A generalized endogenous grid method for discrete-continuous choice

Fedor Iskhakov* John Rust† Bertel Schjerning‡

16th February 2012

PRELIMINARY DRAFT, PLEASE DO NOT QUOTE WITHOUT PERMISSION

Abstract

This paper extends Carroll's endogenous grid method (2006 "The method of endogenous gridpoints for solving dynamic stochastic optimization problems", Economic Letters) for models with sequential discrete and continuous choice. Unlike existing generalizations, we propose solution algorithm that inherits both advantages of the original method, namely it avoids all root finding operations, and also efficiently deals with restrictions on the continuous decision variable. To further speed up the solution, we perform the inevitable optimization across discrete decisions as more efficient computation of upper envelope of a set of piece-wise linear functions. We formulate the algorithm relying as little as possible on a particular model specification, and precisely define the class of dynamic stochastic optimal control problems it can be applied to.

We illustrate our algorithm using finite horizon discrete sector choice model with consumption-savings decisions and borrowing constraints, and show that in comparison to the traditional approach the proposed method runs at least an order of magnitude faster to deliver the same precision of the solution.

To implement the method we develop a generic software package that includes pseudo-language for easy model specification and computational modules which support both shared memory and cluster parallelization. The package is wrapped in a Matlab class and incurs low start-up cost to the user. The software package is accessible in public domain.

Keywords: Discrete and continuous choice, dynamic structural model, consumption and savings, discrete labour supply.
JEL codes: C63

1 Introduction

Solving dynamic stochastic optimization problems is rarely feasible analytically, and numerical solutions are usually computationally intensive. Traditionally numerical solutions are obtained through backward induction (or value function iterations) which require solving for the optimum value of control in each time period and in each point of the state space of the original problem. The state space is usually discretized beforehand with an exogenously fixed grid. Endogenous grid method (EGM) proposed by Carroll [2006] allows to avoid internal optimizations by letting the grid over the state space to vary from one time period to another. The essence of the EGM method is to guess an optimal value of control and back out the values of the state variables for which the guess would indeed be optimal. Repeating this procedure until the whole state space is sufficiently explored in each time period gives the same

*Corresponding author. Center of Study of Choice (CenSoC), Faculty of Business, University of Technology, Sydney Level 4, 645 Harris St. Ultimo, NSW 2007 `fedor.iskhakov@uts.edu.au`

†Department of Economics, University of Maryland, College Park, MD 20742, 4115 Tydings Hall

‡Department of Economics, University of Copenhagen and Centre for Applied Microeconometrics (CAM), Øster Farimagsgade 5, building 26, DK-1353 Copenhagen K, Denmark

mapping of points of the state space to the optimal decisions as in the traditional solution framework. Yet, replacing iterative optimization routine with a single shot algorithm leads to substantial decrease in run-time, especially in large scale problems.

Carroll [2006] presents EGM as a solution method for a standard problem of maximizing discounted utility of consumption subject to intertemporal budget constraint, for which he provides both microeconomic and macroeconomic interpretations. In micro interpretation future consumption is uncertain due to shocks to labour income, and the budget constraint reflects the dynamics of liquid assets with positive returns on savings and in presence of borrowing constraint. In representative agent macro interpretation future consumption is uncertain due to shocks to aggregate productivity, and budget constraint reflects the dynamics of depreciating capital. In both interpretations the model boils down to an optimal control problem with one continuous state and one continuous decision variables.

The purpose of this paper is to generalize EGM method to make it suitable for a lager class of dynamic stochastic optimization problems, namely those which include additional discrete and continuous state variables and more importantly additional discrete choices. The intended application for our modification of EGM is a more involved microeconometric model with discrete labour supply choices accompanying the consumption-savings process covered by Carroll's micro interpretation. We use a simple model from this class in section 3 for numerical illustration of our solution algorithm.

EGM has already been generalized for the case of additional state variables and additional controls by Barillas and Fernandez-Villaverde [2007] who propose an iterative scheme in which EGM step is alternated with traditional value function iteration step. During EGM step the objective function is optimized only with respect to one continuous decision variable while keeping all other policy functions (decision rules) fixed. The latter are updated on the value function iteration step. Barillas and Fernandez-Villaverde use standard representative agent neoclassical growth model with endogenous continuous labour supply to illustrate their method, and show the speed up of the solution to be of two orders of magnitude compared to the traditional approach.

Our generalization of EGM although similar in spirit, differs from [Barillas and Fernandez-Villaverde, 2007] in several important aspects. First, Barillas and Fernandez-Villaverde tailor their algorithm for infinite horizon problems, which allows them to use EGM directly in place of value function iterations in the search for the fixed point that solves the Bellman equation. Restricting attention to infinite horizon problems allows for an easy guess of the very efficient initial point — analytic steady state solution for the deterministic version of the model. As Barillas and Fernandez-Villaverde note[1], for some models (including the one they use to illustrate their method) running EGM with one control fixed to its deterministic steady state value, and extending the calculation with just a single value function iteration afterwards may give nearly as accurate solution as the traditional method. In contrast, our method is developed with finite horizon problem in mind, thus there is no gain in the efficiency of the method due to a possibly better initial guess for the value function. Calculations start at terminal period from the same point (terminal utilities) as they would in the traditional solution algorithm. Yet, our method can be easily extended to infinite horizon.[2]

Second, Barillas and Fernandez-Villaverde [2007] can not avoid root-finding operations completely. Besides optimizations with respect to some decision variables which are inevitable in value function iterations, in their method a non-linear equation has to be numerically solved during each transition between the two types of iterations. In our algorithm we make use of the fact that value function iterations when all decision variables are discrete are not only faster because internal optimization tasks reduce to calculating maximums across finite sets of points, but also they can be "vectorized" when upper envelope of a set of decision-specific value functions is calculated instead. The gain in computational speed is due to the fact that not all decision-specific value functions have to be compared in each state point in order to calculate the upper envelope, in addition we develop an efficient algorithm for combining different grids these value functions are defined over (due to the use of EGM) which avoids multiple re-interpolations. In effect, our algorithm avoids all root-finding operations, except the complicated cases of utility maximization in terminal period[3].

