
Alan Lujan¹

Johns Hopkins University AAP
Econ-ARK

Saturday 19th 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

An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

– [\[1\]\(doi:10.1090/S0002-9904-1954-09848-8\)](https://doi.org/10.1090/S0002-9904-1954-09848-8)

¹Correspondence to: alujan@jhu.edu

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.

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

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

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. [Druehl 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. [Druehl \[2021\]](#) 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 [Druehl \[2021\]](#).

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

The Sequential Endogenous Grid Method (**EGMⁿ**) is a novel extension of the **EGM** to solve dynamic programming problems with multiple choice variables. The key insight is to break down the problem into a sequence of subproblems, each of which can be solved using a simple **EGM** step. To demonstrate the power of thinking sequentially, I first present a simple model with one choice variable that is traditionally solved using one **EGM** step.

2.1 The Standard Incomplete Markets (SIM) Model as a sequential problem

Consider the standard one-asset incomplete markets model by [Krusell and Smith \[1998\]](#), also known as the Bewley-Huggett-Aiyagari-Imrohoroglu-Zeldes-Deaton-Carroll model. The consumer's problem is to maximize the present discounted value of utility from consumption and savings, subject to uncertainty in labor productivity and an aggregate process of interest rates and wages. The Bellman equation is

$$V_t(e, k) = \max_{c, k'} \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_t [V_{t+1}(e', k') | e] \right\} \quad (1)$$

$$c + k' = (1 + r_t)k + w_t e \quad (2)$$

$$k' \geq 0 \quad (3)$$

We can think of this problem as a sequence of two blocks: one where the household makes a consumption-saving decision, and another where the household takes an expectation of the future conditional on their savings choice. Define the continuation value of any savings choice as

$$W_t(e, k') = \mathbb{E}_t [V_{t+1}(e', k') | e]. \quad (4)$$

Importantly, this block does not require any optimization, as it is just an expectation step. The household can then use the value of any savings decision to optimize their consumption choice. The problem is then reduced to the following:

$$V_t(e, k) = \max_{c, k'} \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta W_t(e, k') \right\} \quad (5)$$

$$c + k' = (1 + r_t)k + w_t e \quad (6)$$

$$k' \geq 0 \quad (7)$$

Notice that this block has no uncertainty. This is a key advantage of thinking sequentially. By placing the expectation in a subsequent block, this block is now a simple optimization problem.

We have now dissected this problem into two blocks, an optimization block and an expectation block, each of which is easier to handle than the whole. We now present the **EGM**. The first-order condition with respect to consumption is

$$c(e, k') = (\beta W_t^k(e, k'))^{-1/\sigma} \quad (8)$$

[Carroll \[2024\]](#) refer to this as the “consumed function”, because it provides the amount a household must have “consumed” to optimally achieve a certain level of savings k' given their current productivity state e . From this function, we can derive the endogenous grid of capital k that the agent must have started with to consume $c(e, k')$ and save k' using the resource constraint.

$$k(e, k') = \frac{c(e, k') + k' - w_t e}{1 + r_t} \quad (9)$$

These two equations now jointly define a parameterized curve² for the consumption function $c(e, k)$, which can be interpolated to find the optimal consumption for any level of savings and productivity. This is the **EGM** step.

2.2 The Heterogeneous Agent New Keynesian (HANK) Model

The baseline problem which I will use to demonstrate the Sequential Endogenous Grid Method (**EGM**ⁿ) is the one-asset HANK model with endogenous labor from [Auclert et al. \[2021\]](#).

In particular, this example makes use of an additively separable utility of consumption and disutility of labor as follows:

The use of additively separable utility is important, 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 w_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.

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

2.3 Restating the problem sequentially

Before we continue, it will be useful to introduce a new notation for timing. In dynamic programming, we often think of the period- t problem. However, when thinking sequentially, it will be useful to think about breaking the period into sub-periods, which we'll call moments. When using **EGM**ⁿ, we will index these moments as (t, τ) for the period- t and moment- τ subproblem. We will also use lowercase letters to denote the moment-specific value functions, $v_{(t, \tau)}(\cdot)$, and the decision rules, $\pi_{(t, \tau)}(\cdot)$.

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

The pure consumption-saving problem is then

²This comes from Fedor Iskhakov.

Finally, the expectation block is

Notice that we started with moment $(t, 1)$. We want to think of moment $(t, 0)$ as the moment where uncertainty is resolved. Therefore, the agent first learns r_t and $w_t(e)$ in moment $(t, 0)$ and then uses this information to solve the consumption-savings problem in moment $(t, 1)$.

This sequential approach is explicitly modeled after the nested approaches explored in [Clausen and Strub \[2020\]](#) and [Druehl \[2021\]](#). 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.

