Projection Methods via Neural Networks for Continuous-Time Models

Maxime Sauzet*

February 11, 2022

Link to latest version

I extend traditional projection methods by using neural networks as a function approximator to solve continuous-time models. The method is well-suited for high-dimensional settings such as those that arise in the presence of many assets, and can capture the strong non-linearities that occur for instance when agents face constraints. It is designed to accommodate multiple unknown functions, e.g. multiple value functions, so that it is set to tackle economies with several agents and incomplete markets.

Keywords: Asset Pricing, Macro-Finance, Heterogeneous Agents, Global Methods, Projection Approaches, Neural Networks, High-Dimensional Settings.
JEL codes: C0, E0, G0.

*Boston University, Questrom School of Business. Email: msauzet@bu.edu. For valuable comments and discussions, I thank Markus Brunnermeier, David Chan, Jérôme Detemple, Victor Duarte, Xiang Fang, Nicolae Gârleanu, Pierre-Olivier Gourinchas, Julien Hugonnier, Patrick Kline, Andrew Lyasoff, Sebastian Merkel, Emi Nakamura, Max Reppen, Jón Steinsson, Andrea Vedolin, Hao Xing, Fernando Zapatero, and seminar participants at the Fall 2019 Berkeley GEMS, and Boston University Questrom. All errors are my own.

1. Introduction

Recently, the interest has grown in the theoretical macroeconomics and economics literature in moving away from solving dynamic stochastic equilibrium models using local methods, such as perturbation approaches, towards solving models globally. Doing so is especially important for instance when (i) non-linearities matter, as is the case in the presence of occasionally-binding constraints, (ii) the sytem can spend a significant amount of time away (and potentially far) from a steady state, (iii) a steady state is difficult (stochastic) or impossible (non-stationary) to define, and (iv) more generally markets are incomplete and the economy is populated by multiple heterogeneous agents. On the other hand, asset pricing and macro-finance models have historically mostly focused on global methods, for a large part because the role of risk is central so that relying on local approximations around a steady state where there is no risk is impracticable, if at all possible. In those contexts, global methods are often coupled with models expressed in continuous time, which make expressing variables of interest – conditional risk premia, their volatilities, portfolios, *etc.* – natural. Recently, in those contexts, the interest has grown towards settings with incomplete markets, and several (potentially many) heterogeneous agents, or assets.

Several global methods have been proposed in the literature such as value or policy function iteration, time iteration, endogenous grid methods, and projection methods for discrete-time models, and standard ordinary differential equatiosn (ODE) methods, value or policy function iteration, finite differences, or projection methods for continuous-time models. Among those, projection methods provide a natural way to deal with models with multiple heterogenous agents and complex market structures, which feature a number of unkwown functions (e.g. several value functions) to solve for that are related by systems of coupled non-linear algebraic and differential equations. This approach, known as (pseudo-) spectral or Galerkin-type methods in the partial differential equations (PDE) literature, typically approximate unknown functions by using Chebyshev polynomials, or splines.

Current methods suffer from a number of difficulties however. First, most are subject to the curse of dimensionality, so that solving models beyond a couple of

state variables rapidly becomes computationally too costly in practice. Second, although projection methods based on Chebyshev polynomials have good properties (convergence *etc.*), they still suffer from at least too additional drawbacks, which are problematic in the contexts mentioned above: (i) capturing strong non-linearities such as the kinks that appear in models with constraints requires a prohibitely high order of approximation[1], (ii) Chebyshev polynomials can provide poor approximation when the domain of the state variables becomes unbounded (or even just wide).

In parallel, neural networks, a tool from machine learning, have rapidly gained ground as a function approximator to deal with some of those issues, e.g. in the mathematical finance, reinforcement learning, applied mathematics literature, as well as more recently in economics. In most of those contexts, they have been combined with iterative methods, with the main goal of solving higher-dimensional models. Such iterative methods however also come with drawbacks, the main one being that not all models are easy, or even possible, to solve in this way. This is for instance the case in games settings, in which the usual convergence results for iterative methods do not necessarily apply making them less straightforward to use, and more generally in equilibrium models with multiple heterogeneous agents and in which the economy does not necessarily admit a representative agent formulation (e.g. typically when markets are incomplete).

In this paper, I alleviate this set of difficulties by combining projection methods with neural networks as function approximator, to solve continuous-time models.

The method has several advantages.

First, because neural networks are designed for such settings, the method is well-suited for high-dimensional contexts with many state variables, as well as many grid points. This is made possible in part by the use of automatic differentation, and stochastic gradient descent, two important elements that allow the training of large models. Automatic differentation in particular makes it possible to obtain derivatives very cheaply, which is especially helpful in a continuous-time setting in which expectations can often be turned into partial derivatives. As shown in Barron (1993), Bach

[1]In fact, even if one could afford an extremely high order of approximation, such non-linearities would still be difficult to capture, e.g. due to oscillations around them like the Gibbs phenomenon. In practice, the required order of approximation are in any case too high to be even feasible with more than one or a couple of state variables. Cf. the main text, and Appendix A for details.

(2017), and as discussed in Fernández-Villaverde et al. (2019), neural networks are also more economical than other approximators for middle and high dimensions in that they achieve a lower order of integrated squared errors than e.g. series approximations (polynomials, splines, trigonometric expansions, *etc.*). Taken together, those suggest that neural networks can relax the curse of dimensionality both theoretically, and in practice.

Second, neural networks can approximate any function, even very non-linear, arbitrary well. This has been shown theoretically and is known as the Universal Approximation Theorem (Hornik et al., 1989), but as importantly, turns out to be verified in practice. For instance, strong non-linearities such as kinks, which appear in economies with constraints, pose no particular problems. Relatedly, neural networks also generalize particularly well, even when the domain of the state variables is unbounded or at least very wide (cf. e.g. Fernández-Villaverde et al., 2019), or between grid points, with much less of the oscillations or wild swings that are not uncommon with polynomials.

Third, because I fit the method within the more usual projection approaches framework, it is naturally able to deal with models whose solutions depend on several (potentially many) unknown functions, which are jointly determined in large systems of coupled algebraic and differential equations, and would not be easy or even possible to solve based on more usual iterative methods. This is for instance the case of equilibrium models with multiple heterogeneous agents, and imperfect risk sharing, in which several value functions (as well as other unknown functions) have to be solved for because a representative agent does not necessarily exist. More generally, studying how to generalize projection (i.e. spectral) methods beyond polynomials and to higher-dimensional cases is also of interest on its own, given that those methods have been growing more popular.

Fourth, the method does not rely on simulating paths of the economy to solve for unknown functions. Instead, grid points are drawn on the whole support of the state variables, which means that draws essentially come for free. While this is computationally more costly than relying solely on the parts of the state space that are most often visited in equilibrium, it allows to trace out the solution to the economy throughout the whole state space. This is particularly useful when one is interested in the behavior of agents and of our variables of interest in rare events that occur only with extremely small probability. Drawing points throughout the domain also

helps neural networks generalize well, although they can do so even with few training points. Finally, this also allows the method to deal with non-stationary models, which do not feature a non-degenerate ergodic distribution but are common in finance e.g. in the applications discussed below.[2]

To demonstrate the power of the method, I apply it to the economy with N investors, M trees, L goods (NML framework) being developped in Sauzet (2021a). This context provides a perfect testing ground given that the generality of preferences, and the potential frictions in financial markets, make the equilibrium such that several unknown functions (value functions, policy functions, equilibrium functions) have to be solved for jointly. I focus on three examples. First, I solve a generalized version of a two-tree model à la Cochrane et al. (2008) and Martin (2013) in which the trees produce two differentiated goods. I show that the method reproduces the solution obtained using Chebyshev polynomials, but with the potential to have many grid points and flexibility. Second, I solve a M-tree version of those models, with one consumption good, CRRA preferences, and $M \geqslant 10$ trees. The solution allows us to study, among other things, the impact of the average output of other trees on the returns correlation between two of them. Interestingly, the output of one tree being extremely small can have the same effect on the returns comovement of two other assets as an overall small decrease in the output of all other trees. This could for instance be used to study the impact of severe negative shocks to one asset, or one part of the market or one sector of the economy, to asset prices, returns correlations, and risk sharing between investors more generally. Lastly, I solve the two-investor two-tree two-good framework of Sauzet (2021b), and show that the method can allow us to study parameter configurations that were difficult or extremely slow to explore with a solution based on Chebyshev polynomials, as well as starting to introduce portfolio constraints in such an economy without apparent difficulty. Those are just a few examples of a broader set of applications that I am currently developing: e.g. investors could have recursive (beyond the two-agent case) or other types of preferences, such as general HARA, the stochastic processes driving the economy could be expanded or production introduced, and the number of trees and agents could be increased further, given that I for the moment only train the neural networks on CPUs and do

[2]The sampling of points could still be updated to feature a larger mass of points in the regions most visited in equilibrium, or even points mostly in this region, depending on the specific model of interest.

not make use of various optimizations and of specialized hardward like GPUs, TPUs. What those examples hightlight however, is that the method appears to be extremely promising to solve models that were computationally simply out of reach before.

Related literature

First, the paper relates to the literature on projection/pseudo-spectral approaches as presented in Judd (1992, 1996, 1998), Trefethen (2000 and many others), Boyd (2001), and as used in asset pricing and macro-finance in Drechsler et al. (2018), Fang (2019), Kargar (2019), Sauzet (2021a,b,c), among others. To this literature, I replace the traditional approximation by Chebyshev or other polynomials by neural network as a function approximator.

Second, the paper contributes more generally to the vast literature on solution methods, in particular in heterogeneous-agent contexts and including large scale models, such as Krusell and Smith (1998), Kubler and Schmedders (2003), Martin (2013), Judd et al. (2014), Maliar and Maliar (2014, 2015), Fernández-Villaverde et al. (2015, 2016), Brunnermeier and Sannikov (2014, 2015, 2016, 2019), Ahn et al. (2018), Achdou et al. (2021), Fernández-Villaverde and Levintal (2018), Kaplan et al. (2018), Levintal (2018), Hansen et al. (2018). To this literature, I bring the high-dimensional capacity of neural networks, as well as their flexibility in approximating even very non-linear functions, while fitting it in the projection methods paradigm.

Finally, the paper relates to the rapidly growing literature on using machine learning (especially neural networks) to solve economic and financial models. Those include, among many others, Duarte (2019), Fernández-Villaverde et al. (2019), Maliar et al. (2019), Ebrahimi Kahou et al. (2021), as well as the rapidly growing literature in applied mathematics on using neural networks to solve partial differential equations (PDEs) or for reinforcement learning, e.g. Han and E (2016), Han et al. (2018), among many others. I use the advantages of neural networks emphasized in this literature, but I cast the method within the projection appraoches paradigm. This is more standard and used in the economics and finance literature, and is also naturally able to deal with models whose solutions depend on several (potentially many) unknown functions, which are jointly determined in large systems of coupled

algebraic and differential equations, and would not be easy or even possible to solve based on the methods of that literature, many of them being iterative. Compared to some of those contributions e.g. Maliar et al. (2019), Ebrahimi Kahou et al. (2021) that focus on discrete-time models, I focus on economies set up in continuous time. With respect to the latter paper, I also do not rely on the symmetry of the economy and do not focus on specific parts of the state space. While computationally more costly so that I am likely to reach lower dimensionality, this allows me to trace out the solution to the economy throughout the whole state space. This is particularly useful when one is interested in the behavior of agents and of our variables of interest in rare events that occur only with extremely small probability.

From a broad perspective, the literature on such methods is growing extremely rapidly, and my paper is related to potentially many more contributions (cf. the introduction and the rest of the paper for additional discussions). An important distinction to make is that I focus on economies with a finite number of agents (or types), instead of a continuum. While this number can be large, agents remain non-infinitesimal and potentially heterogeneous in asymmetric ways, contrary to methods relying on the symmetry of the problem to allow for yet higher dimensionality, or relying on a distribution of atomistic agents such as in mean field games (Kaplan et al., 2018; Ahn et al., 2018; Achdou et al., 2021). In other words, my focus is on using neural networks to solve the indivdual problem of a finite number of agents, as in Duarte (2019), although fitting it in a projection framework allows to deal with potentially many unknown functions such as the multiple and non-perfectly correlated value functions that arise under imperfectly integrated markets. Neural networks could also be used on top as an additional aggregation step if the agents were to become infinitesimal, e.g. as in Fernández-Villaverde et al. (2019).

The paper is organized as follows. Section 2 presents the method and how it compares to traditional projection approaches based on Chebyshev polynomials. Section 3 discusses applications, both currently implemented and possible. Section 4 concludes. Additional results and details are provided in Appendix.

2. Method

This section presents projection methods from a general perspective (2.1), their traditional implemention and related difficulties (2.2), the projection methods via neural networks that I propose in this paper (2.3), and briefly discusses the details of its implementation (2.4). The exposition in Parts 2.1 and 2.2 partly draws from Judd (1992, 1998), Fernández-Villaverde and Christiano (2011), and Kargar (2019).

Throughout, I focus on models that can be boiled down to a system of equations

$$\mathcal{H}(\mathbf{G}(\mathbf{X})) = \mathbf{0}_{\mathbb{R}^Q} \tag{1}$$

where $\mathbf{X} \in \Omega \subseteq \mathbb{R}^P$ are the relevant (multi-dimensional) state variables, $g \in \mathcal{G}$ are M unknown functions with $g : \Omega \to \mathcal{D}^g \subseteq \mathbb{R}$, $\mathbf{G} : \Omega \to \mathcal{D} \subseteq \mathbb{R}^M$ is the vector function combining all unknown gs, $\mathcal{H} : \mathcal{B}_1 \to \mathbb{R}^Q$ is the operator encoding the system of Q equations, and $\mathcal{B}_1$ is the space of functions from Ω to $\mathcal{D}$.

Specifically, I focus on continuous-time models, whose solutions can be expressed as stationary recursive Markovian equilibria. Casting models in continuous time allows to use the power of automatic differentiation to obtain derivatives cheaply instead of having to deal with expectations, which themselves involve (potentially high-dimensional) integrals.[3] The method could also be extended to handle cases with explicit time-dependence $\mathbf{G}(t, X_t)$. Overall, the type of settings that the reader should have in mind is that of stochastic optimal control equilibrium models with multiple (potentially many but non-atomistic) agents who have an infinite time horizon and general preferences. In those contexts, the unknow functions gs can be value functions, policy functions, or other equilibrium functions, and $\mathcal{H}$ is typically a non-linear (potentially large) system of coupled algebraic and differential equations. Additional equations that are not strictly speaking equilibrium equations can also be added to $\mathcal{H}$, e.g. to help the convergence, or impose constraints.