Fella [2011] provides a generalization of the endogenous grid point method for non-concave problems which includes the specification we consider here. Fella develops the method to solve the model of

[1] Section 3.2, page 2701.

[2] Although the question of choice of the initial guess for policy functions will then arise.

[3] When terminal utility is increasing monotonically, this maximization is trivial because in presence of borrowing constraint consumption is bounded from above.

durable goods purchases with switching costs. The key idea of the method is to identify the regions of the state space where the problem is non-concave and where standard EGM conditions are not sufficient, and use the approach similar to Barillas and Fernandez-Villaverde [2007] only on those regions. Fella argues (Fella, 2011, p.9) that "since in many problems the non-concave region is a, possibly small, subset of the asset grid, this reduces the set of points at which one has to use a, substantially slower, global method". Contrary to this argument, our method is designed for the problems where non-concavity regions are large, most likely extending to the whole state space.

Another feature that distinguishes our methods from Fella [2011] is the treatment of credit constrains. The model considered by Fella has endogeniezed credit constraint that never binds, whereas we treat credit constraint in the spirit of the original Carroll's method, and succeed in preserving full flexibility and ease in dealing with credit constraints.

Finally, both Fella [2011] and Barillas and Fernandez-Villaverde [2007] center the presentation of their method around a particular specification of the model. In this paper we make special effort to separate the presentation of the solution method from the details of a particular economic model we had in mind while developing it. Thus, section 2 is written primarily using terms from optimal control theory, and contains detailed discussion of the application domain for the method we develop. We turn to economic model of discrete labour supply with consumption, savings and borrowing constraint in section 3 to illustrate our algorithm, and then conclude.

2 Generalized EGM algorithm for discrete-continuous problems

2.1 Structure of the model to be solved

The generalized EGM algorithm is designed to solve a particular dynamic stochastic optimal control problem in discrete time with one continuous and one discrete decision variables. In this section we present the model is most general form to accurately define the application domain of the method, and in subsequent sections give it a more detailed economic specification.

The generalized EGM algorithm solves optimization problems of the form

$$max_{\delta\in\mathfrak{F}}\left\{E\left[\sum_{t=T_0}^{T}\prod_{\tau=T_0}^{t}\beta_\tau\cdot u\left(c_t,d_{t-1},st_t\right)\right]\right\} \tag{1}$$

$$\begin{aligned}
\text{s.t. } M_{t+1} &= \mathcal{R}_{t+1}\left(A_t,d_{t-1},st_t,d_t,st_{t+1},\xi_{t+1}\right) && (2)\\
M_t &= A_t+c_t && (3)\\
A_t &\geqslant A_0, && (4)
\end{aligned}$$

where notations are as follows:

$u\left(c_t,d_{t-1},st_t\right)$ is an instantaneous utility at period t, which is dependent on consumption $c_t=M_t-A_t$ (3), discrete decision d_{t-1} and various state variables st_t;

M_t is a singled out continues state variable which is given special role in the solution algorithm, with standard microeconomic interpretation of total resources (money-at-hand) available for consumption in period t;

$A_t\in\mathbb{R}^1$ is the scalar continues decision variable which is interpreted as *end-of-period* resources remaining after within period consumption (in accordance to (3)), which is subject to credit constraint (4);

$d_t\in\left\{d^{(1)},..,d^{(D)}\right\}$ is the scalar discrete decision variable which has D possible values, d_{T_0-1} is known and fixed;

$\delta=\{\delta_{T_0},..,\delta_T\}$ is a set of decision rules (policy functions)

$$\delta_t:\begin{cases}(M_t,d_{t-1},st_t)\to(A_t,d_t), & T_0\leqslant t<T,\\ (M_T,d_{T-1},st_T)\to A_T, & t=T,\end{cases}$$

which map points of the state space into choices at each time period, and are jointly chosen from the class of feasible decision rules $\mathfrak{F}$;

β_τ is exogenous discount factor which may be time dependent, for example corrected for agent's life expectancy; finally,

$\mathcal{R}_{t+1}(A_t, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1})$ is intertemporal budget constraint which describes how the next period total resources M_{t+1} depend on current period savings A_t given the transition from st_t to st_{t+1} and discrete decisions d_{t-1}, d_t in current and previous period. ξ_{t+1} is a random shock in period $t+1$ that effects the budget constraint.

A standard microeconomic budget constraint may be given with $\mathcal{R}_{t+1}(A_t, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}) = (1+r)A_t + \xi_{t+1}$, where r is a return rate on savings and ξ_{t+1} is a stochastic income. It is also straightforward to verify that with different definitions for $\mathcal{R}_{t+1}(A_t, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1})$ problem (1) nests both micro and macro specifications in [Carroll, 2006].

The solution method relies on the following properties *assumed* about this problem:

A1. Instantaneous utility $u(c, d_{t-1}, st_t)$ is differentiable and has monotonic derivative with respect to c;

A2. Decision space contains one scalar continuous choice variable A_t and one scalar discrete choice variable d_t [4];

A3. The structure of the constrains (2-4) holds, thus singling out one continuous state variable M_t for which the stochastic motion rule is given by (2) and which occurrence in the utility function is only through $c_t = M_t - A_t$;

A4. The decision A_t is bound between A_0 and M_t to ensure through (3) $c_t \geq 0$;

A5. The decision d_t affects the principals of the model, namely the utility $u(c_t, d_{t-1}, st_t)$ and stochastic income $y_{t+1}(d_t, st_{t+1})$ with a one period lag, or under alternative interpretation decision is made in the end of each period about the next period; d_{T_0-1} is known and fixed;

A6. Transition probabilities (densities) for the state process $\{st_t\}$ used in calculating the expectation in (1) are independent of M_t, namely $P(st_{t+1}|st_t, M_t, A_t, d_t) = P(st_{t+1}|st_t, A_t, d_t)$.