2.4 The labor-leisure subproblem

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

$$v_t(t, t) = \max_t (v_t(t) + t (v_{t+1}(1-t))) \quad (10)$$

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

$$v'_t(t) = v'_{t+1}(t) \quad (11)$$

The marginal utility of leisure and its inverse are

$$v'_t(t) = -v'^{-1}_t(t) = (v'_t(t))^{-1/} \quad (12)$$

Using an exogenous grid of t and v , we can find leisure as

$$t_t(v) = v'^{-1}_t(v'_t(t)) \quad (13)$$

In this case, it's important to note that there are conditions for leisure itself. An agent with a small level of market resources t might want to work more than their available time endowment, especially at higher levels of income 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 $\hat{t}_t(v)$, where

$$\hat{t}(\cdot) = \max[\min[t(\cdot), 1], 0] \quad (14)$$

Then, we derive labor as $t(t, t) = 1 - \hat{t}(t, t)$. Finally, for each t and t as an exogenous grid, we can find the endogenous grid of bank balances as $t(t, t) = t - tt(t, t)$.

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

$$\frac{b}{t}(t, t) = \frac{b}{t}(t) = \frac{b}{t}(t)/t. \quad (15)$$

$$v_{(t,0)}(e, a_-) = \max_n \left\{ -\varphi \frac{n^{1+\nu}}{1+\nu} + v_{(t,1)}(e, (1+r_t)a_- + w_t(e)n) \right\} \quad (16)$$

$$\varphi n^\nu = v_{(t,1)}^m(e, m) w_t(e) \quad (\text{euler})$$

$$n(e_\times, m_\times) = \left(\frac{w_t(e_\times)}{\varphi} v_{(t,1)}^m(e_\times, m_\times) \right)^{\frac{1}{\nu}} \quad (\text{egm})$$

$$a_-(e_\times, m_\times) = \frac{m_\times - w_t(e_\times)n(e_\times, m_\times)}{1+r_t} \quad (\text{budget})$$

$$a_-(e_\times, m_\times) \Rightarrow m(e_\times, a_{-\cap}) \quad n(e_\times, m_\times) \Rightarrow n(e_\times, a_{-\cap}) \quad (\text{swap dims})$$

$$n(e_\times, a_{-\cap}) \Rightarrow n(e_\times, a_{-\times}) \quad m(e_\times, a_{-\cap}) \Rightarrow m(e_\times, a_{-\times}) \quad (\text{regrid})$$

$$v_{(t,0)}^{a_-}(e, a_-) = (1+r_t) \left(\frac{\varphi n^\nu}{w_t(e)} \right) \quad (\text{ec})$$

2.5 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:

$$t(t) = \max_t (t) + t(t-t) \quad (17)$$

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

$$v'(t) = v'_t(t-t) = v'_t(t) \quad (18)$$

Inverting the above equation is the (first) **EGM** step.

$$t(t) ='^{-1} (t'(t)) \quad (19)$$

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

$$t'() = - \quad t'^{-1}() = -1/ \quad (20)$$

Carroll [2006] demonstrates that by using an exogenous grid of points we can find the unique $t()$ 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

$$t() = t() + \quad (21)$$

This $t()$ is the “endogenous” grid that is consistent with the exogenous decision grid. Now that we have a $(t(), t())$ pair for each \in , 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.

$$t'(t) = t'(t) = t'(t) \quad (22)$$

$$v_{(t,1)}(e, m) = \max_c \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta v_{(t,2)}(e, m - c) \right\} \quad (23)$$

$$c^{-\sigma} = \beta v_{(t,2)}^a(e, a) \quad (\text{euler})$$

$$c(e_{\times}, a_{\times}) = \left(\beta v_{(t,2)}^a(e_{\times}, a_{\times}) \right)^{-\frac{1}{\sigma}} \quad (\text{egm})$$

$$m(e_{\times}, a_{\times}) = c(e_{\times}, a_{\times}) + a_{\times} \quad (\text{budget})$$

$$m(e_{\times}, a_{\times}), c(e_{\times}, a_{\times}) \Rightarrow a(e_{\times}, m_{\cap}), c(e_{\times}, m_{\cap}) \quad (\text{swap dims})$$

$$c(e_{\times}, m_{\cap}) \Rightarrow c(e_{\times}, m_{\times}) \quad a(e_{\times}, m_{\cap}) \Rightarrow a(e_{\times}, m_{\times}) \quad (\text{regrid})$$

$$v_{(t,1)}^m(e, m) = c^{-\sigma} \quad (\text{ec})$$

exogenous endogenous

3 The EGMⁿ 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 EGMⁿ be used to solve higher dimensional problems? In short, yes, but it requires additional thought on interpolation.