Because most models of interest cannot be solved in closed form, the name of the

[3] For recent related work in a discrete-time context, which also makes use of the symmetry inherent in certain multiple-agent contexts, cf. Ebrahimi Kahou et al. (2021).

game for any numerical method is to solve an approximated version of system (1)

$$\hat{\mathcal{H}}(\hat{\mathbf{G}}(\mathbf{X})) \approx \mathbf{0}_{\mathbb{R}^Q} \tag{2}$$

One can think of approximating $\mathcal{H}$, i.e. the system of equations itself, and this approach is undertaken e.g. in Campbell and Shiller (1988), Campbell and Viceira (1999), Chacko and Viceira (2005), and Tsai and Wachter (2018), among others. In this paper however, I am interested in the potential impact of strong non-linearities, including in the equations themselves, so the system itself $\mathcal{H}$ is kept as is and is not approximated.

Instead, I focus on approximating the unknown functions $g \in \mathcal{G}$, i.e. on finding a $\hat{\mathbf{G}}$ that is a suitable approximation for $\mathbf{G}$. Again, several approaches have been proposed to do so in the literature, ranging from perturbation approaches in which unknown functions are approximated in local neighborhoods of a specific point (typically the steady-state in the economy, in which there is no risk), other Taylor approximations (e.g. Levintal, 2018, Fernández-Villaverde and Levintal, 2018), to projection approaches.[4]

2.1. Projection methods

Projection methods can be seen as consisting of two main steps.

Step 1: Pick a basis In this step, a basis $\{\Psi(\mathbf{X})\}$ for the space of functions $\mathcal{B}_1$ is chosen. Each unknown function g is approximated using this basis, and those functions depend only on the state variables $\mathbf{X}$. In turn, one must be able to express all other variables of interest in the model solely as a function of those gs and the state variables $\mathbf{X}$. In practice, the approximation of $\mathbf{G}$ will depend on parameters $\theta \in \Theta$, so that I sometimes write $\hat{\mathbf{G}}(\mathbf{X}) = \hat{\mathbf{G}}(\mathbf{X}; \theta)$ when making this dependence explicit helps the intuition.

[4] There exist of course potentially many other methods. This list is therefore not exhaustive, although it mentions the most standard ones.

Step 2: Define the residual In order to approximately solve Equation (2), define the residual[5]

$$\mathcal{R}(\mathbf{x};\theta) \equiv \hat{\mathcal{H}}(\hat{\mathbf{G}}(\mathbf{x};\theta)) \tag{3}$$

The goal is then to pick $\theta \in \Theta$ so that $\mathcal{R}(\mathbf{x};\theta) \approx \mathbf{0}_{\mathbb{R}^Q}$, i.e.

$$\hat{\theta} = \arg\min_{\theta\in\Theta} \rho(\mathcal{R}(\mathbf{x};\theta), \mathbf{0}_{\mathbb{R}^Q}) \tag{4}$$

where $\rho(\cdot,\cdot)$ is a measure of distance to be chosen as well.

2.2. Traditional projection methods

There are many possibilites to choose from for both steps.

In traditional projection methods, the basis $\{\Psi(\mathbf{X})\}$ for the space of functions is typically chosen to be some polynomials, with Chebyshev polynomials being the most common. When $\mathbf{X} \in \Omega \subseteq \mathbb{R}^P$ is multi-dimensional, i.e. $P > 1$, those polynomials must be combined to provide an appropriate basis. A common way to do so is to use the tensor-product of Chebyshev polynomials in each dimension. To give a concrete example, say that $\mathbf{X} \equiv (x,y)' \in [0,1]^2$. A basis of order $\mathcal{O}$ based on the tensor-product of Chebyshev polynomials is

$$\Psi_{ij}^{(\mathcal{O})}(x,y) = T_i\left(\omega(x)\right) T_j\left(\omega(y)\right) \tag{5}$$

where T are Chebyshev polynomials, and $\omega(u) = 2(u-1)$ transforms x, y from $[0,1]$ to $[-1,1]$ on which those polynomials are defined. In other words, each $g : \Omega^x \times \Omega^y \to D^g \subseteq \mathbb{R}$ in $\mathcal{G}$ is approximated as

$$\hat{g}(x,y) = \sum_{i=0}^{\mathcal{O}} \sum_{j=0}^{\mathcal{O}} \theta_{ij}^{g,(\mathcal{O})} \Psi_{ij}^{(\mathcal{O})}(x,y) \tag{6}$$

This choice of basis has many nice properties, e.g. the polynomials are orthogonal, and there exist various convergence results when using them to solve differential

[5]Recall that because $\mathcal{H}$ is not approximated, $\hat{\mathcal{H}} \equiv \mathcal{H}$.

equations (cf. Judd, 1992, 1996, 1998, Trefethen, 2000 and many others, Boyd, 2001, Fernández-Villaverde and Christiano, 2011, among others), and they also work well in practice. Other choices are also possible and can be useful depending on the use case such as Legendre polynomials, or splines.[6] In addition, polynomials can be combined using more sophisticated algorithms to reduce the number of parameters to estimate, e.g. using complete polynomials or Smolyak's algorithm.

For step 2, several measures of distance $\rho(\cdot, \cdot)$ have also been used in the literature such as weighted least squares, Garlerkin methods, or the method of moments (cf. Fernández-Villaverde and Christiano, 2011, and Kargar, 2019, for an overview). Among those, a popular choice is to take the Dirac delta as weight function so that the residual is set to 0 at specific points of the state space. Those points are usually taken to be the zeros of the basis, and the resulting method, known as orthogonal collocation (or pseudo-spectral methods) has proven to be useful and popular in practice, and as accurate as more complex choices of ρ.

Despite all those possibilities however, and as mentioned in the introduction, traditional projection methods suffer from a number of difficulties.

First, like many methods, traditional projection approaches are subject to the curse of dimensionality, so that solving models beyond a couple of state variables becomes computationally too costly. This is true in particular when combining polynomials using tensor products, but even if complete polynomials or using Smolyak's algorithm can help, polynomials still become rapidly computationally infeasible in practice (cf. e.g. Maliar et al., 2019, for the number of parameters to be estimated as the order and dimension increase).

Second, capturing strong non-linearities such as the kinks that appear in models with constraints requires a prohibitely high order of approximation. For instance, Appendix A shows that an order as high as $N = 100$ is needed to capture the kinks in one single simple function (cf. in particular Figure A.1). In practice, such a higher order is too high to be feasible with more than one or a couple of state variables. In fact, as is shown in the zoomed Figure A.2 in the same appendix, even if one could afford such a high order of approximation, this type of non-linearities would still be difficult to capture, e.g. due to the oscillations like the well-known Gibbs

[6] Even finite elements can be seen as a local basis in this framework.

phenomenon that appear around the kinks when approximating them with polynomials. This is particularly problematic because many of the important applications of such global methods might be interesting for the very reason that they include those non-linearities. For instance, this is the case when they stem from constraints to which agents are subject, and that are economically meaningful, like in the recent intermediary asset pricing literature.

Third, Chebyshev and other polynomials also turn out to be quite poor at approximating functions when the support of the underlying state variables becomes unbounded or even just wide, as can be the case depending on the economic model of interest.

Taken together, those are some of the main reasons why using traditional projection methods becomes intractable in practice when moving towards more ambitious equilibrium models with multiple (possibly many) state variables, and strong non-linearities.

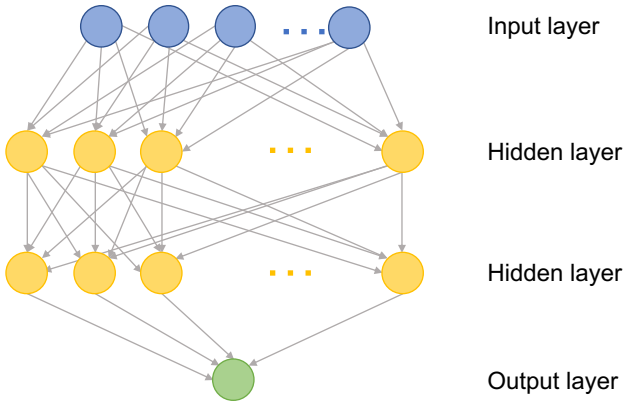
2.3. Projection methods via neural networks

To alleviate those difficutlies, the approach that I propose in this paper combines projection methods, with neural network as a function approximator. The goal is to leverage the tremendous recent progress brought upon by the latter in a variety of fields such as machine learning, applied mathematics, and mathematical finance, and neural networks turn out to address each and every one of the difficulties mentioned above.

In practice, for step 1, I approximate each function $g : \Omega^x \times \Omega^y \to D^g \subseteq \mathbb{R}$ in $\mathcal{G}$ by a neural network parametrized by $\theta \in \Theta$, i.e. $\hat{g}(\mathbf{X}) = \hat{g}(\mathbf{X}; \theta)$. Each neural network has one or several hidden layers, composed of neurons $\phi(b_k + W_k' s_k)$ with activation functions $\phi(\cdot)$. A representative architecture is presented in Figure 1, and I experiment with a number of designs (number of neurons and layers), and activation functions ϕ (e.g. sigmoid, i.e. logistic, SoftPlus, *etc.*). An important aspect is that the common REctified Linear Units (ReLU) activation function involves a kink so that its second-order derivative does not exist, and therefore cannot be used because it would not be able to approximate second-order differential equations that are common for our

models of interest. For additional details on neural networks especially in economic and financial settings, the reader is referred to Duarte (2019), Fernández-Villaverde et al. (2019), Maliar et al. (2019), and Ebrahimi Kahou et al. (2021). For all intents and purposes in this paper, they can be seen as a function approximator, albeit with particularly interesting properties. Introducing them in the system of equations $\mathcal{H}$ yields a combined network that has to be trained.

Figure 1: Representative neural network infrastructure $\hat{g}(\mathbf{X};\theta)$



For step 2, the goal is to choose $\theta \in \Theta$ so that $\mathcal{R}(\mathbf{x};\theta) \approx \mathbf{0}_{\mathbb{R}^Q}$, i.e. $\mathcal{H}^q(\hat{\mathbf{G}}(\mathbf{X};\theta)) \approx 0$ for any $q \in \{1,\dots,Q\}$, where $\mathcal{H}^q(\cdot)$ represents the qth equation of system $\mathcal{H}$. The distance ρ to minimize is therefore taken to be

$$\sum_{q=1}^{Q} \lambda_q \rho_q(\mathcal{H}^q(\hat{\mathbf{G}}(\mathbf{X};\theta)), 0) \tag{7}$$

In practice, I set equal weights to each individual equation of the system $\lambda_q \equiv 1/Q$. The weights could be adjusted however to put a particular emphasis on certain equations, either because they matter more economically, or because their minimization turns out to be more difficult. The equation-level distance or loss measure ρ_q is mostly taken to be the standard Mean Squared Error (MSE), or the Mean Absolute Error (MAE), but I also experiment with other losses such as the Mean Absolute Percent Deviation Error (MAPE), *etc.* In some instances, alternating between different losses at various stages of the training is also helpful to get closer to a more global minimum.

Finally, while traditional projections often rely on specific grid points such as the zeros of the corresponding polynomials, I draw the state variables $\mathbf{X}$ used to train the network uniformally over their whole domain $\Omega \subseteq \mathbb{R}^P$. This has the advantage of being particularly cheap so that training points virtually come for free. Other possibilities could also be explored, for instance by drawing points from a beta or a normal distribution if points at the boundary or at the center of the state space are important, respectively. The sampling of points could also be updated to feature a larger mass of points in the regions most visited in equilibrium, or even points mostly in this region, depending on the specific model and quantity of interest.

The method has several advantages, which turns out to address specifically most of the difficulties of traditional methods mentioned previously.

First, because neural networks are designed for such settings, the method is well-suited for high-dimensional contexts with many state variables, as well as many grid points. This is made possible in part by the use of automatic differentation, and stochastic gradient descent, two important elements that allow the training of large models. Automatic differentation in particular makes it possible to obtain derivatives very cheaply, which is especially helpful in a continuous-time setting in which expectations can often be turned into partial derivatives. As shown in Barron (1993), Bach (2017), and as discussed in Fernández-Villaverde et al. (2019), neural networks are also more economical than other approximators for middle and high dimensions in that they achieve a lower order of integrated squared errors than e.g. series approximations (polynomials, splines, trigonometric expansions, *etc.*). Taken together, those suggest that neural networks can relax the curse of dimensionality both theoretically, and in practice.

Second, neural networks can approximate any function, even very non-linear, arbitrary well. This has been shown theoretically and is known as the Universal Approximation Theorem (Hornik et al., 1989), but as importantly, turns out to be verified in practice. For instance, Appendix A shows that strong non-linearities such as kinks, which appear in economies with constraints and were problematic for Chebyshev polynomials, pose no particular problems. Relatedly, neural networks also generalize particularly well, even when the domain of the state variables is unbounded or at least very wide (cf. e.g. Fernández-Villaverde et al., 2019), or between grid points, with much less of the oscillations or wild swings that are not uncommon with polynomials.

Third, because I fit the method within the more usual projection approaches framework, it is naturally able to deal with models whose solutions depend on several (potentially many) unknown functions, which are jointly determined in large systems of coupled algebraic and differential equations, and would not be easy or even possible to solve based on more usual iterative methods. This is for instance the case of equilibrium models with multiple heterogeneous agents, and imperfect risk sharing, in which several value functions (as well as other unknown functions) have to be solved for because a representative agent does not necessarily exist. More generally, studying how to generalize projection (i.e. spectral) methods beyond polynomials and to higher-dimensional cases is also of interest on its own, given that those methods have been growing more popular.

Fourth, as mentioned above, the method does not rely on simulating paths of the economy to solve for unknown functions. Instead, grid points are drawn on the whole support of the state variables, which means that draws essentially come for free. While this is computationally more costly than relying solely on the parts of the state space that are most often visited in equilibrium, it allows to trace out the solution to the economy throughout the whole state space. This is particularly useful when one is interested in the behavior of agents and of our variables of interest in rare events that occur only with extremely small probability. Drawing points throughout the domain also help neural networks generalize well, although they can do so even with few training points. Finally, this also allows the method to deal with non-stationary models, which do not feature a non-degenerate ergodic distribution but are common in finance e.g. in the applications below. Again, the sampling of points could still be updated to feature a larger mass of points in the regions most visited in equilibrium, or even points mostly in this region, depending on the specific model of interest.

