 20)$	0.22	307	$(-2.55;\ -1.62)$	$(-3.16;\ -2.59)$
$(40, 40)$	0.70	308	$(-3.12;\ -2.28)$	$(-3.75;\ -3.17)$
$(100, 100)$	4.02	308	$(-4.01;\ -2.82)$	$(-4.89;\ -3.87)$
$(200, 200)$	16.99	308	$(-4.37;\ -3.50)$	$(-5.27;\ -4.39)$
HEGM				
$(20, 20)$	0.41	318	$(-2.77;\ -1.96)$	$(-3.41;\ -2.73)$
$(40, 40)$	1.39	316	$(-3.20;\ -2.11)$	$(-3.94;\ -3.17)$
$(100, 100)$	7.97	320	$(-4.12;\ -2.76)$	$(-4.94;\ -3.78)$
$(200, 200)$	31.17	320	$(-4.64;\ -3.39)$	$(-5.36;\ -4.27)$
EXGM				
$(20, 20)$	0.69	346	$(-2,87;\ -1,96)$	$(-3.53;\ -2.73)$
$(40, 40)$	3.03	379	$(-3,30;\ -2,11)$	$(-4.02;\ -3.16)$
$(100, 100)$	21.03	407	$(-4,14;\ -2,76)$	$(-4.93;\ -3.79)$
$(200, 200)$	86.94	418	$(-4,66;\ -3,39)$	$(-5.37;\ -4.27)$

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 (EXGM), Carroll's method of endogenous grid-points (ENDGM), cf. Carroll (2006), and a hybrid method (HEGM)—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 finite and infinite horizon settings. We show that applying the endogenous grid method in more than one dimension gives rise to irregular grids. We apply Delaunay methods to interpolate on these irregular grids. Despite this more complex interpolation, ENDGM and HEGM clearly outperform EXGM. ENDGM dominates HEGM for small to medium sized problems (in terms of number of grid-points) whereas HEGM dominates for a large number of grid-points. For a standard choice of numbers of grid-points, ENDGM is 1.6 to 1.8 times faster than HEGM.

Two additional remarks on ENDGM and HEGM are in order. First, 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 HEGM uses analytical solutions in only one dimension and standard numerical methods in others, it's relative advantage decreases in the dimensionality of the problem.

In this paper, however, we restrict attention to two dimensional problems. Extensions to higher dimensions are left for future research.

6 Appendix

To derive first-order conditions, we focus on the infinite horizon setting. As is standard in the literature, we denote next period values by symbol $'$.

The dynamic version of the household problem reads as

$$V(a,h) = \max_{c,i,a',h'} \{u(c) + \beta s\left(h'\right) V'(a',h')\}$$

subject to

$$\begin{aligned} a' &= R\left(a + wh - c - i\right) \\ h' &= (1-\delta)\left(h + f\left(i\right)\right) \\ a' &\geq 0. \end{aligned}$$

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

$$\frac{\partial V\left(a,h\right)}{\partial c} = u_c - \beta s\left(h'\right) V'_a R - R\mu \overset{!}{=} 0 \;\Leftrightarrow\; u_c - \beta s\left(h'\right) R V'_a = R\mu, \tag{5}$$

$$\begin{aligned} \frac{\partial V\left(a,h\right)}{\partial i} &= s_h\left(h'\right)(1-\delta) f_i \beta V' + s\left(h'\right)\beta\left[V'_a(-R) + V'_h(1-\delta) f_i\right] - R\mu \overset{!}{=} 0 \\ &\Leftrightarrow s_h\left(h'\right)(1-\delta) f_i \beta V' + s\left(h'\right)\beta\left[V'_a(-R) + V'_h(1-\delta) f_i\right] = R\mu \end{aligned} \tag{6}$$

and

$$\begin{aligned} a' &\geq 0 \\ \mu &\geq 0 \\ a'\mu &= 0. \end{aligned}$$

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

$$\begin{aligned} u_c - \beta s\left(h'\right) R V'_a &= 0 \\ s_h\left(h'\right)(1-\delta) f_i \beta V' + s\left(h'\right)\beta\left[V'_a(-R) + V'_h(1-\delta) f_i\right] &= 0. \end{aligned}$$

Rearranging gives

$$
\begin{aligned}
u_c &= \beta s\left(h'\right) V_a' R \\
f_i &= \frac{R}{(1-\delta)} \frac{s\left(h'\right) V_a'}{s_h\left(h'\right) V' + s\left(h'\right) V_h'}.
\end{aligned}
$$

Case 2: Corner Solution—Binding Borrowing Constraint

In the second case the borrowing constraint is binding so that $a' = 0$ and $\mu > 0$. From (5) and (6) it then follows that

$$
u_c = s_h\left(h'\right) \beta V' + s\left(h'\right) \beta V_h' (1-\delta) f_i \tag{7}
$$

and

$$
\begin{gathered}
u_c = \beta (1-\delta) f_i \left[s_h\left(h'\right) V' + s\left(h'\right) V_h'\right] \\
a' = 0 \;\Leftrightarrow\; c = a + wh - i.
\end{gathered}
$$

Making use of our assumptions on functional forms, equation (7) reduces in EXGM and HEGM to

$$
\begin{aligned}
&(a + wh - i)^{-\theta} - \frac{1}{\left[(1-\delta)\left(h + \frac{i^{1-\alpha}}{1-\alpha}\right)\right]^2} V'\left[0, (1-\delta)\left(h + \frac{i^{1-\alpha}}{1-\alpha}\right)\right] \beta (1-\delta) i^{-\alpha} \\
&- \left[1 - \frac{1}{\left[(1-\delta)\left(h + \frac{i^{1-\alpha}}{1-\alpha}\right)\right]}\right] \left[V_h'\left[0, (1-\delta)\left(h + \frac{i^{1-\alpha}}{1-\alpha}\right)\right]\right] \beta (1-\delta) i^{-\alpha} = 0
\end{aligned}
$$

and in ENDGM to

$$
\begin{aligned}
&\left(a + w\frac{h'}{1-\delta} - i^{\alpha} - i\right)^{-\theta} - \frac{1}{\left[h'\right]^2} \beta V'\left[0, h'\right] (1-\delta) i^{-\alpha} \\
&- \left[1 - \frac{1}{1+h'}\right] \beta \left[V_h'\left[0, h'\right]\right] (1-\delta) i^{-\alpha} = 0.
\end{aligned}
$$

Observe that this equation is not linear in i. 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

$$
\begin{aligned}
\frac{\partial V\left(x,h\right)}{\partial a} &\equiv V_a = \beta V_a' R + R\mu = u_c \\
\frac{\partial V\left(x,h\right)}{\partial h} &\equiv V_h \\
&= \beta s_h\left(h'\right) V(a', h')\left(1-\delta\right) + \beta s\left(h'\right) V_a(a', h') w R + \beta s\left(h'\right) V_h(a', h')\left(1-\delta\right) + R\mu \\
&= \left(w + \frac{1}{f_i}\right) u_c.
\end{aligned}
$$

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.

Brumm, J. and M. Grill (2010). Computing equilibria in dynamic models with occasionally binding constraints. Technical report, University of Mannheim.

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.

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.

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.

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., B. Flannery, S. Teukolsky, and W. Vetterling (1992). *Numerical Recipes in FORTRAN 77: Volume 1, Volume 1 of Fortran Numerical Recipes: The Art of Scientific Computing*. Fortran Numerical Recipes. Cambridge University 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.