2.2 General layout of the algorithm

According to the principle of optimality, the problem (1) can be written in recursive form as

$$V(M_t, d_{t-1}, st_t) = \begin{cases} max_{\delta_t} \{u(M_t - A_t, d_{t-1}, st_t) + \beta_{t+1} E[V(\mathcal{R}_{t+1}(A_t, .., \xi_{t+1}), d_t, st_{t+1})]\}, & T_0 \leqslant t < T, \\ max_{\delta_t} \{u(M_t - A_t, d_{t-1}, st_t)\}, & t = T, \end{cases}$$
$$\text{s.t. } A_t \geqslant A_0,\ d_{T_0-1} \text{ fixed,} \tag{5}$$

where $V(M_t, d_{t-1}, st_t)$ is the value function. In the absence of discrete component in the decision rule δ_t first order conditions for (5) in combination with envelope theorem would lead to the single Euler equation that constitutes the foundation of EGM in the original paper [Carroll, 2006].

But because the maximand in (5) is not differentiable with respect to d_t, it is not possible to apply Carroll's argument in full and derive the set of Euler equations to characterize the solution to the problem (1). Instead, we rely on Theorem 3 in Clausen and Strub which states that when utility function is differentiable with respect to consumption (assumption A.1) the value function is differentiable and the Euler equation with respect to continuous control variable holds as long as no constraints bind. The intuition under their result is the following. Because value function $V(M_t, d_{t-1}, st_t)$ is in fact an upper envelope of the discrete choice specific value functions

$$V^{d_t}(M_t, d_{t-1}, st_t) = max_{A_t(M_t, d_{t-1}, st_t)} \{u(M_t - A_t, d_{t-1}, st_t) + \beta_{t+1} E[V(\mathcal{R}_{t+1}(A_t, .., \xi_{t+1}), d_t, st_{t+1})]\}$$

(for $T_0 \leqslant t < T$), and as long as there are no kinks in $V^{d_t}(M_t, d_{t-1}, st_t)$ caused by binding constraints, it may only have downward pointing kinks which can not be even local maximums. In other words, the decision making agent is never indifferent between the discrete choices in the optimum.

[4] Because vector values of any multinomial discrete decision can always be "re-coded" into the single set of values, there is no loss of generality in assuming d_t to be scalar.

Therefore we base our generalization on the fact that Euler equation with respect to the continuous control remains the necessary condition for optimality, and thus the core steps of EGM can be carried through conditional on the discrete choices. The Euler equation with respect to continuous control can be derived in the usual way. For problem (1) it takes the form

$$\begin{cases} \frac{\partial u}{\partial c}\left(c_t, d_{t-1}, st_t\right) = \beta_{t+1} E\left[\frac{\partial \mathcal{R}_{t+1}}{\partial A_t} \cdot \frac{\partial u}{\partial c}\left(c_{t+1}, d_t, st_{t+1}\right) | A_t, st_t\right], & T_0 \leqslant t < T, \\ \frac{\partial u}{\partial c}\left(c_t, d_{t-1}, st_t\right) = \frac{\partial u}{\partial c}\left(c_T, d_{T-1}, st_T\right) = 0, & t = T, \end{cases} \tag{6}$$

where the arguments of the intertemporal budget constraint function $\mathcal{R}_{t+1}\left(A_t, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}\right)$ are dropped to simplify exposition.

The general layout of the algorithm is presented in Table 1; each column represents one of a series of nested loops, and each row represents a distinct operation within a corresponding loop. When $t = T$ (upper panel in the table) there is no consequent period, thus the optimal discrete decision is arbitrary, and the problem can be reformulated as a static maximization of utility of consumption under borrowing constraint. For all utility functions monotonically increasing in consumption ($\frac{\partial u}{\partial c}\left(c_t, d_{t-1}, st_t\right) > 0$) the optimal consumption and savings at the terminal period are $c_T = M_T - A_0$ and $A_T = A_0$. In such case the only equation to be solved numerically is eliminated and the method avoids root finding operations completely.

The lower panel of Table 1 contains a set of nested loops that have to be run in order to find optimal behavior in all the rest of time periods. The outer-most and the next loop are standard in backwards induction solution approach for Markovian decision problems[5], but the content of the latter is specific for our generalization of EGM.

The main principles are the following. Core EGM step is performed conditional on each current period discrete decision d_t to produce the d_t-specific endogenous grid $M_t^{grid}(d_t, d_{t-1}, st_t) \in \mathbb{R}^{n(d_t, d_{t-1}, st_t)}$ and optimal consumption rule $c_t\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ defined over this grid. The standard EGM step is augmented with the calculation of the d_t-specific value functions $V^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$, performed alongside at little extra cost. It is worth noting that instead of the notion of a fixed grid over end of period total resources A_t we adopt the notion of a sequence of guesses of A_t, which are taken one by one and fed into the EGM step. This is important because the proper sequence of guesses may differ for different d_t, and therefore we develop an adaptive algorithm to generate it (which is presented in Appendix C).

Once EGM step is performed for each value of current discrete decision d_t, d_t-specific value functions $V^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ must be compared to reveal the ranges of M_t where each discrete option is optimal. This comparison may be hard and time consuming due to the fact that the functions to be compared are defined over generally unknown d_t-specific grids $M_t^{grid}(d_t, d_{t-1}, st_t)$. One extreme case is when for some d_t' and d_t'' the grids do not overlap, i.e. $max\left\{M_t^{grid}(d_t', d_{t-1}, st_t)\right\} < min\left\{M_t^{grid}(d_t'', d_{t-1}, st_t)\right\}$. The adoptive algorithm for generating the sequences of A_t mentioned above is designed to rule out this case and ensure that all endogenous grids overlap in the range of the initial grid over M_t chosen at $t = T$.

