
EGMⁿ: THE SEQUENTIAL ENDOGENOUS GRID METHOD

Alan Lujan¹

Johns Hopkins University AAP
Econ-ARK

Sunday 27th October, 2024

Abstract

Heterogeneous agent models with multiple decisions are often solved using inefficient grid search methods that require many evaluations and are slow. This paper provides a novel method for solving such models using an extension of the Endogenous Grid Method (EGM) that uses Gaussian Process Regression (GPR) to interpolate functions on unstructured grids. First, I propose an intuitive and strategic procedure for decomposing a problem into sub-problems which allows the use of efficient solution methods. Second, using an exogenous grid of post-decision states and solving for an endogenous grid of pre-decision states that obey a first-order condition greatly speeds up the solution process. Third, since the resulting endogenous grid can often be non-rectangular at best and unstructured at worst, GPR provides an efficient and accurate method for interpolating the value, marginal value, and decision functions. Applied sequentially to each decision within the problem, the method is able to solve heterogeneous agent models with multiple decisions in a fraction of the time and with less computational resources than are required by standard methods currently used. Software to reproduce these methods is available under the [Econ-ARK/HARK](#) project for the `python` programming language.

Keywords endogenous grid method, dynamic programming, machine learning

1 Introduction

1.1 Background

Macroeconomic modeling aims to describe a complex world of agents interacting with each other and making decisions in a dynamic setting. The models are often very complex, require strong underlying assumptions, and use a lot of computational power to solve. One of the most common methods to solve these complex problems is using a grid search method to solve the model. The Endogenous Grid Method (EGM) developed by [Carroll \[2006\]](#) allows dynamic optimization problems to be solved in a more computationally efficient and faster manner than the previous method of convex optimization using grid search. Many problems that before took hours to compute became much easier to solve and allowed macroeconomists and computational economists to focus on estimation and simulation. However, the Endogenous Grid Method is limited to a few specific classes of problems. Recently, the classes of problems to which EGM can be applied have been expanded¹, but with every new method comes a new set of limitations. This paper introduces a new approach to EGM in a multivariate setting. The method is called Sequential EGM (or EGMⁿ) and introduces a novel way of breaking down complex problems into a sequence of simpler, smaller, and more tractable problems, along with an exploration of new multidimensional interpolation methods that can be used to solve these problems.

¹Correspondence to: alujan@jhu.edu

¹[Barillas and Fernández-Villaverde \[2007\]](#), [Maliar and Maliar \[2013\]](#), [Fella \[2014\]](#), [White \[2015\]](#), [Iskhakov et al. \[2017\]](#), among others.

1.2 Literature Review

Carroll [2006] first introduced the Endogenous Grid Method as a way to speed up the solution of dynamic stochastic consumption-savings problems. The method consists of starting with an exogenous grid of post-decision states and using the inverse of the first-order condition to find the optimal consumption policy that rationalizes such post-decision states. Given the optimal policy and post-decision states, it is straightforward to calculate the initial pre-decision state that leads to the optimal policy. Although this method is certainly innovative, it only applied to a model with one control variable and one state variable. Barillas and Fernández-Villaverde [2007] further extend this method by including more than one control variable in the form of a labor-leisure choice, as well as a second state variable for stochastic persistence.

Hintermaier and Koeniger [2010] introduce a model with collateral constraints and non-separable utility and solve using an EGM method that allows for occasionally binding constraints among endogenous variables. Jørgensen [2013] evaluates the performance of the Endogenous Grid Method against other methods for solving dynamic stochastic optimization problems and finds it to be fast and efficient. Maliar and Maliar [2013] develop the Envelope Condition Method based on a similar idea as the Endogenous Grid Method, avoiding the need for costly numerical optimization and grid search. However, their model is limited to infinite horizon problems as it is a forward solution method.

Further development into a multivariate Endogenous Grid Method expanded the ability of researchers to solve models efficiently. White [2015] formally characterized the conditions for the Endogenous Grid Method and developed an interpolation method for structured non-rectilinear, or curvilinear, grids. Iskhakov [2015] additionally establishes conditions for solving multivariate models with EGM, requiring the invertibility of a triangular system of first-order conditions. Ludwig and Schön [2018] also develops a novel interpolating method using Delaunay triangulation of the resulting unstructured endogenous grid. However, the authors show that the gains from avoiding the grid search method are often offset by the costly construction of the triangulation.

For the papers discussed above, continuity and smoothness of the value and first-order conditions are strict requirements. Fella [2014] first introduced a method to solve non-convex problems using the Endogenous Grid Method. The idea is based on evaluating necessary but not sufficient candidates for the first-order condition in overlapping regions of the state space. Arellano et al. [2016] use the Envelope Condition Method to solve a sovereign default risk model with similar efficiency gains to EGM. Iskhakov et al. [2017] further advances the methodology by using extreme errors to solve discrete choice problems with Endogenous Grid Method. These methods however were only applied to a single control variable and a single state variable. Druedahl and Jørgensen [2017] introduces the *G2EGM* to handle non-convex problems with more than 1 control variable and more than 1 state variable. This method is also capable of handling occasionally binding constraints which previous multivariate EGM methods were not.

Clausen and Strub [2020] formalize the applicability of the Endogenous Grid Method and its extensions to discrete choice models and discuss the nesting of problems to efficiently find accurate solutions. Druedahl [2021] similarly suggest the nesting of problems to efficiently use the Endogenous Grid Method within problems with multiple control variables. However, while these nested methods reduce the complexity of solving these models, they often still require grid search methods as is the case with Druedahl [2021].

1.3 Research Question

The purpose of this paper is to describe a new method for solving dynamic optimization problems efficiently and accurately while avoiding convex optimization and grid search methods with the use of the Endogenous Grid Method and first-order conditions. The method is called Sequential EGM (or EGMⁿ) and introduces a novel way of breaking down complex problems into a sequence of simpler, smaller, and more tractable problems, along with an exploration of new multidimensional interpolation methods that can be used to solve these problems. This paper also illustrates an example of how Sequential EGM can be used to solve a dynamic optimization problem in a multivariate setting.

1.4 Methodology

The sequential Endogenous Grid Method consists of 3 major parts: First, the problem to be solved should be broken up into a sequence of smaller problems that themselves don't add any additional state variables or introduce asynchronous dynamics with respect to the uncertainty. If the problem is broken up in such a way that uncertainty can happen in more than one period, then the solution to this sequence of problems might

be different from the aggregate problem due to giving the agent additional information about the future by realizing some uncertainty. Second, I evaluate each of the smaller problems to see if they can be solved using the Endogenous Grid Method. This evaluation is of greater scope than the traditional Endogenous Grid Method, as it allows for the resulting exogenous grid to be non-regular. If the subproblem can not be solved with **EGM**, then convex optimization is used. Third, if the exogenous grid generated by the **EGM** is non-regular, then I use a multidimensional interpolation method that takes advantage of machine learning tools to generate an interpolating function. Solving each subproblem in this way, the sequential Endogenous Grid Method is capable of solving complex problems that are not solvable with the traditional Endogenous Grid Method and are difficult and time-consuming to solve with convex optimization and grid search methods.

1.5 Contributions

The Sequential Endogenous Grid Method is capable of solving multivariate dynamic optimization problems in an efficient and fast manner by avoiding grid search. This should allow researchers and practitioners to solve more complex problems that were previously not easily accessible to them, but more accurately capture the dynamics of the macroeconomy. By using advancements in machine learning techniques such as Gaussian Process Regression, the Sequential Endogenous Grid Method is capable of solving problems that were not previously able to be solved using the traditional Endogenous Grid Method. In particular, the Sequential Endogenous Grid Method is different from **NEGM** in that it allows for using more than one Endogenous Grid Method step to solve a problem, avoiding costly grid search methods to the extent that the problem allows.

Additionally, the Sequential Endogenous Grid Method often sheds light on the problem by breaking it down into a sequence of simpler problems that were not previously apparent. This is because intermediary steps in the solution process generate value and marginal value functions of different pre- and post-decision states that can be used to understand the problem better.

1.6 Outline

Section 2 presents a basic model that illustrates the sequential Endogenous Grid Method in one dimension. Then Section 3 introduces a more complex method with two state variables to demonstrate the use of machine learning tools to generate an interpolating function. In Section 4 I present the unstructured interpolation methods using machine learning in more detail. Section 5 discusses the theoretical requirements to use the Sequential Endogenous Grid Method. Finally, Section 6 concludes with some limitations and future work.

2 The Sequential Endogenous Grid Method

2.1 A basic model

The baseline problem which I will use to demonstrate the Sequential Endogenous Grid Method (**EGM**ⁿ) is a discrete time version of Bodie et al. [1992] where a consumer has the ability to adjust their labor as well as their consumption in response to financial risk. The objective consists of maximizing the present discounted lifetime utility of consumption and leisure.

$$V_0(B_0, \theta_0) = \max \mathbb{E}_t \left[\sum_{n=0}^{T-t} \beta^n u(C_{t+n}, Z_{t+n}) \right] \quad (1)$$

In particular, this example makes use of a utility function that is based on Example 1 in the paper, which is that of additively separable utility of labor and leisure as

$$u(C, Z) = u(C) + h(Z) = \frac{C^{1-\rho}}{1-\rho} + \nu^{1-\rho} \frac{Z^{1-\zeta}}{1-\zeta} \quad (2)$$

where the term $\nu^{1-\rho}$ is introduced to allow for a balanced growth path as in Mertens and Ravn [2011]. The use of additively separable utility is ad-hoc, as it will allow for the use of multiple **EGM** steps in the solution process, as we'll see later.