3.1 The two-asset HANK model

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 c_t and an illiquid account of retirement savings a_t . The worker maximizes their utility by choosing consumption c_t and pension deposit b_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 (e_{t+1}) and a transitory (ϵ_{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:

$$V_t(e, b, a) = \max_{c, b', a'} \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta \mathbb{E}_t V_{t+1}(e', b', a') \right\} \quad (1)$$

$$c + a' + b' = z_t(e) + (1 + r_t^a)a + (1 + r_t^b)b - \Psi(a', a) \quad (24)$$

$$a' \geq \underline{a}, \quad b' \geq \underline{b}, \quad (25)$$

where

$$\Psi(a', a) = \left(\frac{\chi_1}{2} \right) \frac{(a' - (1 + r_t^a)a)^2}{(1 + r_t^a)a + \chi_0} \quad (26)$$

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 c_t and retirement balance a_t by choosing a pension deposit b_t , which must be positive. The retirement balance a_t is the cash value of their retirement account plus their pension deposit and an additional amount g_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.

$$v_{(t,0)}(e, b, a) = \max_{a'} v_{(t,1)}(e, \ell, a') \quad (1)$$

$$m = z_t(e) + (1 + r_t^a)a + (1 + r_t^b)b \quad (27)$$

$$\ell = m - a' - \Psi(a', a) \quad (28)$$

$$a' \geq \underline{a} \quad (29)$$

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

$$v_{(t,1)}(e, \ell, a') = \max_c \left\{ \frac{c^{1-\sigma}}{1-\sigma} + \beta v_{(t,2)}(e, b', a') \right\} \quad (1)$$

$$b' = \ell - c \quad (30)$$

$$b' \geq \underline{b} \quad (31)$$

Finally, the post-decision value function $_t$ 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.

$$v_{(t,2)}(e, b', a') = \mathbb{E}_t V_{t+1}(e', b', a') \quad (32)$$

The advantage of conceptualizing this subproblem as a separate stage is that we can construct a function $_t$ 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 $_t$ 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 $_t$ 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:

$$t(t, t) = \max_t (t - t, t + t + (t)) \quad (33)$$

The first-order condition is

$$t(t, t)(-1) + t(t, t)(1 + ' (t)) = 0. \quad (34)$$

Rearranging this equation gives

$$' (t) = \frac{t(t, t)}{t(t, t)} - 1 \quad (35)$$

where

$$' () = \frac{1}{1+} \quad '^{-1}(y) = /y - 1 \quad (36)$$

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

$$t(t, t) = '^{-1} \left(\frac{t(t, t)}{t(t, t)} - 1 \right) \quad (37)$$

Using this, we can back out t as

$$t(t, t) = t - t(t, t) - (t(t, t)) \quad (38)$$

and t as

$$t(t, t) = t + t(t, t) \quad (39)$$

In sum, given an exogenous grid (t, t) we obtain the triple $(t(t, t), t(t, t), t(t, t))$, which we can use to create an interpolator for the decision rule t .

To close the solution method, the envelope conditions are

$$\begin{aligned} t(t, t) &= t(t, t) \\ t(t, t) &= t(t, t) \end{aligned} \quad (40)$$

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 1 that starting from a regular and rectilinear exogenous grid of liquid assets post-consumption t and pension balances post-deposit t , we obtain Figure 2 which shows an irregular and unstructured endogenous grid of market resources t and pension balances pre-deposit t .

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

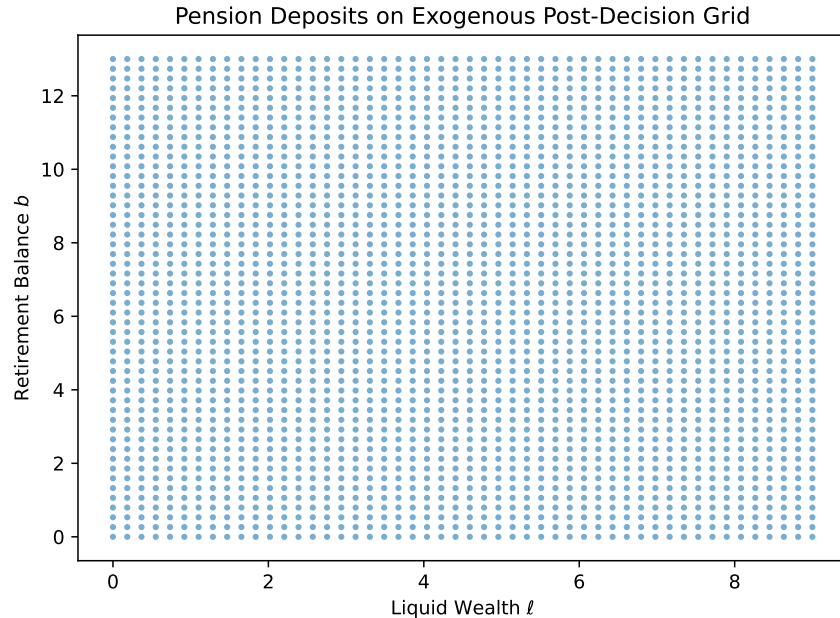


Figure 1: A regular, rectilinear exogenous grid of pension balances after deposit t and liquid assets after consumption t .

4 Multivariate Interpolation on Unstructured Grids

4.1 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 results in a one-dimensional endogenous grid of total market resources. However, as we know from standard **EGM**, the spacing in the grid is different from the spacing in the 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.