2.4. Implementation

Beyond their clear theoretical and technical appeal mentioned above, neural networks also have the advantages that they are part of a vast and rapidly expanding ecosystem of packages developed by large numbers of machine learning specialists, computer scientists, and software engineers at tech companies like Google, Facebook, and others. This means that little time has to be devoted to implementing them and their train-

ing algorithms, and that the focus can be put on adapting them to the projection framework proposed in this paper and our economic models of interest.

In practice, I base the current implementation in this paper on Tensor Flow v2.0, together with Keras as an additional higher-level layer. For prototyping and debugging, I focus on so-called eager execution in which the graph corresponding to the networks are built dynamically. I move to a static graph that is only built once when the focus is on performance. For the moment, I only train the networks on CPUs (either local machines or servers), and do not make use of various optimizations and of specialized hardward like GPUs, TPUs. This means that those results are only lower bounds to the ultimate possibility of the method that I propose. Despite this fact, the applications below already highlights that the method appears to be extremely promising, as it allows to solve models that were computationally simply out of reach before it. Adding various optimizaitons, and the massive parallelization and capabilities of GPUs and TPUSs, could open the door for even much larger-scale settings. In addition, other packages such as PyTorch, JAX, or dedicated languages like OCaml, among others, could raise those possibilities further. I am exploring those in ongoing investigations.

3. Applications

To demonstrate the power of the method, I apply it to the economy with N investors, M trees, L goods (NML framework) being developped in Sauzet (2021a). This context provides a perfect testing ground given that the generality of preferences, and the potential frictions in financial markets, make the equilibrium such that several unknown functions (value functions, policy functions, equilibrium functions) have to be solved for jointly. I focus on three subcases as examples.

In Part 3.1, I solve a generalized version of a two-tree model à la Cochrane et al. (2008) and Martin (2013) in which the trees produce two differentiated goods, i.e. a one-investor two-tree two-good model (122). In Part 3.2, I solve a M-tree version of those models, with one consumption good, CRRA preferences, and $M \geqslant 10$ trees.[7]

[7]For the first two subcases, the model could be solved using other approaches, e.g. based on Monte Carlo or iterative methods, at least when M is small. Comparing those solutions more formally to the one obtained using the method in this paper could be an interesting check. For other cases,

Lastly, in Part 3.3, I solve the two-investor, two-tree, two-good framework of Sauzet (2021b). Those are a few examples of a broader set of applications that I am currently developing.[8]

To fix ideas, I now present equations for the case of one investor and multiple trees. The results for specific subcases are presented in Parts 3.1 and 3.2, while the two-investor case is discussed in Part 3.3. The general NML framework is being developed in Sauzet (2021a). To ease exposition, I focus mostly on the one-good case. The multiple-good case is similar but involves relative goods prices $p_{j,t}$, as well as their (geometric) drifts $\mu_{p_j,t}$ and diffusions $\sigma_{p_j,t}$.

The setup is that of a pure exchange economy. The supply of goods is produced by M trees whose outputs follow

$$\frac{dY_{j,t}}{Y_{j,t}} = \mu_{Y_j} dt + \sigma_{Y_j}^T d\vec{Z}_t \quad \text{with} \quad j \in \{0, 1, \dots, M-1\} \tag{8}$$

where $\vec{Z}_t$ is a M-dimensional standard Brownian motion. Its components are assumed to be uncorrelated for simplicity, but this is not required.

The representative investor has recursive preferences à la Duffie and Epstein (1992)

$$V_t = \max_{\{C_u, \vec{w}_u\}_{u=t}^{\infty}} \mathbb{E}_t \left[\int_t^{\infty} f\left(C_u, V_u\right) du \right] \tag{9}$$

$$f(C, V) \equiv \left(\frac{1-\gamma}{1-1/\psi} \right) V \left[\left(\frac{C}{[(1-\gamma)V]^{1/(1-\gamma)}} \right)^{1-1/\psi} - \rho \right] \tag{10}$$

where γ is the coefficient of relative risk aversion, ψ is the elasticity of intertemporal substitution (IES), and ρ is the discount rate. Again, for expositional simplicity, I

which are of more interest, this is no longer necessarily possible and only the method presented in this paper can always be used, especially when markets become imperfectly integrated.

[8] E.g. investors can have recursive (beyond the two-agent case) or other types of preferences, such as general HARA, the stochastic processes driving the economy could be expanded or production introduced, and the number of trees and agents could be increased further, given that I for the moment only train the neural networks on CPUs and do not make use of various optimizations and of specialized hardward like GPUs, TPUs. What those examples hightlight however, is that the method appears to be extremely promising to solve models that were computationally simply out of reach before.

mostly focus below on the CRRA case in which $\psi = 1/\gamma$, but the method has no difficulty in handling the general case (for instance, it is used in Part 3.3). Because this is an endowment economy, consumption is simply

$$C_t = \sum_{j=0}^{M-1} Y_{j,t} \quad \text{or} \quad C_t = \left(\sum_{j=0}^{M-1} \alpha_j^{\frac{1}{\theta}} Y_{j,t}^{\frac{\theta-1}{\theta}} \right)^{\frac{\theta}{\theta-1}} \tag{11}$$

depending on whether there is $L = 1$ or $L = M$ good(s). $\{\alpha_j\}$s sum to 1 and capture potential bias in consumption towards one of the goods, while θ is the elasticity of substitution across goods. In other words, the $L = 1$-good case corresponds to $\theta \to \infty$. This can also be generalized for $L < M$.

The trees are traded as equity assets in unit supply, together with a riskless bond in zero net supply. In this context, one can show that for any asset $j \in \{0, 1, \ldots, M-1\}$, the risk premium satisfies

$$\mu_{R_j,t} - r_t = \kappa_t^T \sigma_{R_j,t} \tag{12}$$

where $\mu_{R_j,t}$ and $\sigma_{R_j,t}$ are the drift and diffusion of the returns on asset j, r_t is the locally riskless interest rate on the bond, and κ_t is the (M-dimensional) price of risk.

r_t and κ_t can be obtained from the stochastic discount factor ξ_t of the investor

$$\frac{d\xi_t}{\xi_t} \equiv -r_t dt - \kappa_t^T d\vec{Z}_t \tag{13}$$

where

$$\begin{cases} \xi_t = e^{-\rho t} C_t^{-\gamma} & \text{(CRRA)} \\ \xi_t = \exp\{\int_0^t f_V(C_u, V_u) du\} C_t^{-\gamma} J_t^{\frac{1-\gamma}{1-\psi}+\gamma} & \text{(Recursive)} \end{cases} \tag{14}$$

Using the expression for consumption (11) and Itô's Lemma, r_t and κ_t can be expressed in closed form under CRRA preferences, and as a function of J_t the marginal value of wealth of the investor (to be solved for) under recursive preferences.

The other two elements in Equation (12) are $\mu_{R_j,t}$ and $\sigma_{R_j,t}$, the drift and diffusion of the returns on asset j. Those can be obtained from the definition of returns.

Denoting $F_{j,t} \equiv Y_{j,t}/Q_{j,t}$ the dividend yield on asset j, they read[9]

$$dR_{j,t} = \frac{dQ_{j,t}}{Q_{j,t}} + F_{j,t}dt = \frac{d\left(Y_{j,t}/F_{j,t}\right)}{Y_{j,t}/F_{j,t}} + F_{j,t}dt \equiv \mu_{R_j,t}dt + \sigma_{R_j,t}^T d\vec{Z}_t \tag{15}$$

Using Itô's Lemma, for any $j \in \{0, 1, \ldots, M-1\}$

$$\mu_{R_j,t} = F_{j,t} + \mu_{Y_j} - \mu_{F_j,t} + \sigma_{F_j,t}^T \left(\sigma_{F_j,t} - \sigma_{Y_j}\right) \tag{16}$$

$$\sigma_{R_j,t} = \sigma_{Y_j} - \sigma_{F_j,t} \tag{17}$$

where

$$\mu_{F_j,t}F_{J,t} = \sum_{k=0}^{M-1} \frac{\partial F_{j,t}}{\partial Y_{k,t}} \mu_{Y_k} Y_{k,t} + \frac{1}{2} \sum_{k=0}^{M-1} \sum_{l=0}^{M-1} \frac{\partial^2 F_{j,t}}{\partial Y_{l,t} \partial Y_{k,t}} \sigma_{Y_k}^T \sigma_{Y_l} Y_{k,t} Y_{l,t} \tag{18}$$

$$\sigma_{F_j,t}F_{J,t} = \sum_{k=0}^{M-1} \frac{\partial F_{j,t}}{\partial Y_{k,t}} \sigma_{Y_k} Y_{k,t} \tag{19}$$

Plugging Equations (13)-(19) in the risk premium expression (12) yields a system of M decoupled PDEs in the M dividend yields $F_{j,t}$ under CRRA preferences, or a system of $M+1$ coupled PDEs in the M dividend yields $F_{j,t}$ and the marginal value of wealth J_t under recursive preferences.[10]

To fit the equations above in the method of this paper, presented in Section 2, the steps are as follows

1. Approximate each unknown function $(\{F_{j,t}\},\ J_t)$, which is a function of M variables $(\{Y_{k,t}\})$, by a neural network with M inputs.
2. Use automatic differentiation to get the first- and second-order partial derivatives of $F_{j,t}$ and J_t with respect to $\{Y_{k,t}\}$ for any j, k.
3. Combine those according to $\mathcal{H}$, the system of equations.
4. Train the unknown networks (i.e. find parameters) so that $\mathcal{H}(\hat{\boldsymbol{G}}(\boldsymbol{X};\theta)) \approx \mathbf{0}_{\mathbb{R}^Q}$[11].

[9] With $L > 1$ goods, all those expressions also involve the price of goods $p_{j,t}$, which are functions of the relative supply of their respective trees, as well as their (geometric) drifts and diffusion, $\mu_{p_j,t}, \sigma_{p_j,t}$.

[10] The system under recursive preferences also involves the Hamilton-Jacobi-Equation for the investor, shown in Appendix B.1.

[11] Here: $\boldsymbol{G} = (F_{0,t}, F_{1,t}, \ldots, F_{M-1,t}, J_t)^T$, $\boldsymbol{X} = (Y_{0,t}, Y_{1,t}, \ldots, Y_{M-1,t})^T$.

In practice, I focus on the CRRA case in Parts 3.1 and 3.2, while I use recursive preferences in Part 3.3.

3.1. One investor, two trees, two goods (122)

I first solve a generalized version of a two-tree model à la Cochrane et al. (2008) and Martin (2013) in which the trees produce two differentiated goods, i.e. a one-investor two-tree two-good model (122).

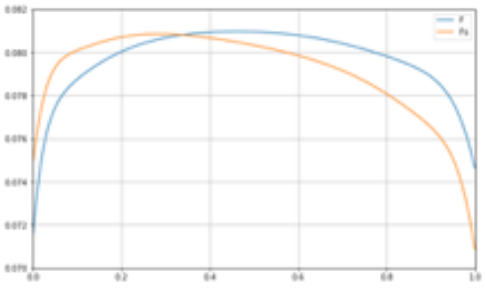
To ease interpretation, one can use the relative supply of good 0 as a single state variable, instead of keeping track of both $Y_{0,t}$ and $Y_{1,t}$. In other words, define

$$X_t = y_t \equiv \frac{Y_t^0}{Y_t^0 + Y_t^1} \in [0,1] \; \forall t \geqslant 0 \tag{20}$$

Figure 2 presents $F_{0,t}, F_{1,t}$, the dividend yields for the two assets as a function of X_t, obtained using projection methods via neural networks (Panel (a)), and traditional projection methods, which are still feasible in this case (Panel (b)).[12] The calibration is as follows: bias in consumption $\alpha = 0.63$, weight in numéraire basket $a = 1/2$[13], risk aversion $\gamma = 5$, elasticitiy of intertemporal substitution $\psi = 1/\gamma$ (CRRA), discount rate $\rho = 0.01$, elasticity of substitution across goods $\theta = 2$.

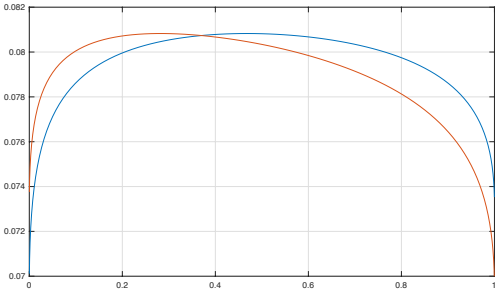
Figure 2: Dividend yields $F = F_{0,t}, Fs = F_{1,t}$

Panel (a): projection via neural networks



1,000 epochs, β grid of 10,000pts

Panel (b): traditional projection methods



Chebyshev at order = 100

[12] The latter is obtained using the *chebfun* package (Trefethen, 2000).

[13] The numéraire with weight a is set so that $\left[ap_{0,t}^{1-\theta} + (1-a)p_{1,t}^{1-\theta}\right]^{1/(1-\theta)} = 1$.

Comparing the two panels confirms that the projection methods via neural networks proposed in this paper delivers an extremely accurate solution, comparable to that obtained with traditional methods with a very high order of approximation $\mathcal{O} = 100$. However, even in in this setting in which a solution using traditional Chebyshev polynomials is still feasible, and in fact faster (10-30 seconds vs. a couple of minutes), the projection methods via neural networks allow to use many more grid points (10,000 or more without difficulty as opposed to 100). This hightlights that the curvature of the dividends yields might in fact be changing slightly more suddenly in some part of the state space than implied by Chebyshev polynomials (e.g. around $X_t = 0.05$ and $X_t = 0.95$). Second, an order of $\mathcal{O} = 100$ as used with traditional methods in Panel (b) would already be prohibitely high when moving to even two state variables, but potentially barely high enough to capture strong non-linearities with a high degree of precision (cf. Appendix A). In other words, even in this simple example, the method proposed in this paper already allows more grid points, i.e. potentially more accuracy, and more flexibility. This will be even more the case as M increases, given that traditional methods are simply not computational feasible in those settings.