This model represents a consumer who begins the period with a level of bank balances b_t and a given wage offer θ_t . Simultaneously, they are able to choose consumption, labor intensity, and a risky portfolio share with the objective of maximizing their utility of consumption and leisure, as well as their future wealth.

The problem can be written in normalized recursive form² as

$$\begin{aligned}
v_t(b_t, \theta_t) &= \max_{\{c_t, z_t, \varsigma_t\}} u(c_t, z_t) + \beta \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(b_{t+1}, \theta_{t+1}) \right] \\
&\text{s.t.} \\
\ell_t &= 1 - z_t \\
m_t &= b_t + \theta_t \ell_t \\
a_t &= m_t - c_t \\
\mathbb{R}_{t+1} &= R + (\mathbf{R}_{t+1} - R) \varsigma_t \\
b_{t+1} &= a_t \mathbb{R}_{t+1} / \Gamma_{t+1}
\end{aligned} \tag{4}$$

in which ℓ_t is the time supplied to labor net of leisure, m_t is the market resources totaling bank balances and labor income, a_t is the amount of saving assets held by the consumer, and ς_t is the risky share of assets, which induce a \mathbb{R}_{t+1} return on portfolio that results in next period's bank balances b_{t+1} normalized by next period's permanent income Γ_{t+1} .

2.2 Restating the problem sequentially

We can make a few choices to create a sequential problem which will allow us to use multiple **EGM** steps in succession. First, the agent decides their labor-leisure trade-off and receives a wage. Their wage plus their previous bank balance then becomes their market resources. Second, given market resources, the agent makes a pure consumption-saving decision. Finally, given an amount of savings, the consumer then decides their risky portfolio share.

Starting from the beginning of the period, we can define the labor-leisure problem as

$$\begin{aligned}
v_t(b_t, \theta_t) &= \max_{z_t} h(z_t) + \tilde{\mathbf{v}}_t(m_t) \\
&\text{s.t.} \\
0 &\leq z_t \leq 1 \\
\ell_t &= 1 - z_t \\
m_t &= b_t + \theta_t \ell_t.
\end{aligned} \tag{5}$$

The pure consumption-saving problem is then

$$\begin{aligned}
\tilde{\mathbf{v}}_t(m_t) &= \max_{c_t} u(c_t) + \beta \mathbf{v}_t(a_t) \\
&\text{s.t.} \\
0 &\leq c_t \leq m_t \\
a_t &= m_t - c_t.
\end{aligned} \tag{6}$$

Finally, the risky portfolio problem is

²As in [Carroll \[2009\]](#), where the utility of normalized consumption and leisure is defined as

$$u(c_t, z_t) = P_t^{1-\rho} \frac{c_t^{1-\rho}}{1-\rho} + (\nu P_t)^{1-\rho} \frac{z_t^{1-\zeta}}{1-\zeta} \tag{3}$$

$$\begin{aligned}
v_t(a_t) &= \max_{\varsigma_t} \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(b_{t+1}, \theta_{t+1}) \right] \\
&\text{s.t.} \\
0 &\leq \varsigma_t \leq 1 \\
\mathbb{R}_{t+1} &= \mathbb{R} + (\mathbf{R}_{t+1} - \mathbb{R})\varsigma_t \\
b_{t+1} &= a_t \mathbb{R}_{t+1} / \Gamma_{t+1}.
\end{aligned} \tag{7}$$

This sequential approach is explicitly modeled after the nested approaches explored in Clausen and Strub [2020] and Druedahl [2021]. However, I will offer additional insights that expand on these methods. An important observation is that now, every single choice is self-contained in a subproblem, and although the structure is specifically chosen to minimize the number of state variables at every stage, the problem does not change by this structural imposition. This is because there is no additional information or realization of uncertainty that happens between decisions, as can be seen by the expectation operator being in the last subproblem. From the perspective of the consumer, these decisions are essentially simultaneous, but a careful organization into sub-period problems enables us to solve the model more efficiently and can provide key economic insights. In this problem, as we will see, a key insight will be the ability to explicitly calculate the marginal value of wealth and the Frisch elasticity of labor.

2.3 The portfolio decision subproblem

As useful as it is to be able to use the EGM step more than once, there are clear problems where the EGM step is not applicable. This basic labor-portfolio choice problem demonstrates where we can use an additional EGM step, and where we can not. First, we go over a subproblem where we can not use the EGM step.

In reorganizing the labor-portfolio problem into subproblems, we assigned the utility of leisure to the leisure-labor subproblem and the utility of consumption to the consumption-savings subproblem. There are no more separable convex utility functions to assign to this problem, and even if we re-organized the problem in a way that moved one of the utility functions into this subproblem, they would not be useful in solving this subproblem via EGM as there is no direct relation between the risky share of portfolio and consumption or leisure. Therefore, the only way to solve this subproblem is through standard convex optimization and root-finding techniques.

Restating the problem in compact form gives

$$v_t(a_t) = \max_{\varsigma_t} \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(a_t(\mathbb{R} + (\mathbf{R}_{t+1} - \mathbb{R})\varsigma_t), \theta_{t+1}) \right]. \tag{8}$$

The first-order condition with respect to the risky portfolio share is then

$$\mathbb{E}_t \left[\Gamma_{t+1}^{-\rho} v_{t+1}^b(b_{t+1}, \theta_{t+1}) (\mathbf{R}_{t+1} - \mathbb{R}) \right] = 0. \tag{9}$$

Finding the optimal risky share requires numerical optimization and root-solving of the first-order condition. To close out the problem, we can calculate the envelope condition as

$$v'_t(a_t) = \mathbb{E}_t \left[\Gamma_{t+1}^{-\rho} v_{t+1}^b(b_{t+1}, \theta_{t+1}) \mathbb{R}_{t+1} \right]. \tag{10}$$

2.3.1 A note on avoiding taking expectations more than once

We could instead define the portfolio choice subproblem as:

$$v_t(a_t) = \max_{\varsigma_t} \check{v}(a_t, \varsigma_t) \tag{11}$$

where

$$\begin{aligned}
\check{\mathbf{v}}_t(a_t, \varsigma_t) &= \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(b_{t+1}, \theta_{t+1}) \right] \\
\mathbb{R}_{t+1} &= \mathbf{R} + (\mathbf{R}_{t+1} - \mathbf{R})\varsigma_t \\
b_{t+1} &= a_t \mathbb{R}_{t+1} / \Gamma_{t+1}
\end{aligned} \tag{12}$$

In this case, the process is similar. The only difference is that we don't have to take expectations more than once. Given the next period's solution, we can calculate the marginal value functions as:

$$\begin{aligned}
\check{\mathbf{v}}_t^a(a_t, \varsigma_t) &= \mathbb{E}_t \left[\Gamma_{t+1}^{-\rho} v'_{t+1}(b_{t+1}, \theta_{t+1}) \mathbb{R}_{t+1} \right] \\
\check{\mathbf{v}}_t^\varsigma(a_t, \varsigma_t) &= \mathbb{E}_t \left[\Gamma_{t+1}^{-\rho} v'_{t+1}(b_{t+1}, \theta_{t+1}) a_t (\mathbf{R}_{t+1} - \mathbf{R}) \right]
\end{aligned} \tag{13}$$

If we are clever, we can calculate both of these in one step. Now, the optimal risky share can be found by the first-order condition and we can use it to evaluate the envelope condition.

$$\text{F.O.C.:} \quad \check{\mathbf{v}}_t^\varsigma(a_t, \varsigma_t^*) = 0 \quad \text{E.C.:} \quad \mathbf{v}_t^a(a_t) = \check{\mathbf{v}}_t^a(a_t, \varsigma_t^*) \tag{14}$$

2.4 The consumption-saving subproblem

The consumption-saving [EGM](#) follows [Carroll \[2006\]](#) but I will cover it for exposition. We can begin the solution process by restating the consumption-savings subproblem in a more compact form, substituting the market resources constraint and ignoring the no-borrowing constraint for now. The problem is:

$$\tilde{\mathbf{v}}_t(m_t) = \max_{c_t} u(c_t) + \beta \mathbf{v}_t(m_t - c_t) \tag{15}$$

To solve, we derive the first-order condition with respect to c_t which gives the familiar Euler equation:

$$u'(c_t) = \beta \mathbf{v}'_t(m_t - c_t) = \beta \mathbf{v}'_t(a_t) \tag{16}$$

Inverting the above equation is the (first) [EGM](#) step.

$$\mathbf{c}_t(a_t) = u'^{-1}(\beta \mathbf{v}'_t(a_t)) \tag{17}$$

Given the utility function above, the marginal utility of consumption and its inverse are

$$u'(c) = c^{-\rho} \quad u'^{-1}(x) = x^{-1/\rho}. \tag{18}$$

[Carroll \[2006\]](#) demonstrates that by using an exogenous grid of $[a]$ points we can find the unique $\mathbf{c}_t([a])$ that optimizes the consumption-saving problem, since the first-order condition is necessary and sufficient. Further, using the market resources constraint, we can recover the exact amount of market resources that is consistent with this consumption-saving decision as

$$\mathbf{m}_t([a]) = \mathbf{c}_t([a]) + [a]. \tag{19}$$

This $\mathbf{m}_t([a])$ is the “endogenous” grid that is consistent with the exogenous decision grid $[a]$. Now that we have a $(\mathbf{m}_t([a]), \mathbf{c}_t([a]))$ pair for each $a \in [a]$, we can construct an interpolating consumption function for market resources points that are off-the-grid.