4.2 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

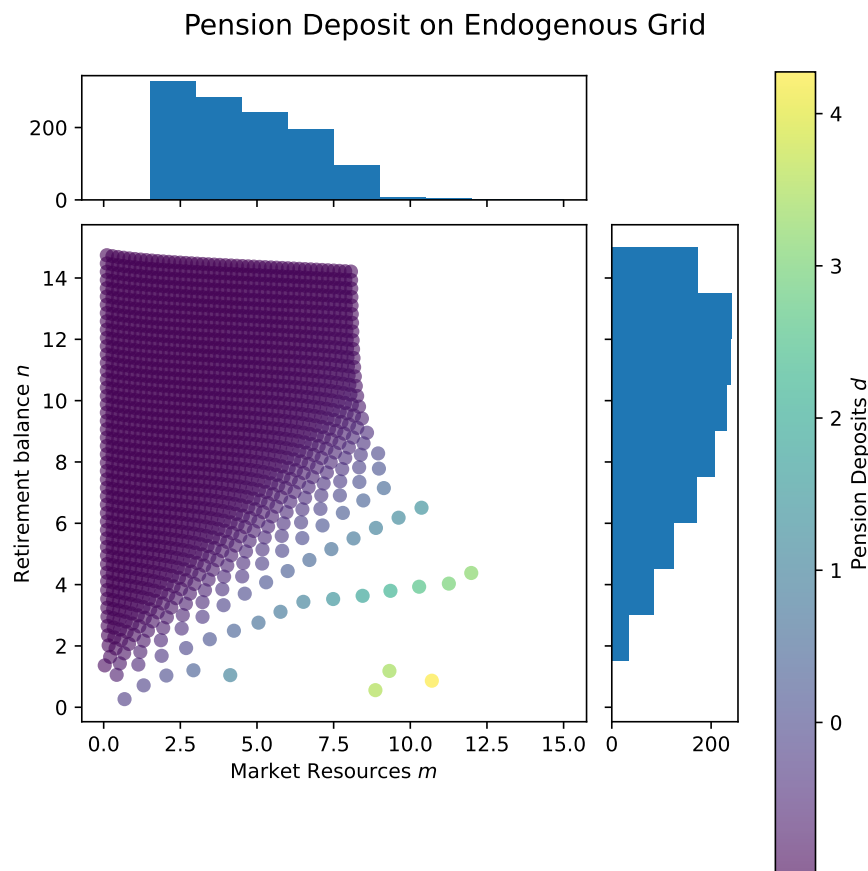


Figure 2: An irregular, unstructured endogenous grid of market resources t and pension balances before deposit t .

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 4, 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).

In Figure 5 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.

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 6. If we could find the corresponding point in the coordinate grid, interpolation would be straightforward. We

³For this illustration, we generate z 's arbitrarily using the function $f(x, y) = (xy)^{1/4}$.

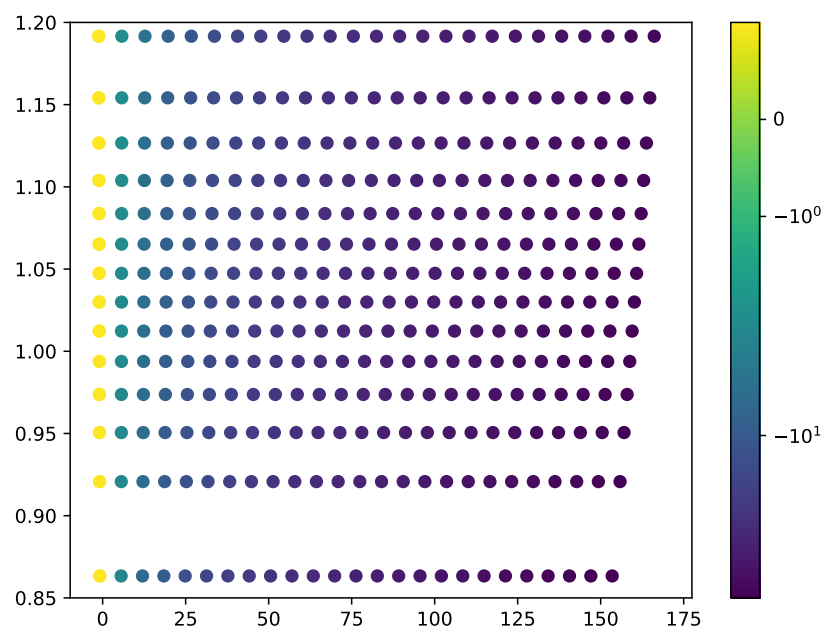


Figure 3: Warped Curvilinear Grid that results from multivariate **EGM**. This grid can be interpolated by **WGI**.

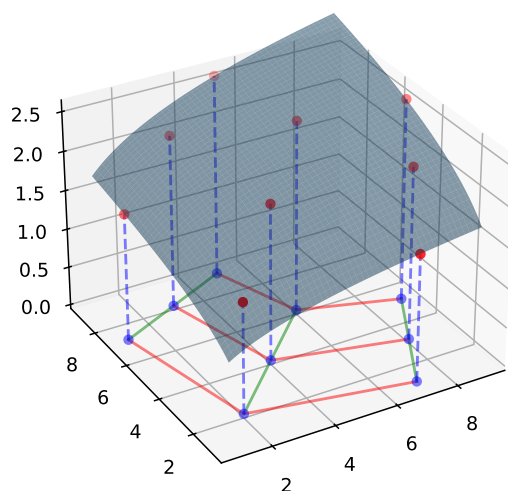


Figure 4: True function and curvilinear grid of points for which we know the value of the function.