3.2. M trees (1M1)

I now solve a M-tree version of the NML model, with one consumption good, CRRA preferences, and $M \geqslant 10$ trees, i.e. a 1M1 model with $M \geqslant 10$. Instead of recasting the model as a function of relative state variables in $[0, 1]$ like in the previous part, I keep $X_t \equiv (Y_{0,t}, Y_{1,t}, \ldots, Y_{M-1,t})'$, where $\{Y_{j,t}\}$ are the outputs of the trees. Those are in $[0, \infty)$ so that I solve the model on a $[0, 100]^M$ grid. This allows to demonstrate the fact that the method has no particular difficulty solving models who state variables are defined on wide domains, i.e. that it generalizes well, unlike Chebyshev polynomials.

After a training of a couple of hours[14] on a grid with 10,000 points drawn uniformally, with batch sizes of 2,500, the methods achieves a MSE of around 10^{-10} on

[14]The first epoch takes much longer on average, while additional epoches are extremely fast. The model can therefore be trained further with proportionally less time. In addition, this is with little particular optimization.

average. (This is low for neural networks.) The MSE loss is even lower at 10^{-15} on average when solving the model on a 100-point grid. As an illustration, Figure 3 shows the 10 residuals as a function of $Y_{0,t}$ and $Y_{1,t}$, holding $Y_{j,t} = 50$ for $j \in \{2, \ldots, 10-1\}$, for a risk aversion of $\gamma = 5$, an elasticitiy of intertemporal substitution of $\psi = 1/\gamma$ (CRRA), and a discount rate of $\rho = 0.01$. In short: the residuals are small.

Figure 3: 1M1: residuals as a function of $Y_{0,t}, Y_{1,t}$ ($Y_{j,t} = 50$ for $j \in \{2, ..., 9\}$)

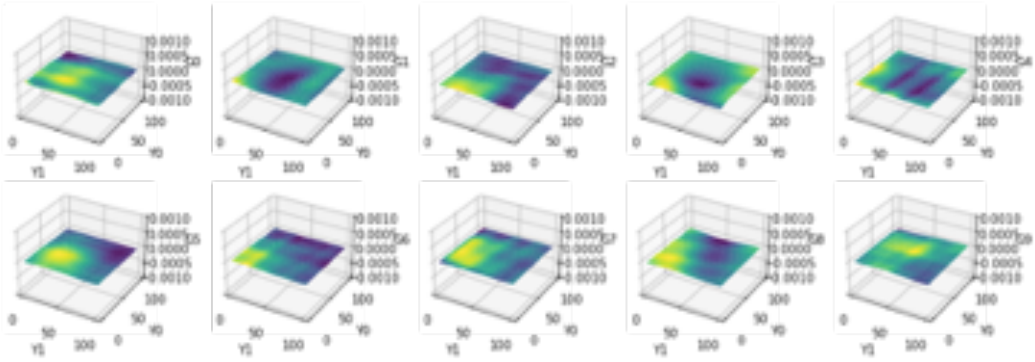


Figure 4: 1M1: div. yields $F_{j,t}$ as a function of $Y_{0,t}, Y_{1,t}$ ($Y_{j,t} = 50$ for $j \in \{2, ..., 9\}$)

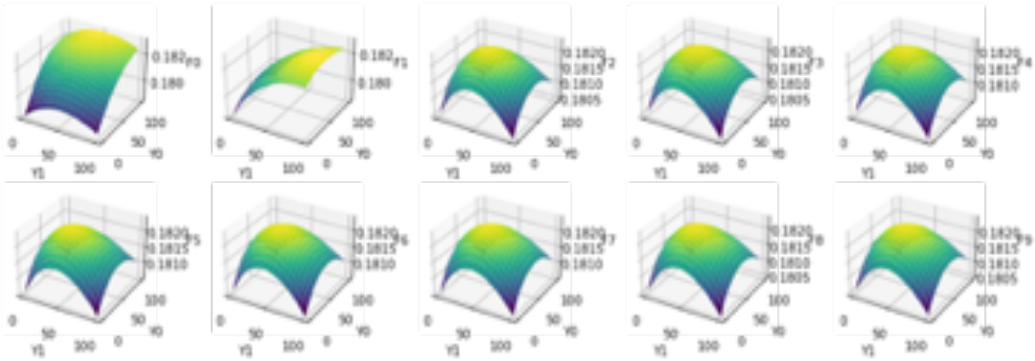
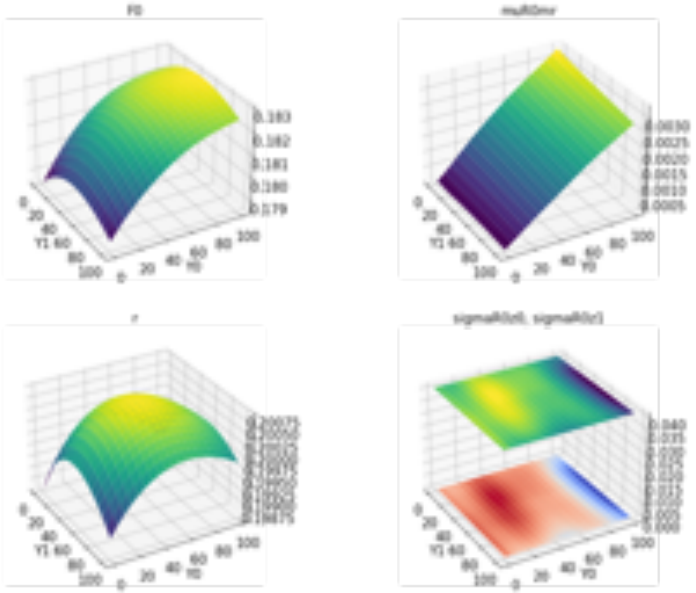


Figure 3 also shows the resulting dividend yields $\{F_{j,t}\}$ for all M trees, and Figure 5 shows the dividend yield ($F_{0,t}$), risk premium ($\mu_{R_0,t} - r_t$), and first two diffusion terms ($\sigma_{R_0 z_0,t}, \sigma_{R_0 z_1,t}$) for asset 0, as well as the locally riskless interest rate (r_t). As expected, the risk premium and dividend yields increase with the supply of good $Y_{0,t}$, consistent with asset 0 becoming riskier. This is because as the supply of good 0 becomes more dominant in total output/consumption, the equity price of asset 0 $Q_{0,t}$ becomes more dominant in the wealth of the representative investor W_t, which makes

it more difficult for her to diversify the risk of asset 0 away. Because the volatility of consumption is ultimately lowest in the middle of the state space (around $Y_{j,t} = 50$ for all $j \in \{0, 1, \ldots, M-1\}$), precautionary savings motive are lowest so that r_t reaches its maximum. Lastly, as expected, the returns on asset 0 load mostly on the first Brownian motion so that $\sigma_{R_0 z_0,t} \gg \sigma_{R_0 z_j,t}$ for all $j \neq 0$, even though those $\sigma_{R_0 z_j,t}$ are non-zero. All those patterns of course vary both qualitatively and quantitatively with the calibration, including the correlation of the different Brownian motions.

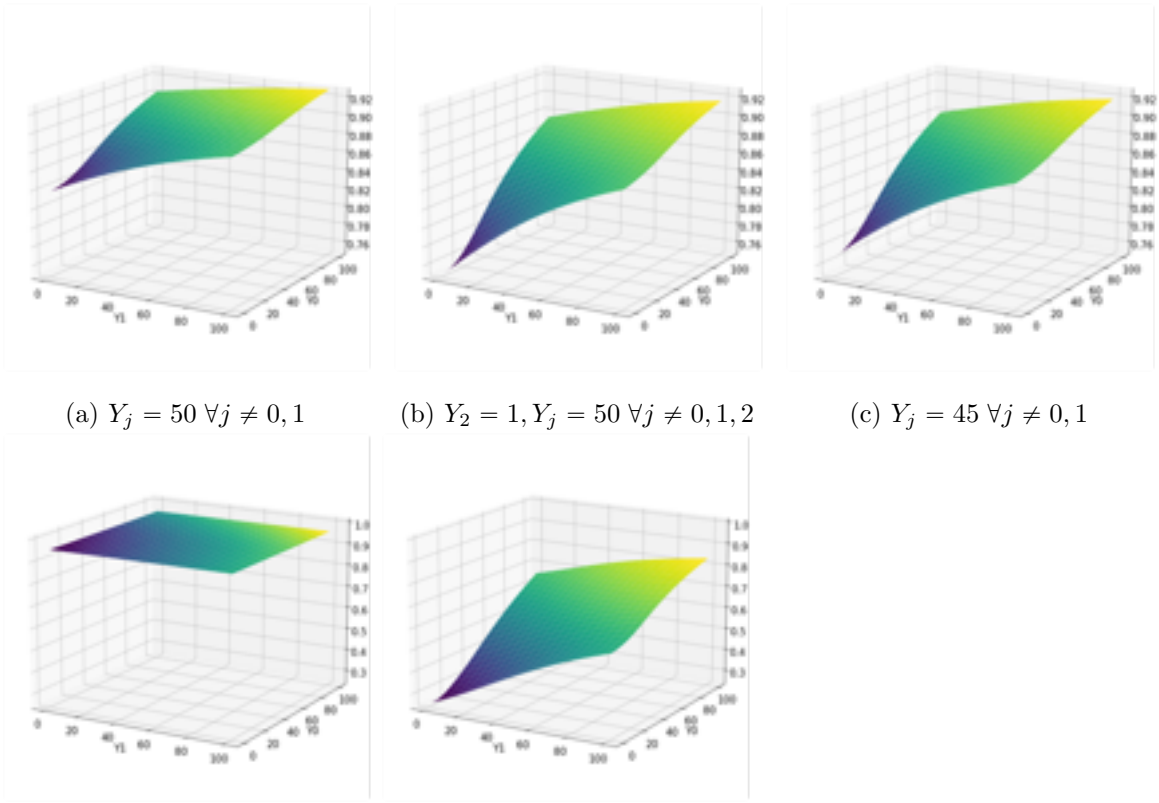
Figure 5: Dividend yield ($F_{0,t}$), risk premium ($\mu_{R_0,t} - r_t$), first two diffusion terms ($\sigma_{R_0 z_0,t}, \sigma_{R_0 z_1,t}$) for asset 0, and locally riskless interest rate r_t,



Among other things, the solution allows us to study the impact of the average output of other trees on the returns correlation between two of them. This is shown in Figure 6, which presents the (instantaneous) correlation between the returns of asset 0 and asset 1, $corr_t(dR_{0,t}, dR_{1,t})$, as a function of $Y_{0,t}, Y_{1,t}$, and various values of $\{Y_{j,t}\}$ for other js. Interestingly, the output of one tree being extremely small (Panel (b)) can have the same effect on the returns comovement of two other assets as an overall small decrease in the output of all other trees (Panel (c)). This could for instance be used to study the impact of severe negative shocks to one asset, or one part of the market or one sector of the economy, to asset prices, returns correlations, and risk sharing between investors more generally. From a broad perspective, the

correlation between two assets can vary substantially quantitatively with the output to other trees. For instance, it goes from close to 1 for any $(Y_{0,t}, Y_{1,t})' \in [0, 100]^2$ when $Y_j = 100 \ \forall j \neq 0, 1$ (Panel (d)), to being much more state-dependent and varying between 0.3 and 0.9 when $Y_j = 30 \ \forall j \neq 0, 1$ (Panel (e)). Those levels, and patterns, which are purely endogenous, would again evolve significantly for non-zero levels of correlations across Brownian motions, or in the presence of several goods or more general preferences (cf. for instance Sauzet, 2021b).

Figure 6: (Instantaneous) Returns correlation of assets 0 and 1 $corr_t(dR_{0,t}, dR_{1,t})$ as a function of $Y_{0,t}, Y_{1,t}$ and various $\{Y_{j,t}\}$ for other js ($\gamma = \psi^{-1} = 5$)



(a) $Y_j = 50 \ \forall j \neq 0, 1$ (b) $Y_2 = 1, Y_j = 50 \ \forall j \neq 0, 1, 2$ (c) $Y_j = 45 \ \forall j \neq 0, 1$

(d) $Y_j = 100 \ \forall j \neq 0, 1$ (e) $Y_j = 30 \ \forall j \neq 0, 1$

Overall, although the focus here is mostly on the numerical approach and this only scratches the surface of what is possible, this example shows that the method can be promising for settings with meaningful economic applications. For instance, an economy with multiple trees such as this could be used to study the impact of indexing on the risk premia and correlation of (potentially many) asset returns in

general equilibrium, and could be combined with multiple agents and heterogeneity in the spirit of the model I turn to now.

3.3. Two investors, two trees, two goods (222)

Lastly, I solve the two-investor, two-tree, two-good framework of Sauzet (2021b), i.e. the 222-version of the NML framework. I show that the method can allow us to solve the model with more grid points, to study parameter configurations that were difficult or extremely slow to explore with a solution based on Chebyshev polynomials, as well as to start introducing portfolio constraints in such an economy without apparent difficulty.

The setup is that of a pure exchange economy with two trees producing two differentiated consumption goods ($j \in \{1, 2\}$), and populated by two heterogeneous investors ($i \in \{A, B\}$) with recursive preferences whose parameters can differ, and a bias in consumption towards one of the goods. Financial markets are also allowed to be imperfectly integrated, in which case the stochastic discount factors of both investors are not perfectly correlated. This is captured at a high-level as a tax on the dividends of the tree producing the least preferred good of an investor. Because I solve for the decentralized equilibrium, the solution method can also handle cases beyond imperfect risk sharing, and in which markets are incomplete. The setup lends itself to several applications and extensions, e.g. in international or environmental contexts (cf. Sauzet, 2021b,c,d,e, for details).

In Sauzet (2021b), I show that the equilibrium can be recast as a stationary recursive Markovian equilibrium with state variables $X_t \equiv (x_t, y_t)' \in [0, 1]^2$, where x_t is the wealth share of investor A, and y_t is the relative supply of good 1. They are defined as

$$x_t \equiv \frac{W_t^A}{W_t^A + W_t^B} \quad ; \quad y_t \equiv \frac{Y_{1,t}}{Y_{1,t} + Y_{2,t}} \tag{21}$$

All variables of interest can be expressed as a function of X_t, and a set $\mathcal{G}$ of unknown functions of X_t, $g : [0, 1]^2 \to \mathcal{D}^g \subseteq \mathbb{R}$:

$$\mathcal{G} \equiv \{J_t^A, J_t^B, F_{1,t}, F_{2,t}, q_t, w_{1,t}^A, w_{2,t}^A\} \tag{22}$$

where J_t^i are the marginal values of wealth of both investors, $F_{j,t}$ are the dividend yields on both assets, q_t is the relative price of good 2, and $w_{j,t}^A$ are the weights on assets 1 and 2 in the portfolios of investor A as a share of her wealth. The share of the bond in the portfolio of each investor, which is non-zero when investors have asymmetric preferences beyond the bias in consumption, is $b_t^i = 1 - w_{1,t}^i - w_{2,t}^i$.