The envelope condition will be useful in the next section, but for completeness is defined here.

$$\tilde{\mathbf{v}}'_t(m_t) = \beta \mathbf{v}'_t(a_t) = u'(c_t) \tag{20}$$

2.5 The labor-leisure subproblem

The labor-leisure subproblem can be restated more compactly as:

$$v_t(b_t, \theta_t) = \max_{z_t} h(z_t) + \tilde{v}_t(b_t + \theta_t(1 - z_t)) \quad (21)$$

The first-order condition with respect to leisure implies the labor-leisure Euler equation

$$h'(z_t) = \tilde{v}'_t(m_t)\theta_t \quad (22)$$

The marginal utility of leisure and its inverse are

$$h'(z) = \nu z^{-\zeta} \quad h'^{-1}(x) = (x/\nu)^{-1/\zeta} \quad (23)$$

Using an exogenous grid of $[m]$ and $[\theta]$, we can find leisure as

$$\mathfrak{z}_t([m], [\theta]) = h'^{-1}(\tilde{v}'_t([m])[\theta]) \quad (24)$$

In this case, it's important to note that there are conditions for leisure itself. An agent with a small level of market resources m_t might want to work more than their available time endowment, especially at higher levels of income θ_t , if the utility of leisure is not enough to compensate for their low wealth. In these situations, the optimal unconstrained leisure might be negative, so we must impose a constraint on the optimal leisure function. This is similar to the treatment of an artificial borrowing constraint in the pure consumption subproblem. From now on, let's call this constrained optimal function $\hat{\mathfrak{z}}_t([m], [\theta])$, where

$$\hat{\mathfrak{z}}_t([m], [\theta]) = \max[\min[\mathfrak{z}_t([m], [\theta]), 1], 0] \quad (25)$$

Then, we derive labor as $l_t(m_t, \theta_t) = 1 - \hat{\mathfrak{z}}_t(m_t, \theta_t)$. Finally, for each θ_t and m_t as an exogenous grid, we can find the endogenous grid of bank balances as $\mathfrak{b}_t(m_t, \theta_t) = m_t - \theta_t l_t(m_t, \theta_t)$.

The envelope condition then provides a heterogeneous Frisch elasticity of labor as simply

$$v_t^b(b_t, \theta_t) = \tilde{v}'_t(m_t) = h'(z_t)/\theta_t. \quad (26)$$

2.6 Alternative Parametrization

An alternative formulation for the utility of leisure is to state it in terms of the disutility of labor as in (references)

$$h(\ell) = -\zeta \frac{\ell^{1+\nu}}{1+\nu} \quad (27)$$

In this case, we can restate the problem as

$$h(z) = -\zeta \frac{(1-z)^{1+\nu}}{1+\nu} \quad (28)$$

The marginal utility of leisure and its inverse are

$$h'(z) = \zeta(1-z)^\nu \quad h'^{-1}(x) = 1 - (x/\zeta)^{1/\nu} \quad (29)$$

2.7 Curvilinear Grids

Although **EGM**ⁿ seems to be a simple approach, there is one important caveat that we have not discussed, which is the details of the interpolation. In the pure consumption-savings problem, a one-dimensional exogenous grid of post-decision liquid assets [a] results in a one-dimensional endogenous grid of total market resources [m]. However, as we know from standard **EGM**, the spacing in the [m] grid is different from the spacing in the [a] grid as the inverted Euler equation is non-linear. This is no problem in a one-dimensional problem as we can simply use non-uniform linear interpolation.

However, the same is true of higher dimensional problems, where the exogenous grid gets mapped to a warped endogenous grid. In this case, it is not possible to use standard multi-linear interpolation, as the resulting endogenous grid is not rectilinear. Instead, I introduce a novel approach to interpolation that I call Warped Grid Interpolation (**WGI**), which is similar to White [2015]’s approach but computationally more efficient and robust. The details of this interpolation method will be further explained in Section 4, but for now, we show the resulting warped endogenous grid for the labor-leisure problem.

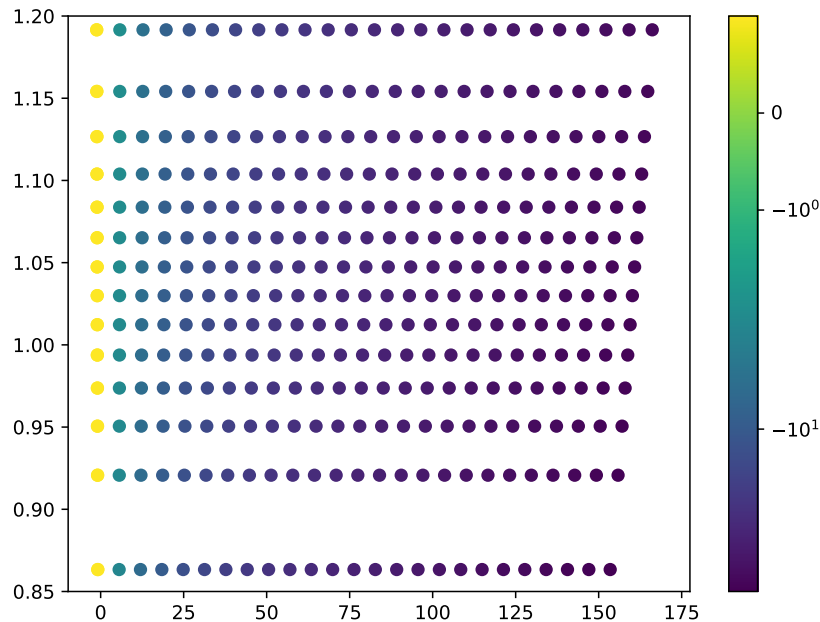


Figure 1: Warped Curvilinear Grid that results from multivariate **EGM**. This grid can be interpolated by **WGI**.

2.8 Warped Grid Interpolation (**WGI**)

Assume we have a set of points indexed by (i, j) in two-dimensional space for which we have corresponding functional values in a third dimension, such that $f(x_{ij}, y_{ij}) = z_{ij}$. In practice, we are interested in cases where the z_{ij} are difficult to compute and $f(x_{ij}, y_{ij})$ is unknown, so we are unable to compute them at other values of x and y — which is why we want to interpolate³. These (x_{ij}, y_{ij}) points however are not evenly spaced and do not form a rectilinear grid which would make it easy to interpolate the function off the grid. Nevertheless, these points do have a regular structure as we will see.

In Figure 2, we can see the true function in three-dimensional space, along with the points for which we actually know the value of the function. The underlying regular structure comes from the points’ position in the matrix, the (i, j) coordinates. If we join the points along every row and every column, we can see that the resulting grid is regular and piecewise affine (curvilinear).

³For this illustration, we generate z ’s arbitrarily using the function

$$f(x, y) = (xy)^{1/4}. \quad (30)$$

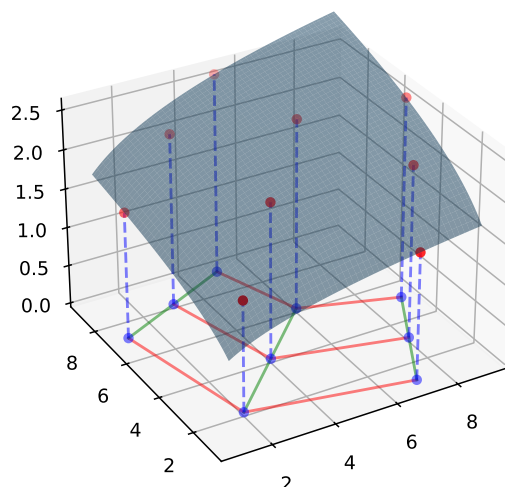


Figure 2: True function and curvilinear grid of points for which we know the value of the function.

In Figure 3 we see the values of the function at their index coordinate points in the matrix. We can see that there exists a mapping between the curvilinear grid and the index coordinates of the matrix.

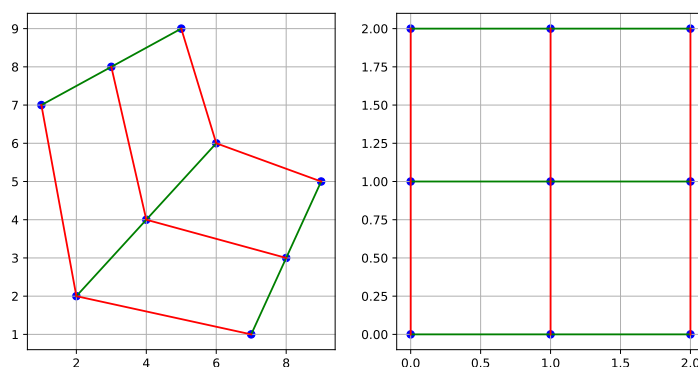


Figure 3: Homotopy between the curvilinear grid and the index coordinates of the matrix.

The objective is to be able to interpolate the value of the function at any point off the grid, where presumably we are only interested in points internal to the curvilinear space and not outside the boundaries. For example, we can imagine that we want an approximation to the function at the point $(x, y) = (3, 5)$ pictured Figure 4. If we could find the corresponding point in the coordinate grid, interpolation would be straightforward. We can find where the x -coordinate of the point of interest intersects with the index-coordinates of the matrix. This is similar to assuming that we have 3 linear interpolators formed by connecting the points on the green lines in the x -direction, and for each interpolator we can approximate the corresponding y and z values using the grid data. Now, for each circle in Figure 4, we have a corresponding pair (y, z) , and we can interpolate in the y -direction to find the corresponding z -value for the point's y -coordinate⁴.