The computational burden of comparison of d_t-specific value functions $V^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ is also due to the fact that functions of interest are defined over different grids. The brute force approach implied in previous papers [Barillas and Fernandez-Villaverde, 2007, Fella, 2011] requires, first, that period t unified endogenous grid $M_t^{grid}(d_{t-1}, st_t)$ is fixed, second, that all value functions at the nodes of this new grid are interpolated, and third, that maximum is found at each point. Additional steps are required to compute the exact boundaries of the regions of optimality for each of the discrete decisions d_t. Instead we adopt an algorithm for computation of the the upper envelope of piece-wise linear functions, which works across the whole range of M_t thus "vectorizing" the comparison. The exact values of switching between different discrete decisions are also found naturally in the computation of upper envelope. Theorem 2 in Pach and Sharir [1989] provides the upper bound on the number of linear segments in the upper envelope, namely $O\left\{D \cdot (n-1) \cdot \alpha\left(D \cdot (n-1)\right)\right\}$, where $\alpha\left(\bullet\right)$ is extremely slowly growing inverse

[5] Here previous period discrete decision d_{t-1} is treated as state variable in period t.

Table 1: Schematic algorithm layout

<table>
<tr><td rowspan="3">In the last period when $t = T$</td><td rowspan="2">For each (d_{T-1}, st_T)</td><td colspan="3">Choose some initial grid over M_T</td></tr>
<tr><td colspan="3">Compute optimal consumption c_T using second line in Euler equation (6) and restriction $0 \leqslant c_T \leqslant M_T - A_T$</td></tr>
<tr><td colspan="4">Output optimal consumption and savings functions $c_T\left(M_T, d_{T-1}, st_T\right)$ and $A_T\left(M_T, d_{T-1}, st_T\right) = M_T - c_T\left(M_T, d_{T-1}, st_T\right)$, as well as value functions $V\left(M_T, d_{T-1}, st_T\right) = u\left(c_T, d_{T-1}, st_T\right)$ for each (d_{T-1}, st_T), and proceed with next iteration of t</td></tr>
<tr><td rowspan="8">For each t from $T-1$ to T_0</td><td rowspan="7">For each (d_{t-1}, st_t)</td><td rowspan="5">For each d_t</td><td rowspan="4">For a sequence of guesses A_t (see pseudo-code in Appendix C)</td><td>Compute left hand side of the Euler equation $E\left[\frac{\partial \mathcal{R}_{t+1}}{\partial A_t} \cdot \frac{\partial u}{\partial c}\left(c_{t+1}, d_t, st_{t+1}\right) \middle| A_t, st_t\right]$ and expected value function $E\left[V\left(\mathcal{R}_{t+1}, d_t, st_{t+1}\right) \middle| A_t, st_t\right]$ using
• transition probabilities for the state process $P\left(st_{t+1} \middle| st_t, A_t, d_t\right)$
• probability distribution of shock ξ_{t+1}
• optimal consumption $c_{t+1}(M_{t+1}, d_t, st_{t+1})$ computed on the previous iteration of t
• value function $V\left(M_{t+1}, d_t, st_{t+1}\right)$ computed on the previous iteration of t</td></tr>
<tr><td>Compute optimal consumption c_t at t for given A_t using Euler equation (6) and the inverse of the marginal utility function</td></tr>
<tr><td>Add another point $M_t = A_t + c_t$ to the (decision specific) endogenous grid $M_t^{grid}(d_t, d_{t-1}, st_t)$</td></tr>
<tr><td>Compute d_t-specific value function at the new grid point $V^{d_t}\left(M_t, d_{t-1}, st_t\right) = u\left(c_t, d_{t-1}, st_t\right) + \beta_{t+1} E\left[V\left(M_{t+1}, d_t, st_{t+1}\right) \middle| A_t, st_t\right]$</td></tr>
<tr><td colspan="2">Save to memory calculated d_t-specific optimal consumption function $c_t^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ and d_t-specific value function $V^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ defined over decision specific grid $M_t^{grid}(d_t, d_{t-1}, st_t)$</td></tr>
<tr><td colspan="3">Compute the upper envelope of decision specific value functions $V^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ while simultaneously constructing unified endogenous grid $M_t^{grid}(d_{t-1}, st_t)$ and optimal discrete decision rule $d_t(M_t, d_{t-1}, st_t)$
(see pseudo-code in Appendix B)</td></tr>
<tr><td colspan="3">Using optimal discrete decision rule and corresponding d_t-specific optimal consumption functions find unified optimal consumption function $c_t\left(M_t^{grid}(d_{t-1}, st_t), d_{t-1}, st_t\right)$ and value function $V\left(M_t^{grid}(d_{t-1}, st_t), d_{t-1}, st_t\right)$</td></tr>
<tr><td colspan="4">Output optimal consumption and savings functions $c_t\left(M_t, d_{t-1}, st_t\right)$ and $A_t\left(M_t, d_{t-1}, st_t\right) = M_t - c_t\left(M_t, d_{t-1}, st_t\right)$, optimal discrete decision rule $d_t(M_t, d_{t-1}, st_t)$ and value functions $V\left(M_t, d_{t-1}, st_t\right)$ for each (d_{t-1}, st_t), and proceed with next iteration of t</td></tr>
</table>

Ackermann function[6] and n is the largest number of points in the endogenous grids.[7]

Our algorithm for upper envelope computation (see pseudo-code in Appendix B) is based on the idea of re-utilizing the existing grid points to avoid some interpolations and the insight that many comparisons can be skipped for inferior functions. The algorithm achieves linear running time.

Finally, unified optimal consumption rule $c_t\left(M_t^{grid}(d_{t-1}, st_t), d_{t-1}, st_t\right)$ and the value function $V\left(M_t^{grid}(d_{t-1}, st_t), d_{t-1}, s\right.$ (in next to last row in Table 1) are also constructed along the corse of the upper envelope calculation.

2.3 Borrowing constraints

Borrowing constrains present both theoretical and numerical problems, but similarly to the original EGM by Carroll [2006] our generalized method deals very effectively with both of them.