The system $\mathcal{H}$ pinning down those unknown functions consists of coupled algebraic and second-order partial differential equations. It invoves, for instance, the Hamilton-Jacobi-Bellman equations of both investors, asset pricing equations featuring the risk premia on both assets, optimal portfolio conditions, and other equilibrium conditions pinning the relative price of consumption goods. All those equations combine the unknown functions in $\mathcal{G}$ and the state variables in non-linear ways, so that they have to be solved jointly. I also add a few equations that help the convergence. Appendix B.2 presents some of those equations, and the reader is referred to Sauzet (2021b) for details.

Figure 7: 222: projection with Chebyshev ($\mathcal{O} = 30$, top row), projection with neural networks (bottom row)

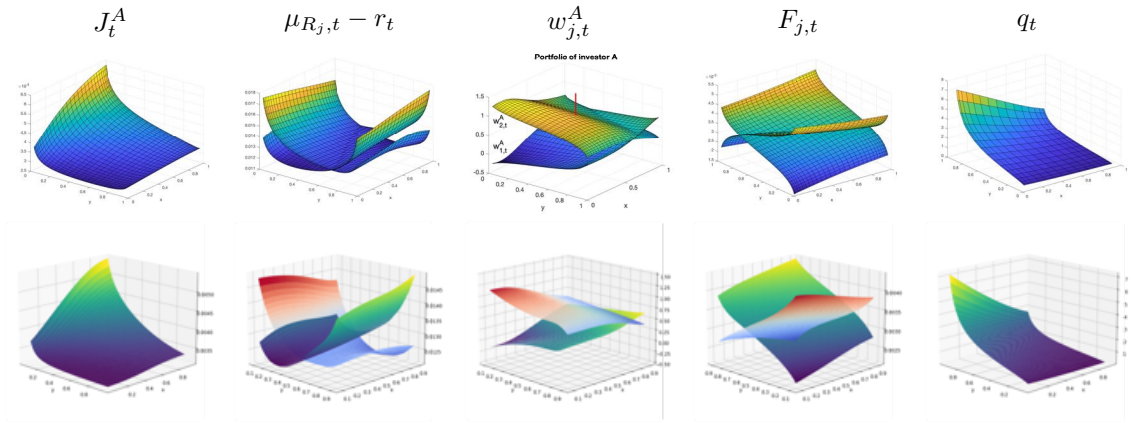


Figure 7 shows the solution obtained from traditional projection methods (top row), and the projection methods via neural networks proposed in this paper (bottom row).[15] Comparing the two confirms that the projection methods via neural networks

[15]Unless otherwise specificed, the baseline calibration is that of Sauzet (2021b). I.e., for $i \in \{A, B\}, j \in \{1, 2\}$: risk aversion $\gamma^i = \gamma = 15$, elasticity of intertemporal substituion $\psi^i = \psi = 2$ (non-CRRA), bias in consumption $\alpha^A = \alpha = 0.75$, $\alpha^B = 1 - \alpha$, numéraire basket $a = 1/2$, elasticity of substitution between goods $\theta^i = \theta = 2$, discount rate $\rho^i = \rho = 1\%$, no labor income

delivers an extremely accurate solution, comparable to that obtained with traditional methods. In fact, because going beyond an order of $\mathcal{O} = 30$ (i.e. 30*30 grid points) for traditional approaches in this two-state-variable setting is computationally already difficult, the solution obtained from neural networks is potentially closer to the "true solution", given that it is based on 10,000 grid points. If required, it could even accomodate more points without much additional computational burden if the batch size remains reasonable.

The method also allows us to study parameter configurations that were difficult to explore with traditional projection methods. A typical example is the elasticity of substitution across goods θ. Because this parameter has a stark quantitative but also qualitative impact on portfolios, relative dividends, and other variables, converging to the solution with the method based on Chebyshev polynomials turns out to require moving θ in a very gradual manner, so that obtaining the final solution can be extremely slow. Obtaining it for values of θ that imply strong non-linearities can also require increasing the order of approximation, making the endeavor even more computationally difficult. In comparison, and although obtaining the solution still requires gradually moving in the parameter space as is common with projection approaches, the method based on neural networks allows to use much larger parameter steps. In practice, this makes the exploration of those parameters much faster, while still using many more grid points so that the solution is potentially more accurate.

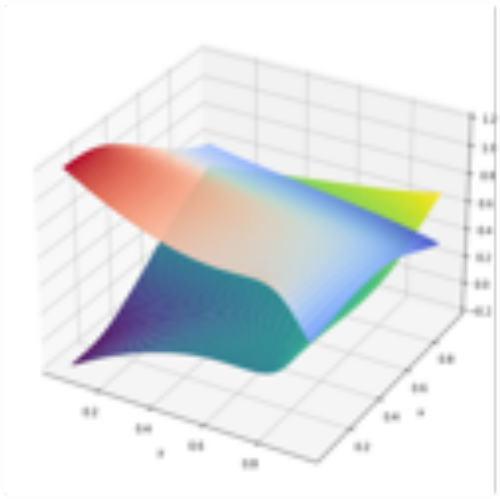
Figure 8 shows the resulting portfolio weights of asset 1 and 2 in the portfolio of investor A $(w^A_{1,t}, w^A_{2,t})$, as a function of x_t the wealth share of investor A and y_t the relative supply of good 1, for various parameter configurations. It confirms the results in Sauzet (2021b), some of which where discussed mostly conceptually. First, in the baseline calibration in which goods are somewhat good substitutes ($\theta = 2 > 1$, Panel (a)), investor A turns out to optimally choose a "foreign bias" in her portfolio, in the sense that she invests vastly more in asset 2, which produces good 2 that she *least* likes (recall that investor A has a preference for good 1, $\alpha^A = 0.75 > 1/2$). This surprising result arises because she invests in the asset whose returns are highest when her marginal value of wealth is high. This happens when the relative supply of her preferred good, good 1, is low, in which case she would like to consume more of that

$\delta_j = \delta = 0$, fully integrated financial markets $\tau^i = \tau = 0$, output $\mu_{Y_j} = \mu_Y = 2\%, \sigma_{Y_1} = (4.1\%, 0)^T, \sigma_{Y_2} = (0, 4.1\%)^T$ (no fundamental correlation).

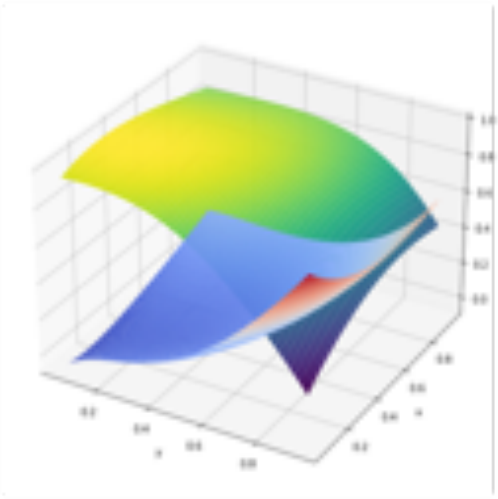
good but cannot, and it turns out that asset 2, whose supply is relatively large, pays well.

Figure 8: Weights of asset 1 and 2 in the portfolio of investor A $(w^A_{1,t}, w^A_{2,t})$

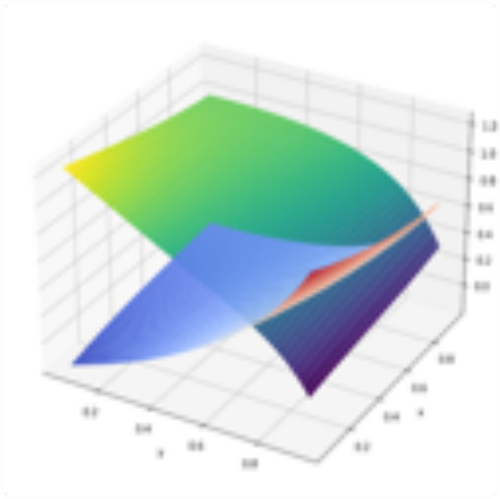
(a) $w^A_{j,t}$ (baseline)



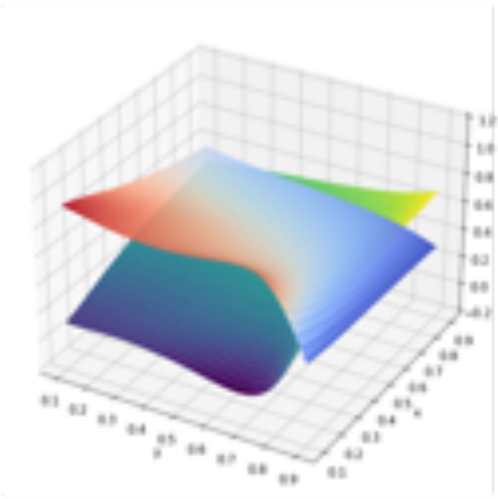
(b) $w^A_{j,t}$ $(\theta = 0.5)$



(c) $w^A_{j,t}$ $(\psi = 0.2, \theta = 0.5)$



(d) $w^A_{j,t}$ $(\tau = 25\%)$



Interestingly, this result hinges on how the relative dividends on both assets evolve with the relative supply of the goods, which itself depends crucially on the elasticity of substitution across goods θ. As θ gets low, especially lower than 1, the evolution of relative dividends of asset 2, $p_{2,t}Y_{2,t}/(p_{1,t}Y_{1,t})$, as a function of relative supply flips direction because even though the relative supply of good 2 itself increases, the relative price of good 2 decreases faster due to the difficulty in substituting the goods.

The end result is that portfolios completely flip and investor A optimally chooses a "home bias" in her equity holdings. This is confirmed quantitatively in Panel (b) and Panel (c) of Figure 8, which plot her portfolio for $\theta = 0.5 < 1$ for a baseline value of the IES $\psi = 2$, and for an alternative value of the IES $\psi = 0.2$, respectively. The value of the IES itself also turns out to have a strong impact on the equilibrium. Namely, if we use the baseline value of $\theta = 2$, a relatively modest degree of friction in financial integration of around $\tau = 10\%$ is enough to reverse the "foreign" portfolio bias towards the tree producing the least preferred good (e.g. good 1 for investor A) into a "home bias" towards the tree that produces the preferred one, *when the IES is low* $\psi = 0.2$*, i.e. when investors are close to CRRA.* When investors get further away from CRRA and towards recursive preferences however, e.g. $\psi = 2$, the quantitative effect of imperfectly integrated markets is in fact much smaller so that even large taxes of $\tau = 25$ or 50% are not enough to flip portflios. This is shown in Panel (d), in which the portfolios of investor A is broadly comparable to the baseline calibration of Panel (a) despite a friction in financial integration as large as $\tau = 25\%$.[16] Overall, the quantitative discussion of those results here, and the analysis of their impact on portfolios throughout the whole state space, is made possible – or at least greatly eased and accelerated – by the projection methods via neural networks proposed in this paper.

Finally, the method also makes it possible to start exploring elements that were simply not possible with traditional methods. One such element, which is particularly relevant for the types of models of interest in the literature, are constraints. Those constraints could stem from the so-called zero lower bound on interest rates, which has been discussed at length in the macroeconomics literature, from balance-sheet constraints (e.g. value-at-risk or skin-in-the-game), whose importance has been emphasized in the recent intermediary asset pricing literature, or simply portfolio constraints, which have the potential to alter equilibrium allocations and asset prices strongly.

To illustrate this point, I introduce portfolio constraints in the 222 framework above. Specifically, investors are now prevented from shorting both individual equity

[16] Note that all those results were discussed based on $w^A_{j,t}$, but hold true when we compare raw portfolio weights with the market portfolio $z_t \equiv Q_{1,t}/(Q_{1,t} + Q_{2,t})$. This comparison makes the discussion of a "bias" in portfolio holdings more formal, and is used and discussed in Sauzet (2021b).

assets, and cannot borrow in the bond more than 10% of their wealth, i.e. for $i \in \{A, B\}, j \in \{1, 2\}$

$$\begin{aligned} w^i_{j,t} &\geqslant 0 \\ b^i_t = 1 - w^i_{1,t} - w^i_{2,t} &\geqslant \bar{b} = -10\% \end{aligned} \tag{23}$$

Because the goal is illustrative here, I enforce the constraints simply by imposing portfolio weights to take adequate values via max operators, and a formal analysis relying on Kuhn-Tucker multipliers is left for ongoing research. For those constraints to have bite, I also introduce asymmetries in the preferences of the investors. Namely, investor A is now more risk-tolerant $\gamma^A = 10 < \gamma^B = 15$. Other asymmetries could also be introduced, as the method does not rely on any particular restrictions on parameters.

Figure 9 plots the results for the equity portfolio weights and the bond position of investor A $(w^A_{1,t}, w^A_{2,t}, b^A_t)$. Focusing first on the left column that does not impose constraints, we observe that investor A is now willing to take more risks by holding larger positions in both risky assets (Panel (a)). To do so, she leverages up her equity portfolio by borrowing heavily – up to about 50% of her wealth in parts of the state space – from investor B in the bond market (Panel (c)). This is consistent for instance with an application of the framework to international finance in which investor A represents the United States, which is borrowing heavily from the rest of the world (investor B) in safe assets and holding a large risky portfolios worldwide. For details of this application, cf. Sauzet (2021c), in which I show that the asymmetric version of the framework can capture a large number of stylized facts about the International Financial System both on average, and in terms of its dynamics in times of crisis.