⁴For more examples of the Warped Grid Interpolation method in action, see the github project [alanlujan91/multinterp](https://github.com/alanlujan91/multinterp).

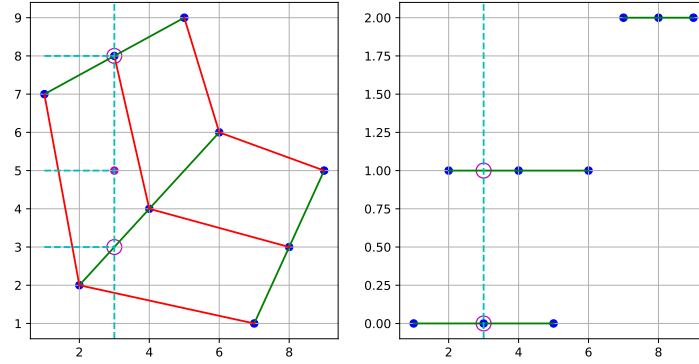


Figure 4: The method consist of extending the loci of points in the x dimension to find the corresponding crossing points in the y dimension.

3 The EGM^n in Higher Dimensions

The problem in Section 2 demonstrates the simplicity of solving problems sequentially. However, as constructed, the problem has only one state variable and one post-decision state variable per stage. Can EGM^n be used to solve higher dimensional problems? In short, yes, but it requires additional thought on interpolation.

3.1 A more complex problem

For a demonstration, we now turn to the problem of a worker saving up for retirement. This worker must consume, save, and deposit resources into a tax-advantaged account that can not be liquidated until retirement. In the recursive problem, the worker begins a new period with a liquid account of market resources m_t and an illiquid account of retirement savings n_t . The worker maximizes their utility by choosing consumption c_t and pension deposit d_t . The pension deposit is set aside on a retirement account that is exposed to a risky return, while their post-consumption liquid assets accrue risk-free interest every period. The worker additionally receives an income that faces a permanent (Γ_{t+1}) and a transitory (θ_{t+1}) shock every period. At the age of 65, the worker is retired and their assets are liquidated, at which point the state reduces to one liquid account of market resources. The worker's recursive problem is:

$$\begin{aligned}
 v_t(m_t, n_t) &= \max_{c_t, d_t} u(c_t) + \beta \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(m_{t+1}, n_{t+1}) \right] \\
 \text{s.t. } & c_t \geq 0, \quad d_t \geq 0 \\
 a_t &= m_t - c_t - d_t \\
 b_t &= n_t + d_t + g(d_t) \\
 m_{t+1} &= a_t \mathbf{R} / \Gamma_{t+1} + \theta_{t+1} \\
 n_{t+1} &= b_t \mathbf{R}_{t+1} / \Gamma_{t+1}
 \end{aligned} \tag{31}$$

where

$$g(d) = \chi \log(1 + d). \tag{32}$$

This problem can subsequently be broken down into 3 stages: a pension deposit stage, a consumption stage, and an income shock stage.

3.2 Breaking down the problem

In the deposit stage, the worker begins with market resources and a retirement savings account. The worker must maximize their value of liquid wealth l_t and retirement balance b_t by choosing a pension deposit d_t ,

which must be positive. The retirement balance b is the cash value of their retirement account plus their pension deposit and an additional amount $g(d_t)$ that provides an incentive to save for retirement. As we'll see, this additional term will allow us to use the Endogenous Grid Method to solve this subproblem.

$$\begin{aligned} v_t(m_t, n_t) &= \max_{d_t} \tilde{v}_t(l_t, b_t) \\ \text{s.t. } d_t &\geq 0 \\ l_t &= m_t - d_t \\ b_t &= n_t + d_t + g(d_t) \end{aligned} \quad (33)$$

After making their pension decision, the worker begins their consumption stage with liquid wealth l_t and retirement balance b_t . From their liquid wealth, the worker must choose a level of consumption to maximize utility and continuation value w_t . After consumption, the worker is left with post-decision states that represent liquid assets a_t and retirement balance b_t , which passes through this problem unaffected because it can't be liquidated until retirement.

$$\begin{aligned} \tilde{v}_t(l_t, b_t) &= \max_{c_t} u(c_t) + \beta w_t(a_t, b_t) \\ \text{s.t. } c_t &\geq 0 \\ a_t &= l_t - c_t \end{aligned} \quad (34)$$

Finally, the post-decision value function w_t represents the value of both liquid and illiquid account balances before the realization of uncertainty regarding the risky return and income shocks. Since we are dealing with a normalized problem, this stage handles the normalization of state variables and value functions into the next period.

$$\begin{aligned} w_t(a_t, b_t) &= \mathbb{E}_t \left[\Gamma_{t+1}^{1-\rho} v_{t+1}(m_{t+1}, m_{t+1}) \right] \\ \text{s.t. } a_t &\geq 0, \quad b_t \geq 0 \\ m_{t+1} &= a_t R / \Gamma_{t+1} + \theta_{t+1} \\ n_{t+1} &= b_t R_{t+1} / \Gamma_{t+1} \end{aligned} \quad (35)$$

The advantage of conceptualizing this subproblem as a separate stage is that we can construct a function w_t and use it in the prior optimization problems without having to worry about stochastic optimization and taking expectations repeatedly.

3.3 The consumption-saving problem

As seen in the consumption stage above, the retirement balance b_t passes through the problem unaffected because it can't be liquidated until retirement. In this sense, it is already a post-decision state variable. To solve this problem, we can use a fixed grid of b_t and for each obtain endogenous consumption and ex-ante market resources using the simple Endogenous Grid Method for the consumption problem.

3.4 The pension deposit problem

In the deposit stage, both the state variables and post-decision variables are different since both are affected by the pension deposit decision.

First, we can rewrite the pension deposit problem more compactly:

$$v_t(m_t, n_t) = \max_{d_t} \tilde{v}_t(m_t - d_t, n_t + d_t + g(d_t)) \quad (36)$$

The first-order condition is

$$\tilde{v}_t^l(l_t, b_t)(-1) + \tilde{v}_t^b(l_t, b_t)(1 + g'(d_t)) = 0. \quad (37)$$

Rearranging this equation gives

$$g'(d_t) = \frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \quad (38)$$

where

$$g'(d) = \frac{\chi}{1+d} \quad g'^{-1}(y) = \chi/y - 1 \quad (39)$$

Given that $g'(d)$ exists and is invertible, we can find

$$\mathfrak{d}_t(l_t, b_t) = g'^{-1} \left(\frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \right) \quad (40)$$

Using this, we can back out n_t as

$$\mathfrak{n}_t(l_t, b_t) = b_t - \mathfrak{d}_t(l_t, b_t) - g(\mathfrak{d}_t(l_t, b_t)) \quad (41)$$

and m_t as

$$\mathfrak{m}_t(l_t, b_t) = l_t + \mathfrak{d}_t(l_t, b_t) \quad (42)$$

In sum, given an exogenous grid (l_t, b_t) we obtain the triple $(\mathfrak{m}_t(l_t, b_t), \mathfrak{n}_t(l_t, b_t), \mathfrak{d}_t(l_t, b_t))$, which we can use to create an interpolator for the decision rule d_t .

To close the solution method, the envelope conditions are

$$\begin{aligned} v_t^m(m_t, n_t) &= \tilde{v}_t^l(l_t, b_t) \\ v_t^n(m_t, n_t) &= \tilde{v}_t^b(l_t, b_t) \end{aligned} \quad (43)$$

3.5 Unstructured Grid Interpolation

As in Section 2, the resulting endogenous grid is not rectilinear, and in this more complex problem it is not even a regular grid. We can see in Figure 5 that starting from a regular and rectilinear exogenous grid of liquid assets post-consumption l_t and pension balances post-deposit b_t , we obtain Figure 6 which shows an irregular and unstructured endogenous grid of market resources m_t and pension balances pre-deposit n_t .

To interpolate a function defined on an unstructured grid, we use Gaussian Process Regression as in Scheidegger and Bilonis [2019].

4 Multivariate Interpolation on Unstructured Grids

This section presents alternative interpolation methods for non-rectilinear grids. First, I present the relatively simple case of fast warped interpolation on a curvilinear grid, which improves upon the interpolation in White [2015]. Then, I present a machine learning approach to interpolation on unstructured grids based on Gaussian Process Regression as presented in Scheidegger and Bilonis [2019].