The theoretical difficulty is due to the fact that Euler equation (6) is only necessary as long as the constraint $A_t \geqslant A_0$ is relaxed. Carroll [2006] deals with it by running the EGM iteration for the guess $A_t = A_0$ and finding the threshold M_0 where the decision maker is at the verge of becoming credit constrained. All points $M_t > M_0$ where the constraint is relaxed are reconstructed in the EGM step using Euler equation, and for all points $M_t < M_0$ the credit constraint binds implying $A_t = A_0$ and $c_t = M_t - A_0$ from (3). So, if optimal consumption is graphed against M_t, Carroll [2006] simply connects the left-most point recovered in EGM step to the point $(A_0, 0)$.

Our generalized EGM applies exactly the same approach when it comes to the calculation of optimal consumption rule $c_t^{d_t}\left(M_t^{grid}(d_t, d_{t-1}, st_t), d_{t-1}, st_t\right)$ during the EGM step. The first point in d_t-specific endogenous grid $M_t^{grid}(d_t, d_{t-1}, st_t)$ is manually set to be A_0 with corresponding consumption $c_t^{d_t}(A_0, d_{t-1}, st_t) = 0$.

Numerical difficulties arise when d_t-specific value functions $V^{d_t}(M_t, d_{t-1}, st_t)$ are calculated on $M_t^{grid}(d_t, d_{t-1}, st_t)$ in the proximity of A_0. For any utility function that assigns negative infinity to zero consumption the value functions become increasingly steep at the left side of the grid, and their piece wise linear approximations become increasingly rough. As a result, the upper envelope of these approximations becomes excessively complex on the left end of the grid, and the algorithm reports lots of switching between different discrete decisions on very small intervals close to A_0. However, we are able to suppress this numerical noise using the following property of the d_t-specific value functions $V^{d_t}(M_t, d_{t-1}, st_t)$.

Denote $M_0^{d_t}(d_{t-1}, st_t)$ the level of total resources that is returned by the EGM step called with A_0, to the left of which the decision maker is credit constrained when making choice d_t. Denote $EV_0^{d_t}(d_{t-1}, st_t)$ the expected choice specific value function corresponding to no savings ($A_t = A_0$), i.e.

$$EV_0^{d_t}(d_{t-1}, st_t) = E\left[V\left(\mathcal{R}_{t+1}\left(A_0, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}\right), d_t, st_{t+1}\right)\right].$$

Then for $M_t < M_0^{d_t}(d_{t-1}, st_t)$ the d_t-specific value function $V^{d_t}(M_t, d_{t-1}, st_t)$ is given by

$$V^{d_t}(M_t, d_{t-1}, st_t) = u(M_t - A_0, d_{t-1}, st_t) + \beta_{t+1} EV_0^{d_t}(d_{t-1}, st_t). \tag{7}$$

Note that the first term in the right hand side of (7) is independent of d_t and the first term in independent of M_t. The latter implies that for any $M_t < M_0^{d_t}(d_{t-1}, st_t)$ d_t-specific value function $V^{d_t}(M_t, d_{t-1}, st_t)$ can be calculated exactly by adding a constant $\beta_{t+1} EV_0^{d_t}(d_{t-1}, st_t)$ to the utility function. But moreover, the former implies that the collection of d_t-specific value functions $V^{d_t}(M_t, d_{t-1}, st_t)$ in the proximity of A_0 appears to be a collection of utility functions with different vertical shifts $\beta_{t+1} EV_0^{d_t}(d_{t-1}, st_t)$. In other words, d_t-specific value functions $V^{d_t}(M_t, d_{t-1}, st_t)$ do not intersect for any $M_t < M_0(d_{t-1}, st_t) = min_{d_t}\left\{M_0^{d_t}(d_{t-1}, st_t)\right\}$.

Our algorithm for upper envelope computation acknowledges these properties, namely it disregards the interval $(A_0, M_0(d_{t-1}, st_t))$ completely, and is capable of computing analytical d_t-specific value functions on the intervals $\left(M_0(d_{t-1}, st_t), M_0^{d_t}(d_{t-1}, st_t)\right)$ to avoid any numerical noise. The resulting upper envelope $V(M_t, d_{t-1}, st_t)$ (in the third row and next to last row in Table 1) is given with a piecewise

[6] For all practical purposes it can be assumed that $\alpha(D(n-1)) < 5$.

[7] Algorithm 1 and Theorem 2.1 in [Edelsbrunner et al., 1989] demonstrates how this boundary is achieved for piece-wise linear planes with respect to both time and storage.

linear approximation $V\left(M_t^{grid}(d_{t-1}, st_t), d_{t-1}, st_t\right)$ to the right of the threshold $M_0(d_{t-1}, st_t)$, and can be calculated exactly to the left of it using

$$V\left(M_t, d_{t-1}, st_t\right) = u\left(M_t - A_0, d_{t-1}, st_t\right) + \beta_{t+1} max_{d_t}\left\{EV_0^{d_t}(d_{t-1}, st_t)\right\}. \quad (8)$$

The use of analytical form of value function for $M_t < M_0(d_{t-1}, st_t)$ on the next t iteration additionally increase the accuracy of the solution.

2.4 Parallelization

Our generalization of EGM allows for efficient parallelization on the most computationally intensive level of the algorithm.

The general scheme for parallelization of Markovian backwards induction computations is to parallelize the within time period operations, and then gather and redistribute to all nodes the results of each time period iteration. In other words, the parallelization is across state space, with synchronization in the end of each time period iteration. This is necessary because entities to be computed in time period t in general may depend on the value in all points of the state space in period $t+1$.

Our algorithm is compatible with this general scheme, and is parallelized on the level of second column in Table 1 with synchronization at the end of each time period iteration.

Deeper parallelization is less obvious. Even though the next nested loop over d_t can be parallelized, the consequent upper envelope algorithm is inherently sequential.[8] Yet, because the dimensionality of the combined state space (d_{t-1}, st_t) is the main factor of computation complexity, good scalability of the method is likely to be seen in practical applications without any deeper parallelization.

3 Numerical example

3.1 Optimal consumption/savings with discrete labour supply

We illustrate the proposed generalized EGM algorithm for discrete-continuous problems with the following microeconometric model of discrete labour supply, savings and credit constraints.