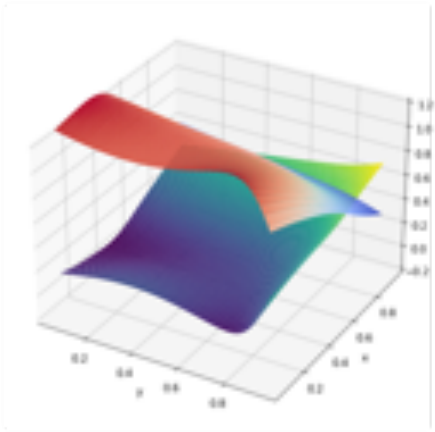
Panel (b) and Panel (d) show what happens when imposing portfolio constraints. As investor A is now prohibited from borrowing a too large share of her wealth, she does so up to the limit of $\bar{b} = -10\%$ for most of the state space, resulting in a large kink around $x_t = 0.8$. This in turn prevents her from leveraging her risky portfolios so that the weights in assets 1 and 2 $(w^A_{1,t}, w^A_{2,t})$ display a plateau for lower values of her wealth share, say $x_t < 50\%$.

Studying the economic impact of such (or more sophisticated) constraints more formally is an important avenue for ongoing and future research. From the perspective of this paper, what this example shows is that the projection methods via neural networks have no difficulty whatsoever capturing strong non-linearities such as kinks

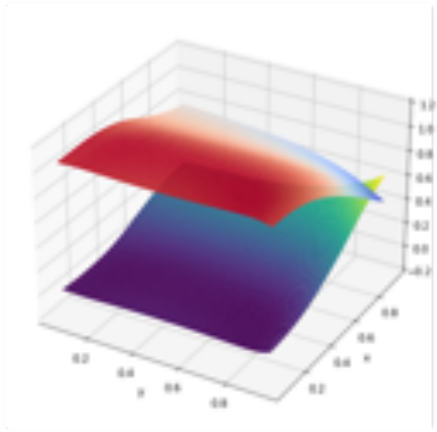
and constraints. This is remarkable because capturing such constraints using traditional projection approaches would require an order of approximation that rapidly becomes prohibitively expensive as the number of state variables increases. For instance, an order of $\mathcal{O} = 100$, as was used to capture a kink in the simple one-equation one-unknown function one-state-variable example of Appendix A, would simply be infeasible even for two state variables.

Figure 9: Asymmetry ($\gamma^A = 10 < \gamma^B = 15$) + constraints (right): $b_t \geqslant \bar{b} = -10\%$, and no shorting

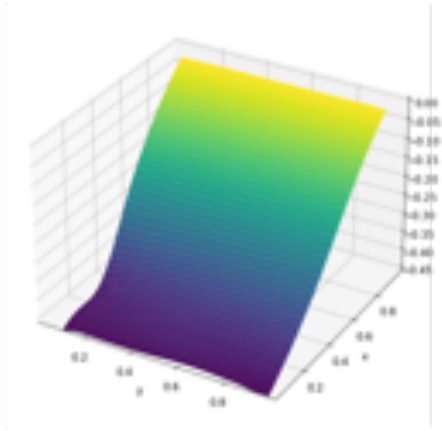
(a) $w^A_{j,t}$



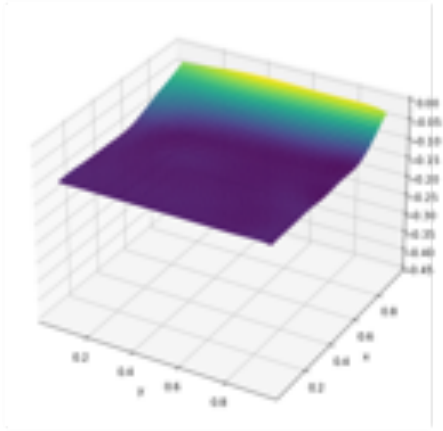
(b) $w^A_{j,t}$



(c) b^A_t



(d) b^A_t



Overall, even in this application to the two-investor two-tree two-good framework, the method therefore already allows us to solve the model with more grid points, to study parameter configurations that were difficult or extremely slow to explore with a solution based on Chebyshev polynomials, and to start introducing portfolio constraints in such an economy without apparent difficulty.

In summary of Section 3, I presented a number of applications of the projection methods via neural networks proposed in this paper. Those are just a few examples of a broader set of applications that I am currently developing: e.g. investors could have recursive (beyond the 222 case) or other types of preferences, such as general HARA, the stochastic processes driving the economy could be expanded or production introduced, and the number of trees and agents could be increased further, given that I for the moment only train the neural networks on CPUs and do not make use of various optimizations and of specialized hardward like GPUs, TPUs. What those examples hightlight however, is that the method appears to be extremely promising to solve models that were computationally simply out of reach before.

4. Conclusion

In this paper, I alleviate difficulties with global resolution approaches by combining projection methods with neural networks as function approximator, to solve continuous-time models. The method appears promising to solve models with multiple heterogeneous agents, high-dimensional state spaces, and potentially very nonlinear policy functions such as those that arise in the presence of constraints.

The projection methods via neural networks has several advantages.

First, because neural networks are designed for such settings, the method is well-suited for high-dimensional contexts with many state variables, as well as many grid points. This is made possible in part by the use of automatic differentation, and stochastic gradient descent, two important elements that allow the training of large models. Automatic differentation in particular makes it possible to obtain derivatives very cheaply, which is especially helpful in a continuous-time setting in which expectations can often be turned into partial derivatives. As shown in Barron (1993), Bach (2017), and as discussed in Fernández-Villaverde et al. (2019), neural networks are also more economical than other approximators for middle and high dimensions in that they achieve a lower order of integrated squared errors than e.g. series approximations (polynomials, splines, trigonometric expansions, *etc.*). Taken together, those suggest that neural networks can relax the curse of dimensionality both theoretically, and in practice.

Second, neural networks can approximate any function, even very non-linear, arbitrary well. This has been shown theoretically and is known as the Universal Approximation Theorem (Hornik et al., 1989), but as importantly, turns out to be verified in practice. For instance, strong non-linearities such as kinks, which appear in economies with constraints, pose no particular problems. Relatedly, neural networks also generalize particularly well, even when the domain of the state variables is unbounded or at least very wide (cf. e.g. Fernández-Villaverde et al., 2019), or between grid points, with much less of the oscillations or wild swings that are not uncommon with polynomials.

Third, because I fit the method within the more usual projection approaches framework, it is naturally able to deal with models whose solutions depend on several (potentially many) unknown functions, which are jointly determined in large systems of coupled algebraic and differential equations, and would not be easy or even possible to solve based on more usual iterative methods. This is for instance the case of equilibrium models with multiple heterogeneous agents, and imperfect risk sharing, in which several value functions (as well as other unknown functions) have to be solved for because a representative agent does not necessarily exist. More generally, studying how to generalize projection (i.e. spectral) methods beyond polynomials and to higher-dimensional cases is also of interest on its own, given that those methods have been growing more popular.

Fourth, the method does not rely on simulating paths of the economy to solve for unknown functions. Instead, grid points are drawn on the whole support of the state variables, which means that draws essentially come for free. While this is computationally more costly than relying solely on the parts of the state space that are most often visited in equilibrium, it allows to trace out the solution to the economy throughout the whole state space. This is particularly useful when one is interested in the behavior of agents and of our variables of interest in rare events that occur only with extremely small probability. Drawing points throughout the domain also help neural networks generalize well, although they can do so even with few training points. Finally, this also allows the method to deal with non-stationary models, which do not feature a non-degenerate ergodic distribution but are common in finance e.g. in the applications discussed in Section 3.[17]

[17] The sampling of points could still be updated to feature a larger mass of points in the regions most visited in equilibrium, or even points mostly in this region, depending on the model of interest.

To demonstrate the power of the method, I apply it to the economy with N investors, M trees, L goods (NML framework) being developped in Sauzet (2021a). This context provides a perfect testing ground given that the generality of preferences, and the potential frictions in financial markets, make the equilibrium such that several unknown functions (value functions, policy functions, equilibrium functions) have to be solved for jointly. I focus on three examples.

First, I solve a generalized version of a two-tree model à la Cochrane et al. (2008) and Martin (2013) in which the trees produce two differentiated goods. I show that the method reproduces the solution obtained using Chebyshev polynomials, but with the potential to have many grid points and flexibility.

Second, I solve a M-tree version of those models, with one consumption good, CRRA preferences, and $M \geqslant 10$ trees. The solution allows us to study, among other things, the impact of the average output of other trees on the returns correlation between two of them. Interestingly, the output of one tree being extremely small can have the same effect on the returns comovement of two other assets as an overall small decrease in the output of all other trees. This could for instance be used to study the impact of severe negative shocks to one asset, or one part of the market or one sector of the economy, to asset prices, returns correlations, and risk sharing between investors more generally.

Lastly, I solve the two-investor two-tree two-good framework of Sauzet (2021b), and show that the method can allow us to solve the model with more grid points, to study parameter configurations that were difficult or slow to explore with a solution based on Chebyshev polynomials, as well as to start introducing portfolio constraints in such an economy without apparent difficulty.

Those are just a few examples of a broader set of applications that I am currently developing: e.g. investors could have recursive (beyond the 222 case) or other types of preferences, such as general HARA, the stochastic processes driving the economy could be expanded or production introduced, and the number of trees and agents could be increased further, given that I for the moment only train the neural networks on CPUs and do not make use of various optimizations and of specialized hardward like GPUs, TPUs. What those examples hightlight however, is that the method appears to be extremely promising to solve models that were computationally simply out of reach before.

References

Achdou, Yves, Jiequn Han, Jean-Michel Lasry, Pierre-Louis Lions, and Benjamin Moll, 2021, Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach, *The Review of Economic Studies* rdab002.

Ahn, SeHyoun, Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf, 2018, When inequality matters for macro and macro matters for inequality, *NBER Macroeconomics Annual* 32, 1–75.

Bach, Francis, 2017, Breaking the curse of dimensionality with convex neural networks, *Journal of Machine Learning Research* 18, 1–53.

Barron, A.R., 1993, Universal approximation bounds for superpositions of a sigmoidal function, *IEEE Transactions on Information Theory* 39, 930–945.

Boyd, John P., 2001, *Chebyshev and Fourier Spectral Methods*, Dover Books on Mathematics, second edition (Dover Publications, Mineola, NY).

Brunnermeier, Markus, and Yuliy Sannikov, 2014, A macroeconomic model with a financial sector, *American Economic Review* 104, 379–421.

Brunnermeier, Markus K., and Yuliy Sannikov, 2015, International credit flows and pecuniary externalities, *American Economic Journal: Macroeconomics* 7, 297–338.

Brunnermeier, Markus K., and Yuliy Sannikov, 2019, International monetary theory: A risk perspective, *Working Paper* .

Brunnermeier, M.K., and Y. Sannikov, 2016, Chapter 18 - macro, money, and finance: A continuous-time approach, volume 2 of *Handbook of Macroeconomics*, 1497–1545 (Elsevier).

Campbell, John Y., and Robert J. Shiller, 1988, The Dividend-Price Ratio and Expectations of Future Dividends and Discount Factors, *Review of Financial Studies* 1, 195–228.

Campbell, John Y., and Luis M. Viceira, 1999, Consumption and portfolio decisions when expected returns are time varying, *The Quarterly Journal of Economics* 114, 433–495.

Chacko, George, and Luis M. Viceira, 2005, Dynamic Consumption and Portfolio Choice with Stochastic Volatility in Incomplete Markets, *The Review of Financial Studies* 18, 1369–1402.

Cochrane, John H., Francis A. Longstaff, and Pedro Santa-Clara, 2008, Two Trees, *The Review of Financial Studies* 21, 347–385.

Drechsler, Itamar, Alexi Savov, and Philipp Schnabl, 2018, A model of monetary policy and risk premia, *The Journal of Finance* 73, 317–373.

Duarte, Victor, 2019, Machine learning for continuous-time economics, *Available at SSRN: https://ssrn.com/abstract=3012602* .

Duffie, Darrell, and Larry Epstein, 1992, Stochastic differential utility, *Econometrica* 60, 353–94.

Ebrahimi Kahou, Mahdi, Jesús Fernández-Villaverde, Jesse Perla, and Arnav Sood, 2021, Exploiting symmetry in high-dimensional dynamic programming, Working Paper 28981, National Bureau of Economic Research.

Fang, Xiang, 2019, Intermediary leverage and currency risk premium, *Available at SSRN: https://ssrn.com/abstract=3290317.* .

Fernández-Villaverde, J., J.F. Rubio-Ramírez, and F. Schorfheide, 2016, Chapter 9 - solution and estimation methods for dsge models, volume 2 of *Handbook of Macroeconomics*, 527–724 (Elsevier).

Fernández-Villaverde, Jesus, and Lawrence Christiano, 2011, Nber summer institute, computational tools & macroeconomic applications, Methods lectures.

Fernández-Villaverde, Jesús, Grey Gordon, Pablo Guerrón-Quintana, and Juan F. Rubio-Ramírez, 2015, Nonlinear adventures at the zero lower bound, *Journal of Economic Dynamics and Control* 57, 182–204.

Fernández-Villaverde, Jesús, Samuel Hurtado, and Galo Nuño, 2019, Financial frictions and the wealth distribution, Working Paper 26302, National Bureau of Economic Research.

Fernández-Villaverde, Jesús, and Oren Levintal, 2018, Solution methods for models with rare disasters, *Quantitative Economics* 9, 903–944.

Han, Jiequn, and Weinan E, 2016, Deep learning approximation for stochastic control problems, *CoRR* abs/1611.07422.

Han, Jiequn, Arnulf Jentzen, and Weinan E, 2018, Solving high-dimensional partial differential equations using deep learning, *Proceedings of the National Academy of Sciences* 115, 8505–8510.

Hansen, Lars Peter, Paymon Khorrami, and Fabrice Tourre, 2018, Comparative valuation dynamics in models with financing restrictions .

Hornik, Kurt, Maxwell Stinchcombe, and Halbert White, 1989, Multilayer feedforward networks are universal approximators, *Neural Networks* 2, 359–366.

Judd, Kenneth, 1992, Projection methods for solving aggregate growth models, *Journal of Economic Theory* 58, 410–452.

Judd, Kenneth, 1996, Approximation, perturbation, and projection methods in economic analysis, in H. M. Amman, D. A. Kendrick, and J. Rust, eds., *Handbook of Computational Economics*, volume 1, first edition, chapter 12, 509–585 (Elsevier).

Judd, Kenneth, 1998, *Numerical Methods in Economics*, volume 1, first edition (The MIT Press).

Judd, Kenneth, Lilia Maliar, Serguei Maliar, and Rafael Valero, 2014, Smolyak method for solving dynamic economic models: Lagrange interpolation, anisotropic grid and adaptive domain, *Journal of Economic Dynamics and Control* 44, 92–123.