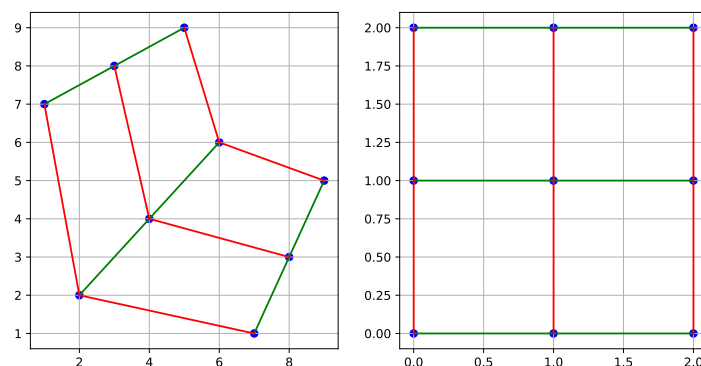


Figure 5: Homotopy between the curvilinear grid and the index coordinates of the matrix.

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 6, 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.

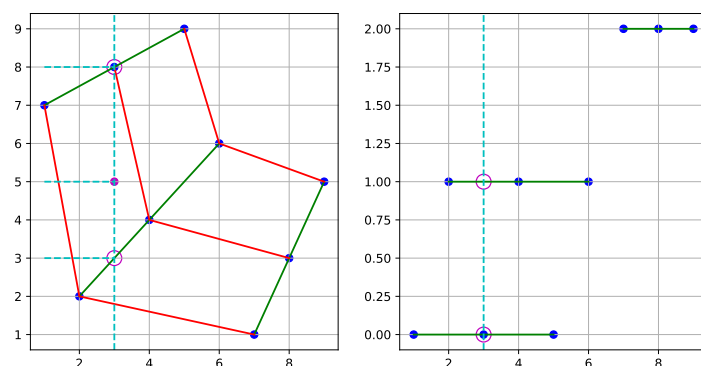


Figure 6: The method consist of extending the loci of points in the x dimension to find the corresponding crossing points in the y dimension.

\\\$\\\$

\\\$\\\$

\\begin{equation}

\\utilFunc{\\cRat_{t}, \\leisure_{t}} = \\PLev_{t}^{\\{1 - \\CRRA\\}} \\frac{\\cRat_{t}^{\\{1 - \\CRRA\\}}}{\\{1 - \\CRRA\\}} + (\\lab

\\end{equation}

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

\$\$

\$\backslash\$

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 \[2019\]](#).

4.3 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 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.

4.3.1 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.

$$\begin{aligned} \mathbf{X} &\sim \mathcal{N}(\mu, \Sigma) \quad \text{s.t.} \quad x_i \sim \mathcal{N}(\mu_i, \sigma_{ii}) \\ \text{and} \quad \sigma_{ij} &= [(x_i - \mu_i)(x_j - \mu_j)] \quad \forall i, j \in \{1, \dots, n\}. \end{aligned} \tag{41}$$

where

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} \quad \Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1n} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1} & \sigma_{n2} & \cdots & \sigma_{nn} \end{bmatrix}. \tag{42}$$

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.

⁵[Gardner et al. \[2018\]](#)

$$\mathbb{P}(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}|\mathbf{m}, \mathbf{K}) \quad (43)$$

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

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp \left(-\frac{1}{2l^2} (\mathbf{x}_i - \mathbf{x}_j)' (\mathbf{x}_i - \mathbf{x}_j) \right). \quad (44)$$

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.

4.3.2 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.

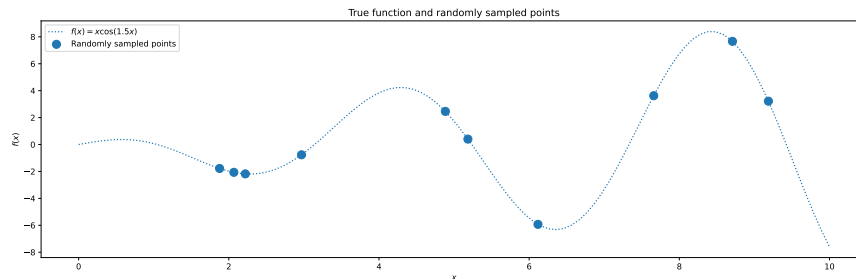


Figure 7: The true function that we are trying to approximate and a sample of data points.

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 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.

⁶For details see notebook.