Let (1) describe the problem of a rational individual maximizing her expected lifetime utility by making each period the decisions of whether to work in public, private sector or become an entrepreneur. Let $d_t \in \{0, 1, 2\}$ denote these three options. Let $A_0 = 0$, so savings (the continuous control) are restricted to $A_t \in [0, M_t] \subset \mathbb{R}^1$. In this micro model the total resources M_t are interpreted as money-at-hand at period t composed of precious period savings A_{t-1} and current period incomes $y_{t+1}(d_t, st_{t+1})$ according to intertemporal budget constraint $\mathcal{R}_{t+1}\left(A_t, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}\right) = (1+r)A_t + w\left(d_t, \xi_{t+1}\right)$, and used for current period consumption c_t and savings A_t carried on to next period according to (3). r is time invariant interest rate, and β_t is the product of the discount factor and the probability of survival from period $t-1$ to period t.

We assume that intertemporal utility function takes the form

$$u\left(c_t, d_{t-1}, st_t\right) = \begin{cases} \frac{c_t^{1-\gamma}-1}{1-\gamma} + \phi\left(d_{t-1}\right), & \gamma \neq 1, \\ log(c_t) + \phi\left(d_{t-1}\right), & \gamma = 1, \end{cases}$$

where parameter γ is CRRA coefficient with respect to consumption, and $\phi\left(d_{t-1}\right)$ takes values of 0, 0.2 and 0.15 correspondingly for public, private sector, and self-employment. The values are chosen to reflect that work in private sector is both stressful and does not inspire as own business. All income in this model is earned, and we specify the wage process as

$$w\left(d_{t-1}, \xi_t\right) = \begin{cases} \frac{1}{2}\xi_t & \text{if } d_{t-1} = 0, \\ \frac{1.35}{2}\xi_t & \text{if } d_{t-1} = 1, \\ 0.56 \cdot log(A_{t-1} + 1) & \text{if } d_{t-1} = 2. \end{cases}$$

[8] Although the brute force approach described in section 2.2 is not, which may give grounds for further parallelization.

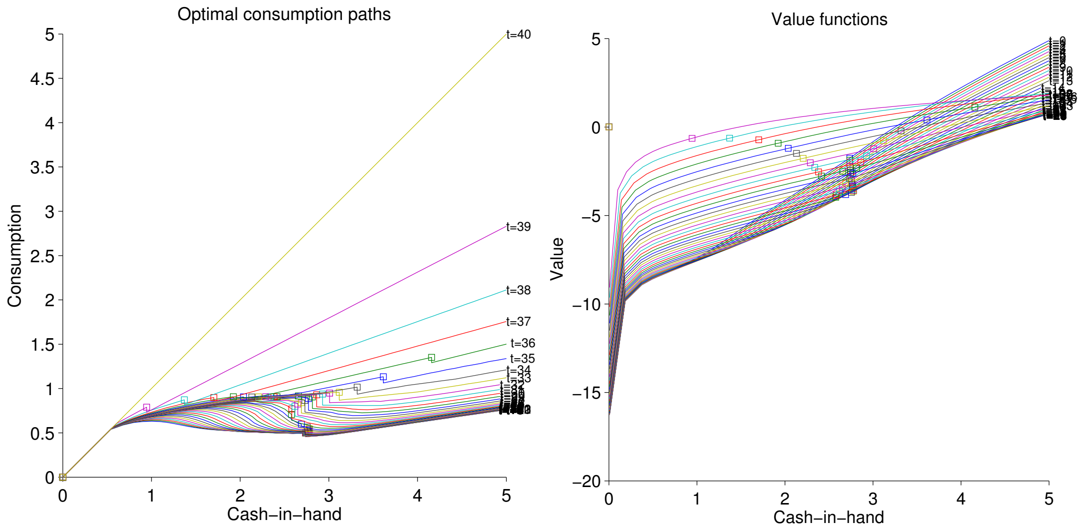

Figure 1: Solution to the numerical example problem

The last case in the expression is a profit function of the entrepreneur, which depends on amount of assets inputed. Parameter and ξ_t is distributed independently across periods with$\xi_t \sim LN\left(-\frac{1}{2}\sigma^2(d_{t-1}), \sigma(d_{t-1})\right)$, where $\sigma(d_{t-1})$ takes values 0.15, 0.35 and 0.75 for the three values of d_{t-1}. Intuition under this specification is the following: public sector provides most secure job with lower pay, private sector provides higher pay but less certainty, and self-employment provides most volatile income.

The state vector in the model is null.

3.2 Solution of the model

Figure 1 depicts the solution to the model. The left panel presents the optimal consumption paths for each time period along with switching points between the discrete decisions. The right panel shown the corresponding value functions.

3.3 Computation time and accuracy of solution

<to be added>

4 Conclusions

<Speed up with no loss of accuracy>

Clear limitation of the proposed method is the restriction to a single continuous decision variable. While other continuous decisions could in principle be discretized and the method forced for the solution of such model, this does not seem an attractive strategy because of the loss of accuracy in discrete representation of the policy function.

<Discuss typical models for which this method is optimal: large number of discrete choices, but not discretized continuous choice (curse of dimensionality kills when accuracy of discretization increases)>

References

Francisco Barillas and Jesus Fernandez-Villaverde. A generalization of the endogenous grid method. *Journal of Economic Dynamics and Control*, 31(8):2698–2712, August 2007. URL `http://ideas.repec.org/a/eee/dyncon/v31y2007i8p2698-2712.html`.

Christopher D. Carroll. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics Letters*, 91(3):312–320, June 2006. URL http://ideas.repec.org/a/eee/ecolet/v91y2006i3p312-320.html.

Andrew Clausen and Carlo Strub. Envelope theorems for non-smooth and non-concave optimization. Preliminary and incomplete, version of February 14, 2011.

Herbert Edelsbrunner, Leonidas Guibas, and Micha Sharir. The upper envelope of piecewise linear functions: Algorithms and applications. *Discrete & Computational Geometry*, 4:311–336, 1989. ISSN 0179-5376. URL http://dx.doi.org/10.1007/BF02187733. 10.1007/BF02187733.