4.1 Unstructured Grids

Unstructured interpolation arises in many dynamic programming applications when using the Endogenous Grid Method because the first-order conditions might be highly non-linear and non-monotonic, or because boundary constraints induce kinks in the policy and value functions. In these cases, the grid points generated by the EGM step are not evenly spaced, leading to the need for curvilinear interpolation. We saw in the previous subsection an approach to curvilinear interpolation based on White [2015] that is incapable of interpolation on structured grids. A similar approach was presented in Ludwig and Schön [2018] which used

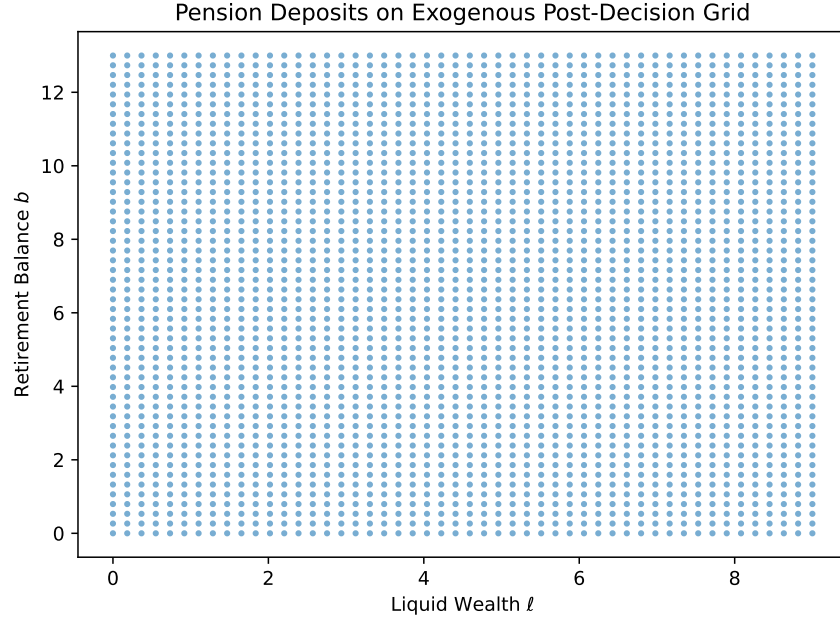


Figure 5: A regular, rectilinear exogenous grid of pension balances after deposit b_t and liquid assets after consumption l_t .

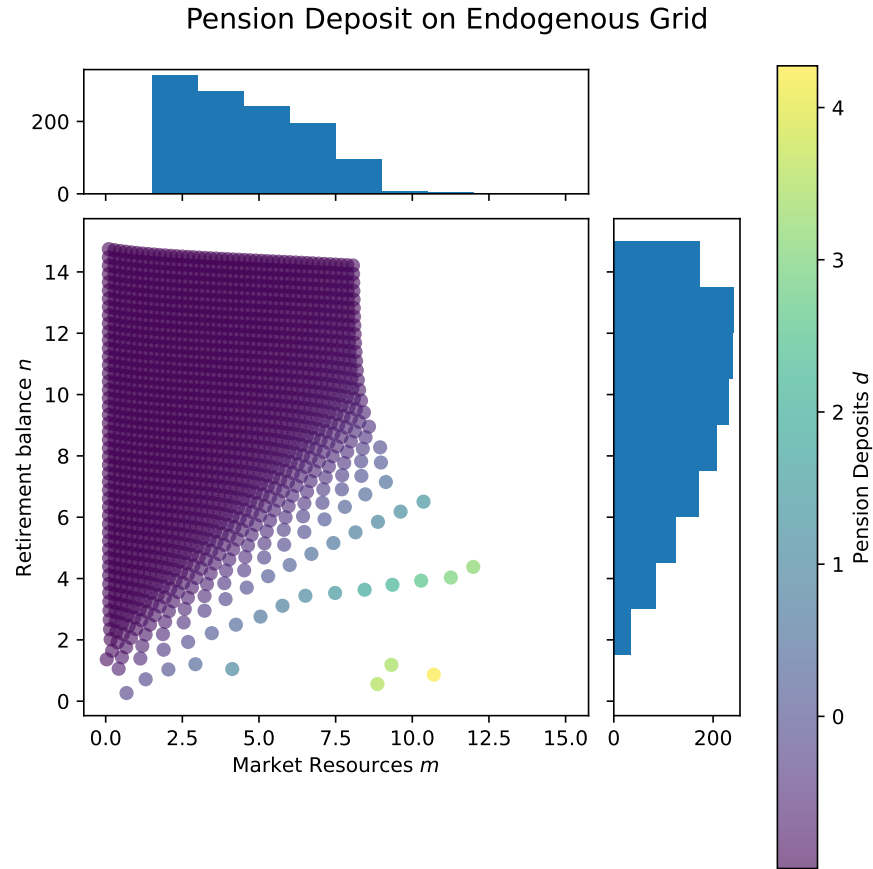


Figure 6: An irregular, unstructured endogenous grid of market resources m_t and pension balances before deposit n_t .

Delaunay interpolation. However, this approach is not well suited for our purposes because triangulation can be computationally intensive and slow, often offsetting the efficiency gains from the Endogenous Grid Method.

As an alternative to these methods, I introduce the use of Gaussian Process Regression (**GPR**) along with the Endogenous Grid Method. **GPR** is computationally efficient, and tools exist to easily parallelize and take advantage of hardware such as Graphics Processing Units (**GPU**)⁵.

4.1.1 Gaussian Process Regression

A Gaussian Process is an infinite dimensional random process for which every subset of random variables is jointly Gaussian or has a multivariate normal distribution.

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad \text{s.t.} \quad x_i \sim \mathcal{N}(\mu_i, \sigma_{ii}) \\ \text{and} \quad \sigma_{ij} &= \mathbb{E}[(x_i - \mu_i)(x_j - \mu_j)] \quad \forall i, j \in \{1, \dots, n\}. \end{aligned} \quad (44)$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \quad \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}. \quad (45)$$

Being infinitely dimensional, a Gaussian Process can be used to represent a probability distribution over the space of functions in n dimensions. Thus, a Gaussian Process Regression is used to find the best fit function to a set of data points.

$$\mathbb{P}(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) \quad (46)$$

where \mathbf{f} is the vector of function values at the points \mathbf{X} , \mathbf{m} is the mean of the function, and \mathbf{K} is a kernel function that describes the covariance between the function values at different points.

A standard kernel function is the squared exponential kernel, or the radial basis function kernel, which is defined as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left(-\frac{1}{2l^2} (\mathbf{x}_i - \mathbf{x}_j)' (\mathbf{x}_i - \mathbf{x}_j) \right). \quad (47)$$

Using **GPR** to interpolate a function f , we can both predict the value of the function at a point \mathbf{x}_* and the uncertainty in the prediction, which provides useful information as to the accuracy of the approximation.

4.1.2 An example of the **GPR**

In Figure 7, we see the function we are trying to approximate along with a sample of data points for which we know the value of the function. In practice, the value of the function is unknown and/or expensive to compute, so we must use a limited amount of data to approximate it.

As we discussed, a Gaussian Process is an infinite dimensional random process which can be used to represent a probability of distributions over the space of functions. In Figure 8, we see a random sample of functions from the **GPR** posterior, which is a Gaussian Process conditioned on fitting the data. From this small sample of functions, we can see that the **GP** generates functions that fit the data well, and the goal of **GPR** is to find the one function that best fits the data given some hyperparameters by minimizing the negative log-likelihood of the data.

In Figure 9, we see the result of **GPR** with a particular parametrization⁶ of the kernel function. The dotted line shows the true function, while the blue dots show the known data points. **GPR** provides the mean function which best fits the data, represented in the figure as an orange line. The shaded region represents

⁵Gardner et al. [2018]

⁶For details see notebook.

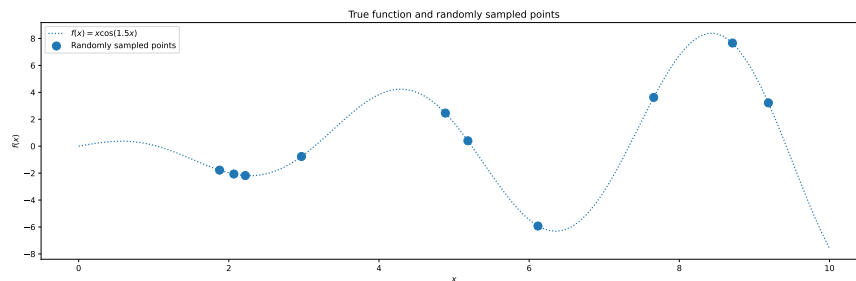


Figure 7: The true function that we are trying to approximate and a sample of data points.

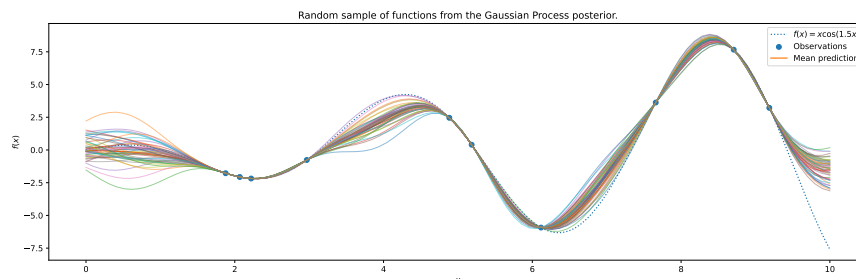


Figure 8: A random sample of functions from the **GPR** posterior that fit the data. The goal of **GPR** is to find the function that best fits the data.

a 95% confidence interval, which is the uncertainty of the predicted function. Along with finding the best fit of the function, **GPR** provides the uncertainty of the prediction, which is useful information as to the accuracy of the approximation.