Kaplan, Greg, Benjamin Moll, and Giovanni L. Violante, 2018, Monetary policy according to hank, *American Economic Review* 108, 697–743.

Kargar, Mahyar, 2019, Heterogeneous intermediary asset pricing, *Available at SSRN: https://ssrn.com/abstract=3258119* .

Krusell, Per, and Anthony A. Smith, Jr., 1998, Income and wealth heterogeneity in the macroeconomy, *Journal of Political Economy* 106, 867–896.

Kubler, Felix, and Karl Schmedders, 2003, Stationary Equilibria in Asset-Pricing Models with Incomplete Markets and Collateral, *Econometrica* 71, 1767–1793.

Levintal, Oren, 2018, Taylor projection: A new solution method for dynamic general equilibrium models, *International Economic Review* 59, 1345–1373.

Maliar, Lilia, and Serguei Maliar, 2014, Chapter 7 - numerical methods for large-scale dynamic economic models, in Karl Schmedders, and Kenneth L. Judd, eds., *Handbook of Computational Economics Vol. 3*, volume 3 of *Handbook of Computational Economics*, 325–477 (Elsevier).

Maliar, Lilia, and Serguei Maliar, 2015, Merging simulation and projection approaches to solve high-dimensional problems with an application to a new keynesian model, *Quantitative Economics* 6, 1–47.

Maliar, Lilia, Serguei Maliar, and Pablo Winant, 2019, Will artificial intelligence replace computational economists any time soon?, CEPR Discussion Papers 14024, C.E.P.R. Discussion Papers.

Martin, Ian, 2013, The lucas orchard, *Econometrica* 81, 55–111.

Sauzet, Maxime, 2021a, N investors, m trees, l goods, Work in progress.

Sauzet, Maxime, 2021b, Two investors, two trees, two goods, Working paper.

Sauzet, Maxime, 2021c, Asset prices, global portfolios, and the international financial system, Working paper.

Sauzet, Maxime, 2021d, Asset prices, portfolios, allocation of wealth, and the environment, Work in progress.

Sauzet, Maxime, 2021e, Global asset manager and the global financial cycle, Work in progress.

Trefethen, Lloyd N., 2000, *Spectral Methods in MatLab* (Society for Industrial and Applied Mathematics, USA).

Tsai, Jerry, and Jessica A. Wachter, 2018, Pricing long-lived securities in dynamic endowment economies, *Journal of Economic Theory* 177, 848–878.

Appendix

A. Example of non-linearity: kinks

As discussed in the main text, strong non-linearities are very relevant for the models of interest in the literature. For instance, strong non-linearities such as kinks arise when those models involve constraints (e.g. portfolio constraints, zero lower bound, value-at-risk, *etc.*). To illustrate the difficulty of traditional methods in capturing such strong non-linearities, I use Chebyshev polynomials of various orders in a simple one-equation one-unknown-function one-state-variable case. Specifically, I use them to capture the kink shown in the top left panel of Figure A.1. The other panels present the resulting approximation based on Chebyshev polynomials for various orders of approximation, $\mathcal{O} = 1, 2, 3, 4, 5, 10, 20, 30, 100$, obtained using the *chebfun* package (Trefethen, 2000). Those results suggest that an extremely high order of approximation is required to capture the kink, even in this stylized one-dimensional setting. Indeed, even at $\mathcal{O} = 30$, the Chebyshev approximation displays oscillations that could be problematic if a high accuracy is required, and only when $\mathcal{O} = 100$ does the approximation look visually mostly adequate.

However, at least two problems remain.

First, when zooming in, Panel (a) of Figure A.2 reveals that the issue does not disappear even at $\mathcal{O} = 100$. This is so because those oscillations are related to the well-known Gibbs phenomenon affecting Chebyshev polynomials, an another example of which is shown in one and two dimensions in Panel (b) of the same figure.

Second, even if we consider such higher orders to be sufficient in terms of accuracy, they are computationally unfeasible as soon as we increase the number of state variables, which is another important aspect to be able to solve more ambitious models. In other words, the curse of dimensionality rapidly comes biting back. For instance, using the tensor product of Chebyshev polynomials, an order of $\mathcal{O} = 100$ and three state variables require an array of size 7908.9GB in memory, completely out of reach even for very large servers. Similarly, even an order as low as $\mathcal{O} = 10$, and 5 state variables require an array of size 193.2GB, already mostly infeasible even for large servers. Of course, replacing tensor products by more parsimonious methods to construct grids and the basis such as complete polynomials or Smolyak's algorithm can

help (cf. for instance Judd et al., 2014). However, the dimensionality is still rapidly much too high, and such methods can also lead to other issues if non-linearities are far from the less numerous grid points.

In short, traditional methods based on Chebyshev polynomials struggle with strong non-linearities, and high-dimensional settings.

In comparison, the projection methods via neural networks that I propose in this paper have no particular difficulty dealing with either. Cf. the main text for a discussion of high dimensionality, e.g. with the M-tree application of Part 3.2. For strong non-linearities, neural networks are particularly adequate both in theory (Universal Approximation Theorem of Hornik et al. (1989), discussed in the main text), and in practice. The latter is shown for the case of portfolio constraints in the two-investor two-tree two-good setting of Part 3.3. As another example, Figure A.3 shows that neural networks have no particular difficulty in capturing the kink discussed for Chebyshev polynomials above. This is true even when zooming in (Figure A.4), and in two-dimensional settings (Figure A.5).

For those to be possible, the neural networks must be rich enough. For instance, in those examples, I use networks with multiple layers that already involve many parameters. A key difference is that those do not suffer from issues such that the Gibbs phenomenon. Even more importantly, they are still easy to train and scalable thanks to the apparatus behind their training (stochastic gradient descent, automatic differentiation, *etc.*). In practice for instance, I train the networks in Figures A.3, A.4, A.5 for only about a minute on a personal laptop. Another helpful aspect is that, unlike with traditional methods, there is a decoupling between the richness of the networks, and the number of grid points. For instance, the number of grid points can be increased significantly without changing the number of parameters to estimate, and without much additional computational burden if the batch size remains reasonable. This aspect might not be as relevant for smaller-scale models in which the number of parameters that can be estimated with neural networks is already orders of magnitude larger than that using traditional approaches, but could be helpful for yet larger-scale setups.

In short, the projection methods via neural networks that I propose in this paper does not seem to have difficutlies with situations that are very relevant for the models

of interest in the literature: strong non-linearities, and higher-dimensional settings.

Figure A.1: Strong non-linearity and traditional methods (Chebyshev polynomials)

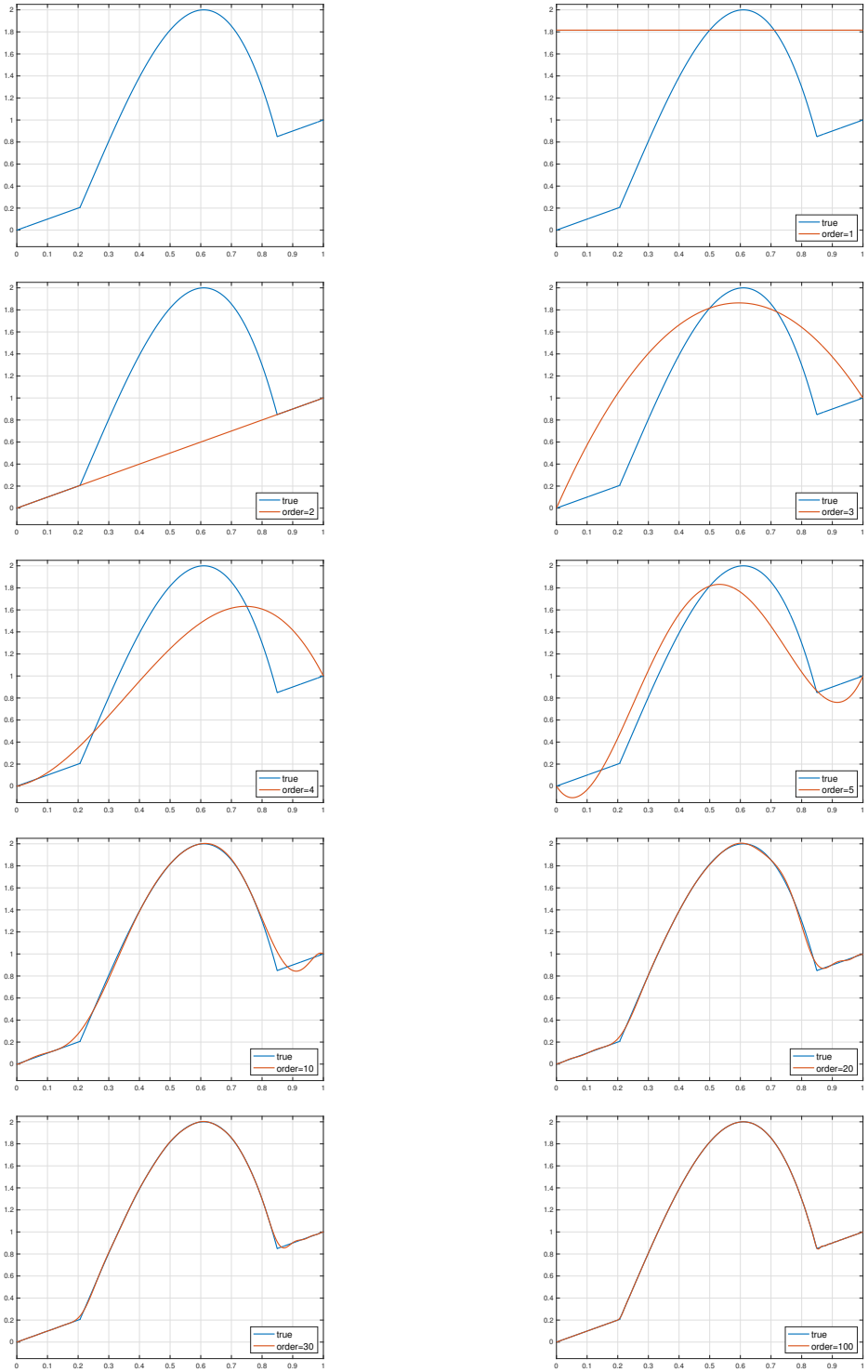
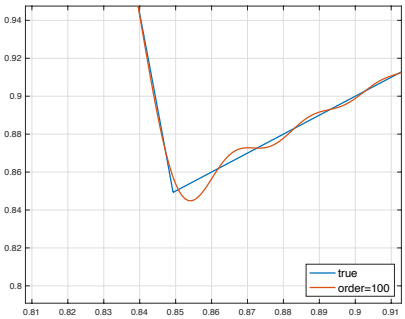


Figure A.2: Strong non-linearity and traditional methods (Chebyshev polynomials)

(a) Zoom in on order = 100



(b) Related: Gibbs phenomenon

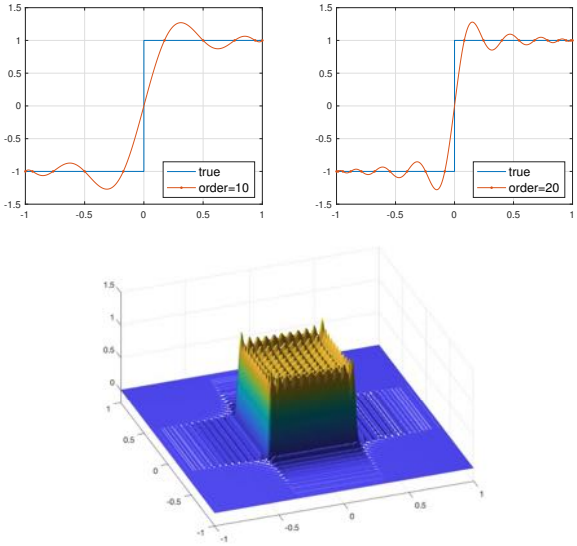


Figure A.3: Strong non-linearity and projection methods via neural networks

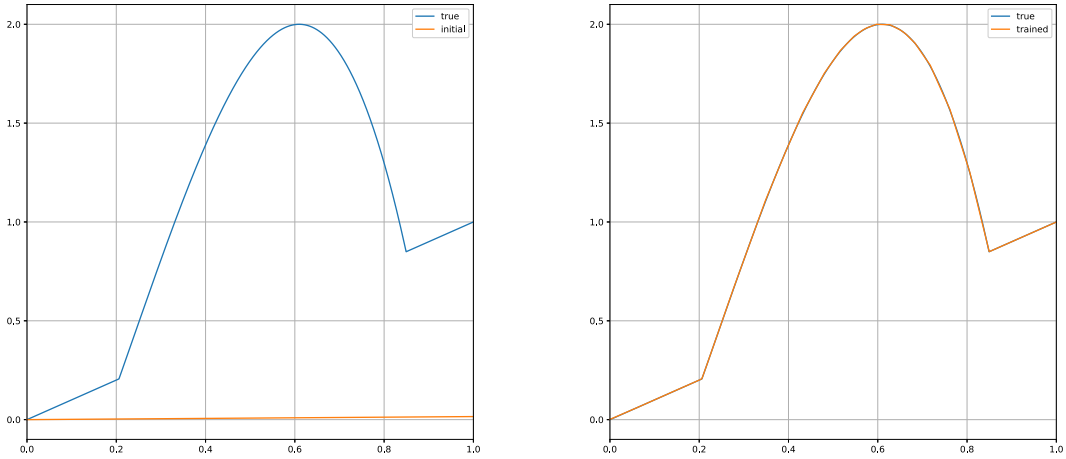


Figure A.4: Strong non-linearity and projection methods via neural networks (zoom)