Giulio Fella. A generalized endogenous grid method for non-concave problems. *School of Economics and Finance, Queen Mary University of London Working Paper*, N. 677, 2011.

János Pach and Micha Sharir. The upper envelope of piecewise linear functions and the boundary of a region enclosed by convex plates: Combinatorial analysis. *Discrete & Computational Geometry*, 4:291–309, 1989. ISSN 0179-5376. URL http://dx.doi.org/10.1007/BF02187732. 10.1007/BF02187732.

George Tauchen. Finite state markov-chain approximations to univariate and vector autoregressions. *Economics Letters*, 20(2):177–181, 1986. URL http://ideas.repec.org/a/eee/ecolet/v20y1986i2p177-181.html.

A Pseudo-Code for generalized EGM

$$\begin{cases} \frac{\partial u}{\partial c}\left(c_t, d_{t-1}, st_t\right) = \beta_{t+1} E\left[\frac{\partial \mathcal{R}_{t+1}}{\partial A_t} \cdot \frac{\partial u}{\partial c}\left(c_{t+1}, d_t, st_{t+1}\right) | A_t, st_t\right], & T_0 \leqslant t < T, \\ \frac{\partial u}{\partial c}\left(c_t, d_{t-1}, st_t\right) = \frac{\partial u}{\partial c}\left(c_T, d_{T-1}, st_T\right) = 0, & t = T, \end{cases} \tag{9}$$

1. Start from the terminal period T. For each value of d_{T-1} and st_T choose a grid $M_t^{grid}(d_{T-1}, st_T)$ and solve
$$\frac{\partial u}{\partial c}\left(c_T, d_{T-1}, st_T\right) = 0 \text{ s.t. } 0 \leq c_T \leq M_T$$
to obtain policy function $c_T\left(M_T, d_{T-1}, st_T\right)$. Set
$$V_T\left(M_t^{grid}(d_{T-1}, st_T), d_{T-1}, st_T\right) = u\left(c_T\left(M_T, d_{T-1}, st_T\right), d_{T-1}, st_T\right).$$
The optimal control is
$$A_T^*\left(M_t^{grid}(d_{T-1}, st_T), d_{T-1}, st_T\right) = M_T - c_T\left(M_T, d_{T-1}, st_T\right)$$
with arbitrary $d_T\left(M_T, d_{T-1}, st_T\right)$. This is the base of the backward induction.

2. Assuming that policy function for consumption $c_{t+1}\left(M_{t+1}, d_t, st_{t+1}\right)$ for period $t+1$ and value functions $V_{t+1}\left(M_{t+1}, d_t, st_{t+1}\right)$ are known and defined over grid $M_{t+1}^{grid}\left(d_t, st_{t+1}\right)$, start period t iteration and proceed for each value of d_{t-1}:

3. Iterate over all points st_t in the state space:

4. Iterate over values of current decision d_t:

5. Iterate over a sequence of potential values of continuous control A_t. It is preferable that draws of A_t are made such that the corresponding M_t (to be found in step 8) is contained within certain constant bounds, pseudo-code for one suitable algorithm is presented in Appendix B. One special case of A_t that should certainly be investigated is $A_t = A_0$. For a particular guess $\bar{A}$:

6. Calculate the right hand side of Euler equation (6) using, for example, discrete approximation of the distribution of ξ_{t+1} with a set of K equiprobable points $\{\xi_{t+1}^{(1)},..,\xi_{t+1}^{(K)}\}$[9]. Compute for each $k \in \{1,..,K\}$

$$\begin{aligned} M_{t+1}^{(k)} &= \mathcal{R}_{t+1}\left(\bar{A}, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}^{(k)}\right), \\ c_{t+1}^{(k)} &= c_{t+1}\left(M_{t+1}^{(k)}, d_t, st_{t+1}\right), \\ E\left[\frac{\partial \mathcal{R}_{t+1}}{\partial A_t}\cdot\frac{\partial u}{\partial c}\left(c_{t+1}, d_t, st_{t+1}\right)|\bar{A}, st_t\right] &= \frac{1}{K}\sum_{st_{t+1}}\sum_{k=1}^{K} P\left(st_{t+1}|st_t,\bar{A},d_t\right)\cdot \\ &\quad \frac{\partial \mathcal{R}_{t+1}}{\partial A_t}\left(\bar{A}, d_{t-1}, st_t, d_t, st_{t+1}, \xi_{t+1}^{(k)}\right)\cdot \\ &\quad \frac{\partial u}{\partial c}\left(c_{t+1}^{(k)}, d_t, st_{t+1}\right) \end{aligned} \tag{10}$$

7. Also calculate the expected value function at period $t+1$ by interpolating $V_{t+1}\left(M_{t+1}^{grid}(d_t, st_{t+1}), d_t, st_{t+1}\right)$ in points $M_{t+1}^{(k)}$

$$E\left[V_{t+1}\left(M_{t+1}, d_t, st_{t+1}\right)|\bar{A}, st_t\right] = \frac{1}{K}\sum_{st_{t+1}}\sum_{i=1}^{K} P\left(st_{t+1}|st_t,\bar{A},d_t\right)V_{t+1}\left(M_{t+1}^{(k)}, d_t, st_{t+1}\right) \tag{11}$$

8. Inverting the Euler equation and using (10) compute

$$c_t^{d_t}\left(\bullet, d_{t-1}, st_t\right) = \left(u'(\bullet, d_{t-1}, st_t)\right)^{-1}\left(\beta_{t+1}E\left[\frac{\partial \mathcal{R}_{t+1}}{\partial A_t}\cdot\frac{\partial u}{\partial c}\left(c_{t+1}, d_t, st_{t+1}\right)|\bar{A}, st_t\right]\right),$$

and using (3)

$$M_t = \bar{A} + c_t^{d_t}\left(\bullet, d_{t-1}, st_t\right).$$

Acknowledging the correspondence between M_t, $c_t^{d_t}\left(\bullet, d_{t-1}, st_t\right)$ and $E\left[V_{t+1}\left(M_{t+1}, d_t, st_{t+1}\right)|\bar{A}\right]$ for each guess $\bar{A}$ gives the resulting policy function $c_t^{d_t}\left(M_t, d_{t-1}, st_t\right)$, d_t-specific value function

$$V^{d_t}\left(M_t, d_{t-1}, st_t\right) = u\left(c_t^{d_t}\left(M_t, d_{t-1}, st_t\right), d_{t-1}, st_t\right) + \beta_{t+1}E\left[V\left(M_{t+1}, d_t, st_{t+1}\right)|A_t, st_t\right]$$

and the endogenous grid $M_t^{grid}(d_t, d_{t-1}, st_t)$ on which they are defined. This completes the loop over potential values of A_t (started in 5).