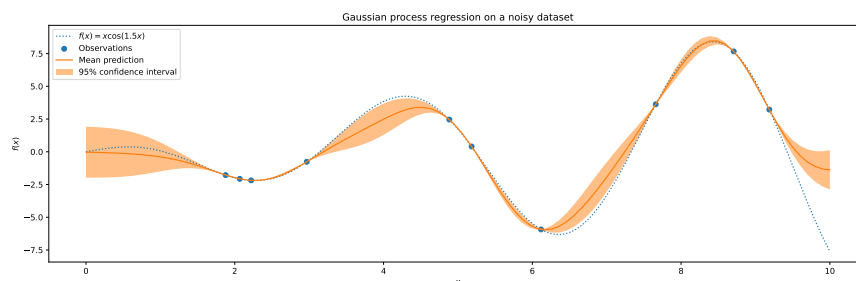


Figure 9: **GPR** finds the function that best fits the data given some hyperparameters. **GPR** then optimizes over the parameter space to find the function that minimizes the negative log-likelihood of the data.

5 Conditions for using the Sequential Endogenous Grid Method

5.1 Splitting the problem into subproblems

The first step in using the Sequential Endogenous Grid Method is to split the problem into subproblems. This process of splitting up the problem has to be strategic to not insert additional complexity into the original problem. If one is not careful when doing this, the subproblems can become more complex and intractable than the original problem.

To split up the problem, we first count the number of control variables or decisions faced by the agent. Ideally, if the agent has n control variables, then the problem should be split into n subproblems, each handling a different control variable. For counting the number of control variables, it is important to not double count variables which are equivalent and have market clearing conditions. For example, the decision of how much to consume and how much to save may seem like two different choices, but because of the market clearing

condition $c + a = m$ they are resolved simultaneously and count as only one decision variable. Similarly, the choice between labor and leisure are simultaneous and count as only one decision.

Having counted our control variables, we look for differentiable and invertible utility functions which are separable in the dynamic programming problem, such as in Section 2 of the paper, or differentiable and invertible functions in the transition, as in Section 3 of the paper.

5.1.1 Separable utility functions

In Section 2, we have additively separable utility of consumption and leisure, which allows for each of these control variables to be handled by separate subproblems. So, it makes sense to split the utility between subproblems and attach one to the consumption subproblem and one to the leisure subproblem.

As mentioned in that section, however, there are only two separable utility functions in the problem which have been assigned to two subproblems already. This leaves one control variable without a separable utility function. In that case, there is not another Endogenous Grid Method step to exploit, and this subproblem has to be handled by standard convex optimization techniques such as maximization of the value function (VFI) or finding the root of the Euler equation (PFI).

Now that we have split the problem into conceptual subproblems, it is important to sequence them in such a way that they don't become more complex than the original problem. The key here is to avoid adding unnecessary state variables. For example, in the consumption-leisure-portfolio problem, if we were to choose consumption first, we would have to track the wage rate into the following leisure subproblem. This would mean that our consumption problem would be two-dimensional as well as our labor decision problem. As presented, the choice of order in Section 2 ensures that the consumption problem is one-dimensional, as we can shed the information about the wage rate offer after the agent has made their labor-leisure decision. If we did this the other way, the problem would be more complex and require additional computational resources.

The consumption subproblem would be two-dimensional instead of one-dimensional, adding more complexity,

$$\begin{aligned} v(b, \theta) &= \max_c u(c) + \tilde{v}(b', \theta) \\ \text{s.t.} \\ b' &= b - c \geq -\theta \end{aligned} \tag{48}$$

while the labor-leisure subproblem would have an additional constraint

$$\begin{aligned} \tilde{v}(b', \theta) &= \max_z h(z) + v(a) \\ \text{s.t.} \\ 0 &\leq z \leq 1 \\ a &= b' + \theta(1 - z) \geq 0. \end{aligned} \tag{49}$$

Therefore, strategic ordering of subproblems can greatly simplify the solution process and reduce computational the burden.

Consider the utility function of the form

$$U(y) = u_{-i}(y^{-i}) + u_i(y^i) \tag{50}$$

where y^i is the i -th control variable and y^{-i} is the vector of all control variables except the i -th one.

which is separable in the state and control variables that correspond to the index i .

$$\begin{aligned} V_t(x_t, s_t) &= \max_{y_t \in \Gamma_t(x_t, s_t)} U(y_t) + \beta \mathbb{E}_t [V_{t+1}(x_{t+1}, s_{t+1}) | \tilde{x}_t, s_t] \\ \text{s.t.} \\ \tilde{x}_t &= T_t(x_t, y_t) \\ x_{t+1} &= G_{t+1}(\tilde{x}_t, s_t) \end{aligned} \tag{51}$$

For simplicity, define

$$W_t(\tilde{x}_t, s_t) = \beta \mathbb{E}_t [V_{t+1}(G_{t+1}(\tilde{x}_t, s_t), s_{t+1}) | \tilde{x}_t, s_t] \quad (52)$$

then

$$\begin{aligned} V_t(x_t, s_t) &= \max_{y_t \in \Gamma_t(x_t, s_t)} U(y_t) + W_t(\tilde{x}_t, s_t) \\ \text{s.t.} \\ \tilde{x}_t &= T_t(x_t, y_t) \end{aligned} \quad (53)$$

the first order condition

$$\frac{\partial U(y_t)}{\partial y_t^i} + \sum_{j=1}^n \frac{\partial W_t(\tilde{x}_t, s_t)}{\partial \tilde{x}_t^j} \frac{\partial T_t^j(x_t, y_t)}{\partial y_t^i} = 0 \quad (54)$$

we require $\frac{\partial T_t^j(x_t, y_t)}{\partial y_t^i} = 0$ for $j \neq i$ to be able to solve for y_t^i .

$$\frac{\partial U(y_t)}{\partial y_t^i} + \frac{\partial W_t(\tilde{x}_t, s_t)}{\partial \tilde{x}_t^i} \frac{\partial T_t^i(x_t, y_t)}{\partial y_t^i} = 0 \quad (55)$$

5.1.2 Differentiable and invertible transition

In Section 3, we see that a problem with a differentiable and invertible transition can also be used to embed an additional Endogenous Grid Method step. Because the transition applies independently to a state variable that is not related to the other control variable, consumption, it can be handled separately from the consumption subproblem.

In this particular problem, however, it turns out to make no difference how we order the two subproblems. This is because the control variables, consumption and pension deposit, each affect a separate resource account, namely market resources and pension balance. Because of this, the two subproblems are independent of each other and can be solved in any order.

A good rule of thumb is that when splitting up a problem into subproblems, we should try to reduce the information set that is passed onto the next subproblem. In Section 2, choosing leisure-labor and realizing total market resources before consumption allows us to shed the wage rate offer state variable before the consumption problem, and we know that for the portfolio choice we only need to know liquid assets after expenditures (consumption). Thus, the order makes intuitive sense; agent first chooses leisure-labor, realizing total market resources, then chooses consumption and savings, and finally chooses their risky portfolio choice. In Section 3, there are two expenditures that are independent of each other, consumption and deposit, and making one decision or the other first does not reduce the information set for the agent, thus the order of these subproblems does not matter.

5.2 The Endogenous Grid Method for Subproblems

Once we have strategically split the problem into subproblems, we can use the Endogenous Grid Method in each applicable subproblem while iterating backwards from the terminal period. As we discussed in Sections Section 2 and Section 3, the EGM step can be applied when there is a separable, differentiable and invertible utility function in the subproblem or when there is a differentiable and invertible transition in the subproblem. We will discuss each of these cases in turn.

5.2.1 Utility function

A generic subproblem with a differentiable and invertible utility function can be characterized as follows:

$$\begin{aligned}
V(x) &= \max_{y \in \Gamma(x)} U(x, y) + \beta W(a) \\
&\text{s.t.} \\
a &= T(x, y)
\end{aligned} \tag{56}$$

For an interior solution, the first-order condition is thus

$$U'_y(x, y) + \beta W'(a) T'_y(x, y) = 0 \tag{57}$$

If, as we assumed, the utility function is differentiable and invertible, then the Endogenous Grid Method consists of

$$y = (U'_y(x, y))^{-1} [-\beta W'(a) T'_y(x, y)] \tag{58}$$

By using an exogenous grid of the post-decision state a , we can solve for the optimal decision rule y at each point on the grid. This is the Endogenous Grid Method step.

5.2.2 Transition

If the generic subproblem has no separable utility, but instead has a differentiable and invertible transition, then the Endogenous Grid Method can still be used.

$$\begin{aligned}
V(x) &= \max_{y \in \Gamma(x)} W(a) \\
&\text{s.t.} \\
a &= T(x, y)
\end{aligned} \tag{59}$$

Here, the first-order condition is

$$W'(a) T'_y(x, y) = 0 \tag{60}$$

and the Endogenous Grid Method step is

$$y = (T'_y(x, y))^{-1} [1/W'(a)] \tag{61}$$

6 Conclusion

This paper introduces a novel method for solving dynamic stochastic optimization problems called the Sequential Endogenous Grid Method (**EGM**ⁿ). Given a problem with multiple decisions (or control variables), the Sequential Endogenous Grid Method proposes separating the problem into a sequence of smaller subproblems that can be solved sequentially by using more than one **EGM** step. Then, depending on the resulting endogenous grid from each subproblem, this paper proposes different methods for interpolating functions on non-rectilinear grids, called the Warped Grid Interpolation (**WGI**) and the Gaussian Process Regression (**GPR**) method.

EGMⁿ is similar to the Nested Endogenous Grid Method (**NEGM**)⁷ and the Generalized Endogenous Grid Method (**G2EGM**)⁸ in that it can solve problems with multiple decisions, but it differs from these methods in that by choosing the subproblems strategically, we can take advantage of multiple sequential **EGM** steps to solve complex multidimensional models in a fast and efficient manner. Additionally, the use of machine learning tools such as the **GPR** overcomes bottlenecks seen in unstructured interpolation using Delaunay triangulation and other similar methods.