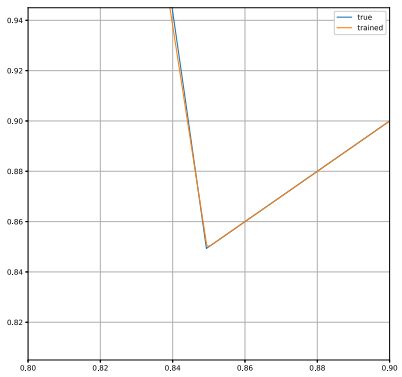
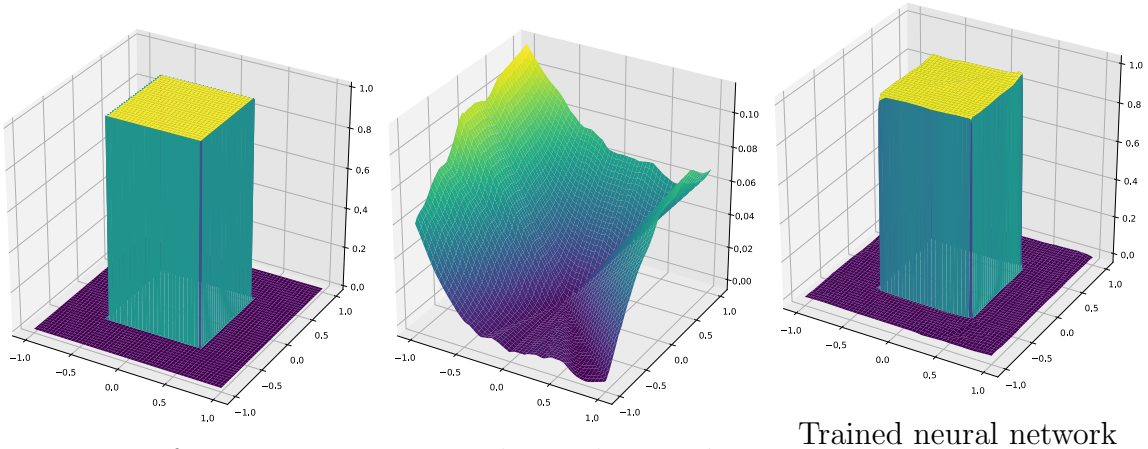


Figure A.5: Strong non-linearity and projection methods via neural networks (2D)



True function

Initial neural network

Trained neural network
(1500 ep., 1.5 min)

B. Additional equations and results

B.1. M trees

Recall the Duffie and Epstein (1992) preferences of investors, shown in Equation (9) of the main text

$$V_t = \max_{\{C_u, \vec{w}_u\}_{u=t}^{\infty}} \mathbb{E}_t \left[\int_t^{\infty} f\left(C_u, V_u\right) du \right]$$
$$f(C,V) \equiv \left(\frac{1-\gamma}{1-1/\psi}\right) V \left[\left(\frac{C}{\left[(1-\gamma)V\right]^{1/(1-\gamma)}} \right)^{1-1/\psi} - \rho \right]$$

One can show that

$$V_t = \frac{W_t^{1-\gamma}}{1-\gamma} J_t^{\frac{1-\gamma}{1-\psi}} \tag{B.1}$$

where J_t is the marginal value of wealth of the investor, which is an unknown function to solve for.

In the one-investor M-tree one-good case, J_t is governed in part by the Hamilton-Jacobi-Bellman equation of the investor. After plugging optimality conditions, it reads

$$\begin{aligned} 0 = & \left(\frac{1}{1-1/\psi}\right)(-\rho) + \left(\frac{1}{\psi-1}\right) J_t \\ & + r_t + \frac{\gamma}{2} \left(\sum_{j=0}^{M-1} w_{j,t} \sigma_{R_j,t} \right)^T \left(\sum_{j=0}^{M-1} w_{j,t} \sigma_{R_j,t} \right) \\ & + \left(\frac{1}{1-\psi}\right) \mu_{J,t} + \frac{1}{2} \left(\frac{1}{1-\psi}\right) \left(\frac{\psi-\gamma}{1-\psi}\right) \sigma_{J,t}^T \sigma_{J,t} \end{aligned} \tag{B.2}$$

where $\mu_{J,t}$ and $\sigma_{J,t}$ are the Geometric drift and diffusion of J_t:

$$\frac{dJ_t}{J_t} = \mu_{J,t} dt + \sigma_{J,t}^T d\vec{z}_t$$

From Itô's Lemma:

$$\mu_{J,t}J_t = \sum_{j=0}^{M-1} \frac{\partial J_t}{\partial Y_{j,t}} \mu_{Y_j,t} Y_{j,t} + \frac{1}{2} \sum_{j=0}^{M-1} \sum_{k=0}^{M-1} \frac{\partial^2 J_t}{\partial Y_{j,t} \partial Y_{k,t}} \sigma_{Y_j}^T \sigma_{Y_k} Y_{j,t} Y_{k,t} \tag{B.3}$$

$$\sigma_{J,t}J_t = \sum_{j=0}^{M-1} \frac{\partial J_t}{\partial Y_{j,t}} \sigma_{Y_j} Y_{j,t}$$

With $L > 1$ goods, all those expressions also involve the price of goods $p_{j,t}$, which are functions of the relative supply of their respective trees, as well as their (geometric) drifts and diffusion, $\mu_{p_j,t}, \sigma_{p_j,t}$.

B.2. Two investors, two trees, two goods

Details, economic intuitions, and applications of this case are presented at length in Sauzet (2021b,c). As emphasized there and in the main text (Part 3.3), the equilibrium can be recast as a stationary recursive Markovian equilibrium with state variables $X_t \equiv (x_t, y_t)' \in [0,1]^2$, where x_t is the wealth share of investor A, and y_t is the relative supply of good 1. They are defined as

$$x_t \equiv \frac{W_t^A}{W_t^A + W_t^B} \quad ; \quad y_t \equiv \frac{Y_{1,t}}{Y_{1,t} + Y_{2,t}} \tag{B.4}$$

All variables of interest can be expressed as a function of X_t, and a set $\mathcal{G}$ of unknown functions of X_t, $g : [0,1]^2 \to \mathcal{D}^g \subseteq \mathbb{R}$:

$$\mathcal{G} \equiv \{J_t^A, J_t^B, F_{1,t}, F_{2,t}, q_t, w_{1,t}^A, w_{2,t}^A\} \tag{B.5}$$

where J_t^i are the marginal values of wealth of both investors, $F_{j,t}$ are the dividend yields on both assets, q_t is the relative price of good 2, and $w_{j,t}^A$ are the weights on assets 1 and 2 in the portfolios of investor A as a share of her wealth. The share of the bond in the portfolio of each investor, which is non-zero when investors have asymmetric preferences beyond the bias in consumption, is $b_t^i = 1 - w_{1,t}^i - w_{2,t}^i$.

The system $\mathcal{H}$ pinning down those unknown functions consists of coupled algebraic and second-order partial differential equations. It invoves, for instance, the Hamilton-

Jacobi-Bellman equations of both investors, asset pricing equations featuring the risk premia on both assets, optimal portfolio conditions, and other equilibrium conditions pinning the relative price of consumption goods. All those equations combine the unknown functions in $\mathcal{G}$ and the state variables in non-linear ways, so that they have to be solved jointly. I also add a few equations that help the convergence.

To illustrate, I now present some of the equations that are part of the sytem $\mathcal{H}$. The reader is referred to Sauzet (2021b) for details.

Hamilton-Jacobi-Bellman equations For investor A, J_t^A satisfies

$$\begin{aligned}0 = &\left(\frac{1}{\psi^A - 1}\right) P_t^{A1-\psi^A} J_t^A - \left(\frac{1}{1-1/\psi^A}\right)\rho^A + r_t + \frac{\gamma^A}{2}\left(w_{1,t}^A \sigma_{R_1,t} + w_{2,t}^A \sigma_{R_2,t}\right) \\ &+ \left(\frac{1}{1-\psi^A}\right)\mu_{J^A,t} + \frac{1}{2}\left(\frac{1}{1-\psi^A}\right)\left(\frac{\psi^A - \gamma^A}{1-\psi^A}\right)\sigma_{J^A,t}^T \sigma_{J^A,t} \\ &+ \left(\frac{\delta^A}{1-\delta^A}\right) F_{1,t}\left(\frac{z_t}{x_t}\right) + w_{1,t}^B \left(\frac{1-x_t}{x_t}\right)\tau^B F_{1,t}\end{aligned}$$

where $\mu_{J^A,t}, \sigma_{J^A,t}$ are the geometric drift and diffusion terms of J_t^A

$$\frac{dJ_t^A}{J_t^A} \equiv \mu_{J^A,t}dt + \sigma_{J^A,t}^T d\vec{Z}_t \tag{B.6}$$

$$\mu_{J^A,t} \equiv \frac{J_{x,t}^A}{J_t^A} x_t \mu_{x,t} + \frac{J_{y,t}^A}{J_t} y_t \mu_{y,t} + \frac{1}{2}\frac{J_{xx,t}^A}{J_t^A} x_t^2 \sigma_{x,t}^T \sigma_{x,t} + \frac{1}{2}\frac{J_{yy,t}^A}{J_t^A} y_t^2 \sigma_{y,t}^T \sigma_{y,t} + \frac{J_{xy,t}^A}{J_t^A} x_t y_t \sigma_{x,t}^T \sigma_{y,t} \tag{B.7}$$

$$\sigma_{J^A,t} \equiv \frac{J_{x,t}^A}{J_t^A} x_t \sigma_{x,t} + \frac{J_{y,t}^A}{J_t^A} y_t \sigma_{y,t}$$

For investor B, J_t^B satisfies a similar Hamilton-Jacobi-Bellman equation.

Risk premia of assets 1 and 2

$$\begin{aligned}\mu_{R_1,t} - r_t &= \gamma_t \sigma_{R_1,t}^T \left\{ z_t \sigma_{R_1,t} + (1-z_t)\sigma_{R_2,t} \right\} \\ &- \gamma_t \sigma_{R_1,t}^T \left\{ x_t \left(\frac{1}{\gamma^A}\right)\left(\frac{1-\gamma^A}{1-\psi^A}\right)\sigma_{J^A,t} + (1-x_t)\left(\frac{1}{\gamma^B}\right)\left(\frac{1-\gamma^B}{1-\psi^B}\right)\sigma_{J^B,t} \right\} \\ &+ \gamma_t \left(\frac{1-x_t}{\gamma^B}\right)\tau^B F_{1,t}\end{aligned} \tag{B.8}$$

$$\begin{aligned}\mu_{R_2,t} - r_t &= \gamma_t \sigma_{R_2,t}^T \left\{ z_t \sigma_{R_1,t} + (1-z_t)\sigma_{R_2,t} \right\} \\ &- \gamma_t \sigma_{R_2,t}^T \left\{ x_t \left(\frac{1}{\gamma^A}\right)\left(\frac{1-\gamma^A}{1-\psi^A}\right)\sigma_{J^A,t} + (1-x_t)\left(\frac{1}{\gamma^B}\right)\left(\frac{1-\gamma^B}{1-\psi^B}\right)\sigma_{J^B,t} \right\} \\ &+ \gamma_t \left(\frac{x_t}{\gamma^A}\right)\tau^A F_{2,t}\end{aligned} \tag{B.9}$$

where $\gamma_t \equiv \left(\frac{x_t}{\gamma^A} + \frac{1-x_t}{\gamma^B}\right)^{-1}$ is the wealth-weighted risk aversion, and

$$\begin{aligned}dR_{j,t} &\equiv \mu_{R_j,t}dt + \sigma_{R_j,t}^T d\vec{Z}_t \\ \mu_{R_j,t} &\equiv F_{j,t} + \mu_{p_j,t} + \mu_{Y_j} + \sigma_{p_j,t}^T \sigma_{Y_j} - \mu_{F_j,t} + \sigma_{F_j,t}^T \sigma_{F_j,t} - \left(\sigma_{p_j,t} + \sigma_{Y_j}\right)^T \sigma_{F_j,t} \\ \sigma_{R_j,t} &\equiv \sigma_{p_j,t} + \sigma_{Y_j} - \sigma_{F_j,t}\end{aligned} \tag{B.10}$$

Goods prices

$$q_t = S_t^{1/\theta}\left(\frac{y_t}{1-y_t}\right)^{1/\theta} \tag{B.11}$$

where

$$S_t = \frac{(1-\alpha^A)P_t^{A\theta-\psi^A} J_t^A x_t + (1-\alpha^B)P_t^{B\theta-\psi^B} J_t^B (1-x_t)}{\alpha^A J_t^A x_t P_t^{A\theta-\psi^A} + (1-\alpha^B)P_t^{B\theta-\psi^B} J_t^B (1-x_t)}$$

Using the defintion of the numéraire, prices follow

$$p_{1,t} = \left(a + (1-a)q_t^{1-\theta}\right)^{1/(\theta-1)} \tag{B.12}$$

$$p_{2,t} = p_{1,t}q_t = \left(aq_t^{\theta-1} + (1-a)\right)^{1/(\theta-1)} \tag{B.13}$$

$$P_t^i = \left[\alpha^i p_{1,t}^{1-\theta} + (1-\alpha^i)p_{2,t}^{1-\theta}\right]^{1/(1-\theta)} \tag{B.14}$$

$$\mathcal{E}_t = P_t^B / P_t^A \tag{B.15}$$

Portfolios of both investors

$$\begin{pmatrix} w_{1,t}^A \\ w_{2,t}^A \end{pmatrix} = \frac{1}{\gamma^A} \left(\Sigma_t^T \Sigma_t\right)^{-1} \left\{ \begin{pmatrix} \mu_{R_1,t} - r_t \\ \mu_{R_2,t} - r_t - \tau^A F_{2,t} \end{pmatrix} + \left(\frac{1-\gamma^A}{1-\psi^A}\right) \Sigma_t^T \left(\frac{J_{x,t}^A}{J_t^A} x_t \sigma_{x,t} + \frac{J_{y,t}^A}{J_t^A} y_t \sigma_{y,t} \right) \right\}$$
$$b_t^A = 1 - w_{h,t}^A - w_{f,t}^A \tag{B.16}$$

$$\begin{pmatrix} w_{1,t}^B \\ w_{2,t}^B \end{pmatrix} = \frac{1}{\gamma^B} \left(\Sigma_t^T \Sigma_t\right)^{-1} \left\{ \begin{pmatrix} \mu_{R_1,t} - r_t - \tau^B F_{1,t} \\ \mu_{R_2,t} - r_t \end{pmatrix} + \left(\frac{1-\gamma^B}{1-\psi^B}\right) \Sigma_t^T \left(\frac{J_{x,t}^B}{J_t^B} x_t \sigma_{x,t} + \frac{J_{y,t}^B}{J_t^B} y_t \sigma_{y,t} \right) \right\}$$
$$b_t^B = 1 - w_{1,t}^B - w_{2,t}^B \tag{B.17}$$

where $\Sigma_t \equiv \begin{bmatrix} \sigma_{R_1,t} & \sigma_{R_2,t} \end{bmatrix}$.

Again, the reader is referred to Sauzet (2021b) for details.