9. Complete the loop over d_t (started in 4) to obtain all policy functions $c_t^{d_t}\left(M_t, d_{t-1}, st_t\right)$ and d_t-specific value functions $V^{d_t}\left(M_t, d_{t-1}, st_t\right)$ defined over grids $M_t^{grid}(d_t, d_{t-1}, st_t)$ for all values of d_t.

10. Compute the upper envelope of decision specific value functions $V^{d_t}\left(M_t, d_{t-1}, st_t\right)$ to obtain value function $V_t\left(M_t, st_t|d_{t-1}\right)$ defined over the unified grid $M_t^{grid}\left(d_{t-1}, st_t\right)$, and the optimal decision rule $d_t\left(M_t, d_{t-1}, st_t\right)$. Pseudo-code for this sub-routine is presented in the Appendix B.

11. Compute optimal policy for consumption and optimal decision rule for the continuous control

$$c_t\left(M_t, d_{t-1}, st_t\right) = c_t^{d_t(M_t, d_{t-1}, st_t)}\left(M_t, d_{t-1}, st_t\right).$$

12. Proceed with the next iteration over values of st_t (started in 3).

13. Proceed with the next iteration over values of d_{t-1} (started in 2).

14. When the loops over st_t and d_{t-1} are completed the policy function for continuous control $A_t\left(M_t, d_{t-1}, st_t\right) = M_t - c_t\left(M_t, d_{t-1}, st_t\right)$ for period t and value functions $V\left(M_t, d_{t-1}, st_t\right)$ are calculated for all values of st_t and d_{t-1} and algorithm can proceed to the next iteration in backward induction.

15. The backward induction is carried on till the first period T_0, for which d_{T_0-1} is known and fixed by assumption A5.

[9] This is static version of approximation in [Tauchen, 1986]

B Pseudo-code for upper envelope algorithm

<to be added>

C Pseudo-code for generator of guesses of A_t

For the purpose of this appendix, denote $m(a)$ the function which performs core EGM steps (steps 6 to 8 in pseudo-code in Appendix A), and return the value of money-at-hand $M_t = m(a)$ corresponding to a guess $A_t = a$. First, note that since $A_0 \le A_t \le M_t$, $m(a)$ is defined for $a \ge A_0$ and lies above the forty five degree line. The following pseudo-code is designed to generate n values $a_0, .., a_{n-1}$ such that the interval between $m(A_0) = m(a_0)$ and some arbitrary upper bound $\bar{M} = m(a_n)$ is filled with values $m(a_0), .., m(a_{n-1})$ such that $lim_{n\to\infty} \{max_{j=1,..,n} [m(a_j) - m(a_{j-1})]\} = 0$.

1. Allow $a' = \bar{M}$. Calculate $m(a')$. If $m(a') \le \bar{M}$, set $\widehat{a} = a'$, $\widehat{m} = m(a_0)$ and proceed to next step, otherwise let $a' \leftarrow \frac{1}{2}(a' + A_0)$ and repeat this step.
2. Set $i = 0$ $a_i = A_0$ and calculate $m(a_i)$.
3. Find $\bar{a}_i$ such that the straight line through $(a_i, m(a_i))$ and $(\widehat{a}, \widehat{m})$ takes value $\bar{M}$ at $\bar{a}_i$.
4. Divide the interval $(a_i, \bar{a}_i)$ into $n - i$ equal parts and allow the next draw be $a_{i+1} = a_i + \frac{\bar{a}_i - a_i}{n-i}$.
5. Update $i \leftarrow i + 1$ and return to step 3 unless $i = n$.

Iterations are illustrated in figure 2 for $n = 10$, $A_0 = 0$, $\bar{M} = 100$ and quadratic $m(a)$. The algorithm produces the needed sequence $\{a_1, .., a_n\}$ after 3 iterations over step 1. These initial iterations are needed to ensure that the calculations (and effectively the whole solution of the model) is contained within the bound $\bar{M}$. It is hard to say exactly how many initial iterations are needed — it depends on how fast $m(a)$ grows — but even for fast growing functions like $m(a) = exp(a)$ or $m(a) = exp\,(exp(a))$ the binary section approach takes only several steps (8 and 9 respectively) to find the point within the bound.

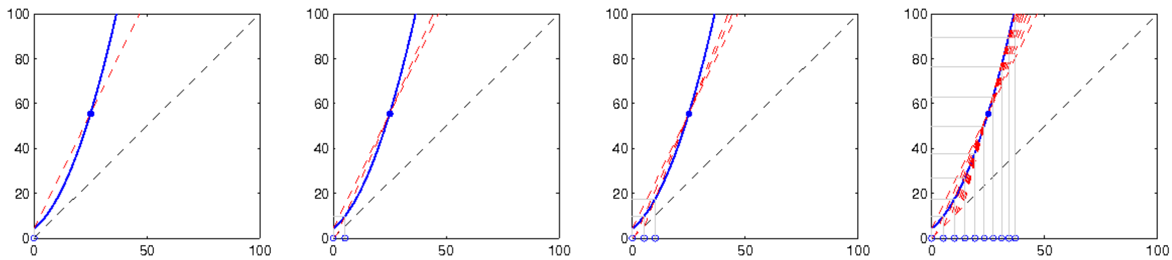

Figure 2: Algorithm for generating guesses of A_t.