⁷Druehl [2021].

⁸Druehl and Jørgensen [2017].

7 Appendix: Solving the illustrative **G2EGM** model with **EGM**ⁿ

7.1 The problem for a retired household

I designate as $w_t(m_t)$ the problem of a retired household at time t with total resources m . The retired household solves a simple consumption-savings problem with no income uncertainty and a certain next period pension of $\underline{\theta}$.

$$\begin{aligned} w_t(m_t) &= \max_{c_t} u(c_t) + \beta w_{t+1}(m_t) \\ \text{s.t.} \\ a_t &= m_t - c_t \\ m_{t+1} &= R_a a_t + \underline{\theta} \end{aligned} \tag{62}$$

Notice that there is no uncertainty and the household receives a retirement income $\underline{\theta}$ every period until death.

7.2 The problem for a worker household

The value function of a worker household is

$$V_t(m_t, n_t) = \mathbb{E}_\epsilon \max \{v_t(m_t, n_t, \mathbb{W}) + \sigma_{\epsilon \in \mathbb{W}}, v_t(m_t, n_t, \mathbb{R}) + \sigma_{\epsilon \in \mathbb{R}}\} \tag{63}$$

where the choice specific problem for a working household that decides to continue working is

$$\begin{aligned} v_t(m_t, n_t, \mathbb{W}) &= \max_{c_t, d_t} u(c_t) - \alpha + \beta \mathbb{E}_t [V_{t+1}(m_{t+1}, n_{t+1})] \\ \text{s.t.} \\ a_t &= m_t - c_t - d_t \\ b_t &= n_t + d_t + g(d_t) \\ m_{t+1} &= R_a a_t + \theta_{t+1} \\ n_{t+1} &= R_b b_t \end{aligned} \tag{64}$$

and the choice specific problem for a working household that decides to retire is

$$v_t(m_t, n_t, \mathbb{R}) = w_t(m_t + n_t) \tag{65}$$

7.3 Applying the Sequential **EGM**

The first step is to define a post-decision value function. Once the household decides their level of consumption and pension deposits, they are left with liquid assets they are saving for the future and illiquid assets in their pension account which they can't access again until retirement. The post-decision value function can be defined as

$$\begin{aligned} v_t(a_t, b_t) &= \beta \mathbb{E}_t [V_{t+1}(m_{t+1}, n_{t+1})] \\ \text{s.t.} \\ m_{t+1} &= R_a a_t + \theta_{t+1} \\ n_{t+1} &= R_b b_t \end{aligned} \tag{66}$$

Then redefine the working agent's problem as

$$\begin{aligned} v_t(m_t, n_t, \mathbb{W}) &= \max_{c_t, d_t} u(c_t) - \alpha + v_t(a_t, b_t) \\ a_t &= m_t - c_t - d_t \\ b_t &= n_t + d_t + g(d_t) \end{aligned} \tag{67}$$

Clearly, the structure of the problem remains the same, and this is the problem that **G2EGM** solves. We've only moved some of the stochastic mechanics out of the problem. Now, we can apply the sequential **EGM**ⁿ method. Let the agent first decide d_t , the deposit amount into their retirement; we will call this the deposit problem, or outer loop. Thereafter, the agent will have net liquid assets of l_t and pension assets of b_t .

$$\begin{aligned} v_t(m_t, n_t, \mathbb{W}) &= \max_{d_t} \tilde{v}_t(l_t, b_t) \\ \text{s.t.} \\ l_t &= m_t - d_t \\ b_t &= n_t + d_t + g(d_t) \end{aligned} \quad (68)$$

Now, the agent can move on to picking their consumption and savings; we can call this the pure consumption problem or inner loop.

$$\begin{aligned} \tilde{v}_t(l_t, b_t) &= \max_{c_t} u(c_t) - \alpha + v_t(a_t, b_t) \\ \text{s.t.} \\ a_t &= l_t - c_t \end{aligned} \quad (69)$$

Because we've already made the pension decision, the amount of pension assets does not change in this loop and it just passes through to the post-decision value function.

7.4 Solving the problem

7.4.1 Solving the Inner Consumption Saving Problem

Let's start with the pure consumption-saving problem, which we can summarize by substitution as

$$\tilde{v}_t(l_t, b_t) = \max_{c_t} u(c_t) - \alpha + v_t(l_t - c_t, b_t) \quad (70)$$

The first-order condition is

$$u'(c_t) = v_t^a(l_t - c_t, b_t) = v_t^a(a_t, b_t) \quad (71)$$

We can invert this Euler equation as in standard **EGM** to obtain the consumption function.

$$c_t(a_t, b_t) = u'^{-1}(v_t^a(a_t, b_t)) \quad (72)$$

Again as before, $l_t(a_t, b_t) = c_t(a_t, b_t) + a_t$. To sum up, using an exogenous grid of (a_t, b_t) we obtain the trio $(c_t(a_t, b_t), l_t(a_t, b_t), b_t)$ which provides an interpolating function for our optimal consumption decision rule over the (l, b) grid. Without loss of generality, assume $l_t = l_t(a_t, b_t)$ and define the interpolating function as

$$\check{c}_t(l_t, b_t) \equiv c_t(a_t, b_t) \quad (73)$$

For completeness, we derive the envelope conditions as well, and as we will see, these will be useful when solving the next section.

$$\begin{aligned} \tilde{v}_t^l(l_t, b_t) &= v_t^a(a_t, b_t) = u'(c_t) \\ \tilde{v}_t^b(l_t, b_t) &= v_t^b(a_t, b_t) \end{aligned} \quad (74)$$

7.4.2 Solving the Outer Pension Deposit Problem

Now, we can move on to solving the deposit problem, which we can also summarize as

$$v_t(m_t, n_t, \mathbb{W}) = \max_{d_t} \tilde{v}_t(m_t - d_t, n_t + d_t + g(d_t)) \quad (75)$$

The first-order condition is

$$\tilde{v}_t^l(l_t, b_t)(-1) + \tilde{v}_t^b(l_t, b_t)(1 + g'(d_t)) = 0 \quad (76)$$

Rearranging this equation gives

$$g'(d_t) = \frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \quad (77)$$

Assuming that $g'(d)$ exists and is invertible, we can find

$$\mathfrak{d}_t(l_t, b_t) = g'^{-1} \left(\frac{\tilde{v}_t^l(l_t, b_t)}{\tilde{v}_t^b(l_t, b_t)} - 1 \right) \quad (78)$$

Using this, we can back out n_t as

$$\mathfrak{n}_t(l_t, b_t) = b_t - \mathfrak{d}_t(l_t, b_t) - g(\mathfrak{d}_t(l_t, b_t)) \quad (79)$$

and m_t as

$$\mathfrak{m}_t(l_t, b_t) = l_t + \mathfrak{d}_t(l_t, b_t) \quad (80)$$

In sum, given an exogenous grid (l_t, b_t) we obtain the triple $(\mathfrak{m}_t(l_t, b_t), \mathfrak{n}_t(l_t, b_t), \mathfrak{d}_t(l_t, b_t))$, which we can use to create an interpolator for the decision rule d_t .

To close the solution method, the envelope conditions are

$$\begin{aligned} v_t^m(m_t, n_t, \mathbb{W}) &= \tilde{v}_t^l(l_t, b_t) \\ v_t^n(m_t, n_t, \mathbb{W}) &= \tilde{v}_t^b(l_t, b_t) \end{aligned} \quad (81)$$

7.5 Is g invertible?

We've already seen that $u'(\cdot)$ is invertible, but is g ?

$$g(d) = \chi \log(1 + d) \quad g'(d) = \frac{\chi}{1 + d} \quad g'^{-1}(y) = \chi/y - 1 \quad (82)$$

7.6 The Post-Decision Value and Marginal Value Functions

$$\begin{aligned} v_t(a, b) &= \beta \mathbb{E}_t [V(m_{t+1}, n_{t+1})] \\ \text{s.t.} \\ m_{t+1} &= R_a a_t + \theta_{t+1} \\ n_{t+1} &= R_b b_t \end{aligned} \quad (83)$$

and

$$\begin{aligned}
\mathbf{v}_t^a(a_t, b_t) &= \beta R_a \mathbb{E}_t [V_{t+1}^m(m_{t+1}, n_{t+1})] \\
&\text{s.t.} \\
m_{t+1} &= R_a a_t + \theta_{t+1} \\
n_{t+1} &= R_b b_t
\end{aligned} \tag{84}$$

and

$$\begin{aligned}
\mathbf{v}_t^b(a_t, b_t) &= \beta R_b \mathbb{E}_t [V_{t+1}^n(m_{t+1}, n_{t+1})] \\
&\text{s.t.} \\
m_{t+1} &= R_a a_t + \theta_{t+1} \\
n_{t+1} &= R_b b_t
\end{aligned} \tag{85}$$