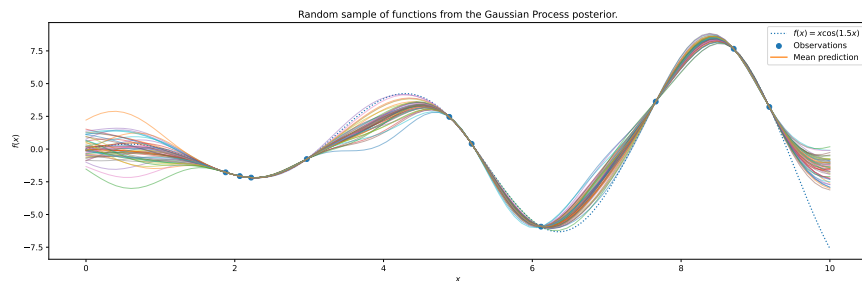


Figure 8: A random sample of functions from the **GPR** posterior that fit the data. The goal of **GPR** is to find the function that best fits the data.

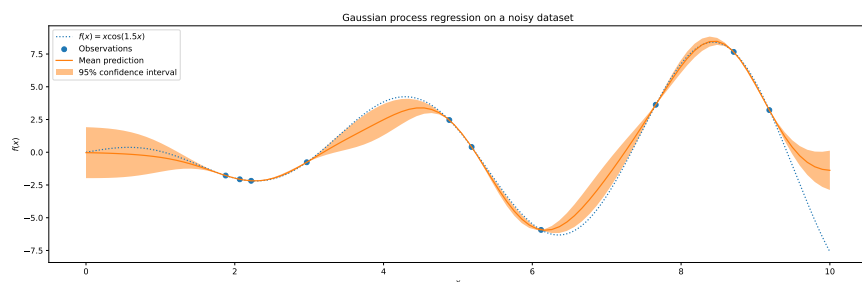


Figure 9: **GPR** finds the function that best fits the data given some hyperparameters. **GPR** then optimizes over the parameter space to find the function that minimizes the negative log-likelihood of the data.

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 $+$ $=$ 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, while the labor-leisure subproblem would have an additional constraint

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

Consider the utility function of the form

where i is the i -th control variable and $-i$ 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 .

For simplicity, define

then

the first order condition

we require $\frac{\partial^j v(\mathbf{t}, \mathbf{t})}{\partial^i} = 0$ for $j \neq i$ to be able to solve for i .

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:

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

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

By using an exogenous grid of the post-decision state , we can solve for the optimal decision rule 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.

Here, the first-order condition is

and the Endogenous Grid Method step is

5.3 Extending the Triangular Dynamic Optimization Theorem

In this section, we extend the Triangular Dynamic Optimization Theorem from Iskhakov [2015] to allow for a more general class of multidimensional dynamic optimization problems.

Consider a dynamic stochastic optimization problem with m endogenous and continuous state variables $x_t = (x_t^1, x_t^2, \dots, x_t^m)'$ and n exogenous (and potentially discrete) state variables⁷ $\varepsilon_t = (\varepsilon_t^1, \varepsilon_t^2, \dots, \varepsilon_t^n)'$ which follow a Markov process $\varepsilon_{t+1}|\varepsilon_t \sim F$. The agent solves this problem by choosing m continuous action variables $a_t = (a_t^1, a_t^2, \dots, a_t^m)'$ simultaneously. Let $y_t = (y_t^1, y_t^2, \dots, y_t^m)'$ be the vector of post-decision state variables, such that:

$$y_t = g(x_t, a_t, \varepsilon_t). \quad (45)$$

This implies that our post-decision state vector is a *sufficient statistic* of the current period's state and action, conditional on the exogenous state vector, which means that it contains all the information necessary to determine the next period's state. This can be seen in the following expression

$$x_{t+1} = f(y_t, \xi_{t+1}) \quad (46)$$

where ξ_{t+1} is a vector of exogenous, independent, and idiosyncratic shocks to the agent's problem. We will call $g(\cdot)$ the *transition function* and $f(\cdot)$ the *next-period state function*.

The objective of the agent is to maximize the present discounted utility of its actions $u(a, \varepsilon)$ as follows:

$$v_0(x_0, \varepsilon_0) = \max_{a_t} \sum_{t=0}^T \beta^t u(a_t, \varepsilon_t). \quad (47)$$

Finally, we can define the recursive dynamic programming problem as:

$$v_t(x_t, \varepsilon_t) = \max_{a_t \in \Gamma_t(x_t, \varepsilon_t)} \{u(a_t, \varepsilon_t) + \beta \mathbb{E}_t[v_{t+1}(x_{t+1}, \varepsilon_{t+1})|x_t, a_t, \varepsilon_t]\} \quad (48)$$

$$y_t = g(x_t, a_t, \varepsilon_t) \quad (49)$$

$$x_{t+1} = f(y_t, \xi_{t+1}) \quad (50)$$

where $\Gamma_t(\cdot)$ is the set of feasible actions at time t .

5.3.1 The Continuation Value Function

The first set of assumptions we make will allow us to reduce the problem to a non-stochastic one.

⁷The exogenous state vector can be composed of both discrete and continuous state variables, which will not affect the following analysis. Alternatively, we can think of this exogenous state vector as being aggregate shocks to the agent's environment.

This assumption allows us to factor the expectation out of the recursive problem since $\mathbb{E}_t[\cdot|x_t, a_t, \varepsilon_t] = \mathbb{E}_t[\cdot|y_t, \varepsilon_t]$. Then, we can define the following function

$$w(y_t, \varepsilon_t) = \mathbb{E}_t[v_{t+1}(x_{t+1}, \varepsilon_{t+1})|y_t, \varepsilon_t] \quad (51)$$

$$x_{t+1} = f(y_t, \xi_{t+1}) \quad (52)$$

and substitute it into the original problem to obtain the following problem

$$v_t(x_t, \varepsilon_t) = \max_{a_t \in \Gamma_t(x_t, \varepsilon_t)} \{u(a_t, \varepsilon_t) + \beta w(y_t, \varepsilon_t)\} \quad (53)$$

$$y_t = g(x_t, a_t, \varepsilon_t) \quad (54)$$

This is now a non-stochastic problem, which simplifies our analysis considerably. We will refer to $w(\cdot)$ as the *continuation value function*, which is the value function evaluated before the realization of any uncertainty. Our problem is now reduced to maximizing the trade-off between the utility of the action a_t and the discounted continuation value given the consequences of said action.

This assumption allows the existence of a continuous derivative of $w(\cdot)$ with respect to y_t , which is necessary for the existence of the first-order condition.

$$\nabla w(y_t, \varepsilon_t) = \mathbb{E}_t[\nabla v_{t+1}(f(y_t, \xi_{t+1}), \varepsilon_{t+1})|y_t, \varepsilon_t] \quad (55)$$

5.3.2 Standard Assumptions for EGM

As in the standard EGM, we require the utility function to be strictly concave and differentiable. This guarantees that the first-order condition is both necessary and sufficient for a maximum. In the examples presented in this paper, we used additively separable utility functions, which satisfy these conditions.

This condition allows us to work backwards from the post-decision state y_t and the action a_t to obtain the pre-decision state x_t that optimally corresponds to them.

As Iskhakov [2015] shows, more assumptions are needed for the multidimensional case.

5.3.3 Triangular Assumptions

5.3.4 Triangular Dynamic Optimization

Theorem 5.1 (Triangular Dynamic Optimization). *Under assumptions 1-5, the dynamic optimization problem is triangular and admits a solution method which avoids all root-finding operations.*

Proof.

$$u'_j(a_t) = \beta \frac{\partial w(y_t, \varepsilon_t)}{\partial y_t^j} \frac{\partial g^j(x_t, a_t, \varepsilon_t)}{\partial a_t^j} \quad (56)$$

□

5.3.5 Sequential Dynamic Optimization

Theorem 5.2 (Sequential Dynamic Optimization). *Under assumptions 1-5, the dynamic optimization problem is triangular and can be solved sequentially.*

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**)⁸ and the Generalized Endogenous Grid Method (**G2EGM**)⁹ 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.

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

7.1 The problem for a retired household

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

$$V_t(e, b, a) = \max_{c, n, b', a'} \left\{ \frac{c^{1-\sigma}}{1-\sigma} - \varphi \frac{n^{1+\nu}}{1+\nu} + \beta \mathbb{E}_t V_{t+1}(e', b', a') \right\} \quad (1)$$

$$c + a' + b' = z_t(e) + (1 + r_t^a)a + (1 + r_t^b)b - \Psi(a', a) \quad (57)$$

$$a' \geq \underline{a}, \quad b' \geq \underline{b}, \quad (58)$$

Notice that there is no uncertainty and the household receives a retirement income \cdot every period until death.

7.2 The problem for a worker household

The value function of a worker household is

⁸Druehl [2021].

⁹Druehl and Jørgensen [2017].

$$v_t(t, t) = \max_{\{t(t, t) + \sigma, t(t, t) + \sigma\}} (59)$$

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

$$\begin{aligned} v_t(t, t) &= \max_{t, t} (v_t) - \beta_t [v_{t+1}(t+1, t+1)] \\ \text{s.t.} \\ v_t &= v_t - \tau - t \\ v_t &= v_t + t + (t) \\ v_{t+1} &= v_t + t+1 \\ v_{t+1} &= v_t \end{aligned} \tag{60}$$

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

$$v_t(t, t) = v_t(t+1) \tag{61}$$

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

$$\begin{aligned} v_t(t, t) &= v_t [v_{t+1}(t+1, t+1)] \\ \text{s.t.} \\ v_{t+1} &= v_t + t+1 \\ v_{t+1} &= v_t \end{aligned} \tag{62}$$

Then redefine the working agent's problem as

$$\begin{aligned} v_t(t, t) &= \max_{t, t} (v_t) - \beta_t (v_t(t, t)) \\ v_t &= v_t - \tau - t \\ v_t &= v_t + t + (t) \end{aligned} \tag{63}$$

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 EGMⁿ method. Let the agent first decide 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 t and pension assets of t .

$$\begin{aligned}
c_t(t, t) &= \max_t c_t(t, t) \\
\text{s.t.} \\
a_t &= a_t - c_t \\
a_t &= a_t + c_t + (t)
\end{aligned} \tag{64}$$