7.7 Taste Shocks

From discrete choice theory and from **DCEGM** paper, we know that

$$\mathbb{E}_t [V_{t+1}(m_{t+1}, n_{t+1}, \epsilon_{t+1})] = \sigma \log \left[\sum_{\mathbb{D} \in \{\mathbb{W}, \mathbb{R}\}} \exp \left(\frac{v_{t+1}(m_{t+1}, n_{t+1}, \mathbb{D})}{\sigma_\epsilon} \right) \right] \tag{86}$$

and

$$\mathbb{P}_t(\mathbb{D} \mid m_{t+1}, n_{t+1}) = \frac{\exp(v_{t+1}(m_{t+1}, n_{t+1}, \mathbb{D})/\sigma_\epsilon)}{\sum_{\mathbb{D} \in \{\mathbb{W}, \mathbb{R}\}} \exp \left(\frac{v_{t+1}(m_{t+1}, n_{t+1}, \mathbb{D})}{\sigma_\epsilon} \right)} \tag{87}$$

the first-order conditions are therefore

$$\dot{\mathbf{v}}_t^m(m_{t+1}, n_{t+1}) = \sum_{\mathbb{D} \in \{\mathbb{W}, \mathbb{R}\}} \mathbb{P}_t(\mathbb{D} \mid m_{t+1}, n_{t+1}) \mathbf{v}_{t+1}^m(m_{t+1}, n_{t+1}, \mathbb{D}) \tag{88}$$

Acknowledgments

I would like to thank Chris Carroll, Matthew White, and Simon Scheidegger for their helpful comments and suggestions. The remaining errors are my own. All figures and other numerical results were produced using the **Econ-ARK/HARK** toolkit [Carroll et al., 2018]. Additional libraries used in the production of this paper include but are not limited to: **scipy** [Virtanen et al., 2020], **numpy** [Harris et al., 2020], **numba** [Lam et al., 2015], **cupy** [Okuta et al., 2017], **scikit-learn** [Pedregosa et al., 2011], **pytorch** [Paszke et al., 2019], and **gpytorch** [Gardner et al., 2018].

References

- Christopher D Carroll. The method of endogenous gridpoints for solving dynamic stochastic optimization problems. *Economics letters*, 91(3):312–320, 6 2006. ISSN 0165-1765, 1873-7374. doi:[10.1016/j.econlet.2005.09.013](https://doi.org/10.1016/j.econlet.2005.09.013).
- Francisco Barillas and Jesús Fernández-Villaverde. A generalization of the endogenous grid method. *Journal of economic dynamics control*, 31(8):2698–2712, 8 2007. ISSN 0165-1889, 1879-1743. doi:[10.1016/j.jedc.2006.08.005](https://doi.org/10.1016/j.jedc.2006.08.005).
- Lilia Maliar and Serguei Maliar. Envelope condition method versus endogenous grid method for solving dynamic programming problems. *Economics letters*, 120(2):262–266, 8 2013. ISSN 0165-1765. doi:[10.1016/j.econlet.2013.04.031](https://doi.org/10.1016/j.econlet.2013.04.031).

- Giulio Fella. A generalized endogenous grid method for non-smooth and non-concave problems. *Review of economic dynamics*, 17(2):329–344, 4 2014. ISSN 1094-2025. doi:[10.1016/j.red.2013.07.001](https://doi.org/10.1016/j.red.2013.07.001).
- Matthew N White. The method of endogenous gridpoints in theory and practice. *Journal of economic dynamics control*, 60:26–41, 11 2015. ISSN 0165-1889, 1879-1743. doi:[10.1016/j.jedc.2015.08.001](https://doi.org/10.1016/j.jedc.2015.08.001).
- Fedor Iskhakov, Thomas H Jørgensen, John Rust, and Bertel Schjerning. The endogenous grid method for discretecontinuous dynamic choice models with (or without) taste shocks. *Quantitative economics*, 8(2): 317–365, 7 2017. ISSN 1759-7323, 1759-7331. doi:[10.3982/QE643](https://doi.org/10.3982/QE643).
- Thomas Hintermaier and Winfried Koeniger. The method of endogenous gridpoints with occasionally binding constraints among endogenous variables. *Journal of economic dynamics control*, 34(10):2074–2088, 10 2010. ISSN 0165-1889. doi:[10.1016/j.jedc.2010.05.002](https://doi.org/10.1016/j.jedc.2010.05.002).
- Thomas H Jørgensen. Structural estimation of continuous choice models: Evaluating the EGM and MPEC. *Economics letters*, 119(3):287–290, 6 2013. ISSN 0165-1765. doi:[10.1016/j.econlet.2013.02.027](https://doi.org/10.1016/j.econlet.2013.02.027).
- Fedor Iskhakov. Multidimensional endogenous gridpoint method: Solving triangular dynamic stochastic optimization problems without root-finding operations. *Economics letters*, 135:72–76, 10 2015. ISSN 0165-1765. doi:[10.1016/j.econlet.2015.07.033](https://doi.org/10.1016/j.econlet.2015.07.033).
- Alexander Ludwig and Matthias Schön. Endogenous Grids in Higher Dimensions: Delaunay Interpolation and Hybrid Methods. *Computational Economics*, 51(3):463–492, 3 2018. ISSN 0927-7099. doi:[10.1007/s10614-016-9611-2](https://doi.org/10.1007/s10614-016-9611-2).
- Cristina Arellano, Lilia Maliar, Serguei Maliar, and Viktor Tsyrennikov. Envelope condition method with an application to default risk models. *Journal of economic dynamics control*, 69:436–459, 8 2016. ISSN 0165-1889, 1879-1743. doi:[10.1016/j.jedc.2016.05.016](https://doi.org/10.1016/j.jedc.2016.05.016).
- Jeppe Druedahl and Thomas Høgholm Jørgensen. A general endogenous grid method for multi-dimensional models with non-convexities and constraints. *Journal of economic dynamics control*, 74:87–107, 1 2017. ISSN 0165-1889. doi:[10.1016/j.jedc.2016.11.005](https://doi.org/10.1016/j.jedc.2016.11.005).
- Andrew Clausen and Carlo Strub. Reverse calculus and nested optimization. *Journal of economic theory*, 187(105019):105019, 5 2020. ISSN 0022-0531, 1095-7235. doi:[10.1016/j.jet.2020.105019](https://doi.org/10.1016/j.jet.2020.105019).
- Jeppe Druedahl. A Guide on Solving Non-convex Consumption-Saving Models. *Computational Economics*, 58(3):747–775, 10 2021. ISSN 0927-7099. doi:[10.1007/s10614-020-10045-x](https://doi.org/10.1007/s10614-020-10045-x).
- Zvi Bodie, Robert C Merton, and William F Samuelson. Labor supply flexibility and portfolio choice in a life cycle model. *Journal of economic dynamics control*, 16(3-4):427–449, 7 1992. ISSN 0165-1889. doi:[10.1016/0165-1889\(92\)90044-F](https://doi.org/10.1016/0165-1889(92)90044-F).
- Karel Mertens and Morten O Ravn. Understanding the aggregate effects of anticipated and unanticipated tax policy shocks. *Review of economic dynamics*, 14(1):27–54, 1 2011. ISSN 1094-2025. doi:[10.1016/j.red.2010.07.004](https://doi.org/10.1016/j.red.2010.07.004).
- Christopher D Carroll. Precautionary saving and the marginal propensity to consume out of permanent income. *Journal of monetary economics*, 56(6):780–790, 9 2009. ISSN 0304-3932. doi:[10.1016/j.jmoneco.2009.06.016](https://doi.org/10.1016/j.jmoneco.2009.06.016).
- Simon Scheidegger and Ilias Bilionis. Machine learning for high-dimensional dynamic stochastic economies. *Journal of computational science*, 33:68–82, 4 2019. ISSN 1877-7503. doi:[10.1016/j.jocs.2019.03.004](https://doi.org/10.1016/j.jocs.2019.03.004).
- Jacob R Gardner, Geoff Pleiss, David Bindel, Kilian Q Weinberger, and Andrew Gordon Wilson. Gpytorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In *Advances in Neural Information Processing Systems*, 2018.
- Christopher Carroll, Alexander Kaufman, Jacqueline Kazil, Nathan Palmer, and Matthew White. The econ-ARK and HARK: Open source tools for computational economics. In Fatih Akici, David Lippa, Dillon Niederhut, and M Pacer, editors, *Proceedings of the Python in Science Conference*, pages 25–30, Austin, Texas, 2018. SciPy. doi:[10.25080/majora-4af1f417-004](https://doi.org/10.25080/majora-4af1f417-004).
- Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J van der Walt, Matthew Brett, Joshua Wilson, K Jarrod Millman, Nikolay Mayorov, Andrew R J Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E A Quintero, Charles R Harris, Anne M Archibald, Antônio H Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. Scipy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17(3):261–272, 3 2020. ISSN 1548-7091, 1548-7105. doi:[10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2).

- Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández Del Rio, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 9 2020. ISSN 0028-0836, 1476-4687. doi:[10.1038/s41586-020-2649-2](https://doi.org/10.1038/s41586-020-2649-2).
- Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: a LLVM-based Python JIT compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, number Article 7 in LLVM '15, pages 1–6, Austin, Texas, 11 2015. Association for Computing Machinery. ISBN 9781450340052. doi:[10.1145/2833157.2833162](https://doi.org/10.1145/2833157.2833162).
- Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. Cupy: A numpy-compatible library for nvidia gpu calculations. In *Proceedings of workshop on machine learning systems (LearningSys) in the thirty-first annual conference on neural information processing systems (NIPS)*, volume 5, 2017.
- F Pedregosa, G Varoquaux, A Gramfort, V Michel, B Thirion, O Grisel, M Blondel, P Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-Learn: Machine Learning in Python. *Journal of machine learning research: JMLR*, 12:2825–2830, 2011. ISSN 1532-4435, 1533-7928.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and Others. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019. ISSN 1049-5258.