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

$$\begin{aligned}
c_t(t, t) &= \max_t (c_t) - c_t(t, t) \\
\text{s.t.} \\
a_t &= a_t - c_t
\end{aligned} \tag{65}$$

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

$$c_t(t, t) = \max_t (c_t) - c_t(t - t, t) \tag{66}$$

The first-order condition is

$$v'(c_t) = v'_t(c_{t-t, t}) = v'_t(c_t, t) \tag{67}$$

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

$$c_t(t, t) = v'^{-1}_t(v'_t(c_t, t)) \tag{68}$$

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

$$c_t(t, t) \equiv c_t(c_t, t) \tag{69}$$

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

$$\begin{aligned} t(t, t) &= t(t, t) = ' (t) \\ t(t, t) &= t(t, t) \end{aligned} \quad (70)$$

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

$$t(t, t) = \max_t (t - t, t + t + (t)) \quad (71)$$

The first-order condition is

$$t(t, t)(-1) + t(t, t)(1 + ' (t)) = 0 \quad (72)$$

Rearranging this equation gives

$$' (t) = \frac{t(t, t)}{t(t, t)} - 1 \quad (73)$$

Assuming that $' (\cdot)$ exists and is invertible, we can find

$$t(t, t) = {}'^{-1} \left(\frac{t(t, t)}{t(t, t)} - 1 \right) \quad (74)$$

Using this, we can back out t as

$$t(t, t) = t - t(t, t) - (t(t, t)) \quad (75)$$

and t as

$$t(t, t) = t + t(t, t) \quad (76)$$

In sum, given an exogenous grid (t, t) we obtain the triple $(t(t, t), t(t, t), t(t, t))$, which we can use to create an interpolator for the decision rule t .

To close the solution method, the envelope conditions are

$$\begin{aligned} t(t, t) &= t(t, t) \\ t(t, t) &= t(t, t) \end{aligned} \quad (77)$$

7.5 Is g invertible?

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

$$() = \log(1+) \quad ' () = \frac{1}{1+} \quad {}'^{-1}(y) = /y - 1 \quad (78)$$

7.6 The Post-Decision Value and Marginal Value Functions

$$\begin{aligned}
 {}_t(\cdot) &= {}_t[(t+1, t+1)] \\
 \text{s.t.} \\
 {}_{t+1} &= {}_t + {}_{t+1} \\
 {}_{t+1} &= {}_t
 \end{aligned} \tag{79}$$

and

$$\begin{aligned}
 {}_t(t, t) &= {}_t[{}_{t+1}(t+1, t+1)] \\
 \text{s.t.} \\
 {}_{t+1} &= {}_t + {}_{t+1} \\
 {}_{t+1} &= {}_t
 \end{aligned} \tag{80}$$

and

$$\begin{aligned}
 {}_t(t, t) &= {}_t[{}_{t+1}(t+1, t+1)] \\
 \text{s.t.} \\
 {}_{t+1} &= {}_t + {}_{t+1} \\
 {}_{t+1} &= {}_t
 \end{aligned} \tag{81}$$

7.7 Taste Shocks

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

$${}_t[{}_{t+1}(t+1, t+1, \cdot)] = \sigma \log \left[\sum_{\in \{\cdot, \cdot\}} \exp \left(\frac{{}_{t+1}(t+1, t+1, \cdot)}{\sigma} \right) \right] \tag{82}$$

and

$${}_t(\cdot \mid {}_{t+1}, t+1) = \frac{\exp({}_{t+1}(t+1, t+1, \cdot)/\sigma)}{\sum_{\in \{\cdot, \cdot\}} \exp \left(\frac{{}_{t+1}(t+1, t+1, \cdot)}{\sigma} \right)} \tag{83}$$

the first-order conditions are therefore

$${}_t(t+1, t+1) = \sum_{\in \{\cdot, \cdot\}} {}_t(\cdot \mid {}_{t+1}, t+1) {}_{t+1}(t+1, t+1, \cdot) \tag{84}$$

Acknowledgments

I would like to thank Chris Carroll, Matthew White, and Fedor Iskhakov 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).
- 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](#).

- 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).
- Per Krusell and Anthony A. Smith, Jr. Income and Wealth Heterogeneity in the Macroeconomy. *Journal of Political Economy*, 106(5):867–896, 10 1998. ISSN 0022-3808. doi:[10.1086/250034](https://doi.org/10.1086/250034). URL <http://dx.doi.org/10.1086/250034>.
- Christopher D. Carroll. Solution Methods for Microeconomic Dynamic Stochastic Optimization Problems. <https://github.com/econ-ark/SolvingMicroDSOPs>, 2 2024. URL <https://github.com/econ-ark/SolvingMicroDSOPs>.
- Adrien Auclert, Bence Bardóczy, Matthew Rognlie, and Ludwig Straub. Using the SequenceSpace Jacobian to Solve and Estimate HeterogeneousAgent Models. *Econometrica*, 89(5):2375–2408, 2021. ISSN 0012-9682. doi:[10.3982/ecta17434](https://doi.org/10.3982/ecta17434). URL <http://dx.doi.org/10.3982/ECTA17434>.
- 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.