

Computing equilibria in dynamic stochastic macro-models with heterogeneous agents*

Johannes Brumm

DBF, University of Zurich
johannes.brumm@gmail.com

Felix Kubler

DBF, University of Zurich
and Swiss Finance Institute
fkubler@gmail.com

Simon Scheidegger

DBF, University of Zurich
and Hoover Institution, Stanford
simon.scheidegger@gmail.com

December 30, 2015

*This paper was prepared for the 11th World Congress of the Econometric Society held in Montreal in 2015. We thank Dirk Krueger and Tony Smith for their comments. Johannes Brumm and Felix Kubler gratefully acknowledge financial support from the ERC. Felix Kubler and Simon Scheidegger gratefully acknowledge financial support from PASC. This work was also supported by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID s555. Additionally, we acknowledge CPU time granted on the University of Zurich's "Piz Dora" HPC cluster.

1 Introduction

Discrete-time, infinite-horizon, general equilibrium models are routinely used in macroeconomics and in public finance for exploring the quantitative features of model economies and for counterfactual policy analysis. With the development of powerful desktop computers, economists have started to use modern numerical methods for integration, interpolation, and for solving nonlinear systems of equations. Depending on the exact specification of the model—for example, whether there is one representative agent or several agents, whether agents are finitely lived or infinitely lived, or whether there is uncertainty in the model or not—there are a variety of computational methods for approximating equilibria numerically. This paper focuses on computational methods for stochastic equilibrium models with heterogeneous agents and aggregate uncertainty where the welfare theorems fail and the equilibrium allocation cannot be decentralized by a simple (convex) social planner problem. These could be models with overlapping generations (as in, e.g., Krueger and Kubler (2006), Favilukis et al. (2010), or Harenberg and Ludwig (2014)), models with heterogeneous producers (as in, e.g., Khan and Thomas (2013) or Bloom et al. (2012)), or models with infinitely lived heterogeneous consumers (as in, e.g., Bhandari et al. (2013), Brumm et al. (2015), Chien et al. (2011), Krueger et al. (2015), or McKay and Reis (2013)).

There are many excellent surveys on the computation of equilibria in these models and we will discuss the most popular methods briefly below. Instead of comparing those methods in detail, the main part of this paper focuses on a particular high-performance computing (HPC) approach for solving models with large heterogeneity. This approach was first introduced in Brumm and Scheidegger (2014) and we will expand it in this paper to tackle models with overlapping generations and with idiosyncratic risk.

Our approach makes use of two recent developments in scientific computing. First, advances in numerical analysis enable researchers to approximate very-high-dimensional functions. Employing standard discretization methods for the domain of such functions is computationally infeasible, as these approaches yield too many gridpoints at which the functions have to be evaluated. Starting with a one-dimensional discretization scheme that employs N gridpoints, a straightforward extension to d dimensions by a tensor product construction would lead to N^d gridpoints, encountering the so-called curse of dimensionality (see Bellman (1961)). The exponential dependence of the overall computational effort on the number of dimensions is a prohibitive obstacle to the numerical treatment of high-dimensional problems. Sparse grids, on the other hand, are able to alleviate this curse of dimensionality by reducing the number of gridpoints by orders of magnitude with only slightly deteriorated accuracy if the function to be interpolated is sufficiently smooth (see, e.g., Bungartz and Griebel (2004) and references therein). The main reason why these methods can be successfully applied to economic models is that for many economic applications the assumption of bounded mixed derivatives of value and policy functions is naturally satisfied—a prerequisite for the theory

of sparse grids. However, there is also a wide range of economic applications where the functions to be interpolated do not meet the regularity conditions mentioned above, but instead have steep gradients or non-differentiabilities. In such cases, ordinary sparse grids may be very inefficient in providing a good approximation. One effective way of overcoming this problem is to adaptively refine the sparse grid in regions with high function variation and spend fewer points in regions of low variation—thereby imposing a second layer of sparsity on top of the sparse grid (see, e.g., Pflüger (2010) or Ma and Zabararas (2009), and references therein). Brumm and Scheidegger (2014) show how adaptive grids can be applied in models with occasionally binding constraints where policy functions exhibit non-differentiabilities. In this paper we consider economies without occasionally binding constraints and show that both standard and adaptive sparse grid methods allow researchers to analyze models with up to 60 continuous state variables. We argue that projection methods can be fruitfully employed both for models with finitely many agents and for Bewley-style models with a continuum of ex ante identical agents within each generation.

Second, we show that the classes of dynamic models mentioned above are ideal candidates to be solved by high-performance computing—that is to say, by the efficient use of modern supercomputers (see, e.g., Dongarra and van der Steen (2012)), as they allow solution procedures that are naturally parallelizable. Such systems are nowadays not restricted to their traditional user communities such as nuclear physics; but are opening up to emerging domains like computational economics. It is clear that not everyone needs to access such high-end systems. Nevertheless, the computational economics community needs to start thinking more carefully about numerical parallelism. The trends in hardware design are such that in the near future even ordinary desktops will host dozens of general purpose processors (possibly combined with co-processors and/or GPUs) making the use of parallelization techniques important even for standard users.

When it comes to stochastic equilibrium models, it turns out that in solving for the unknown coefficients of the approximating functions one can typically employ a time-iteration method that naturally divides the (often intractably large) problem into small subproblems that can be solved independently. In each iteration step one has to solve a medium-sized system of nonlinear equations (around 60 equations in 60 unknowns) for each point in the grid. Fortunately, these tasks are fully independent from each other and can thus be solved in parallel by distributing them via the Message Passing Interface (MPI; see, e.g., Skjellum et al. (1999)) among different compute nodes and via thread building blocks (TBB; see, e.g., Reinders (2007)) within nodes. A non-trivial complication comes from the fact that, when searching for the solution to the system of equations at a given gridpoint, the algorithm has to frequently interpolate the function computed in the previous iteration step. These interpolations take up 99 percent of the computation time needed to solve the equation system. As they have a high arithmetic intensity—that is to say, many arithmetic operations are performed for each byte of memory transfer and access—they are perfectly suited

for graphics processor units (GPUs) (see Brumm et al. (2015) and references therein). We therefore offload parts of the interpolation tasks to GPUs. This parallelization scheme enables us to make efficient use of modern hybrid HPC facilities, featuring CPU compute nodes with attached GPUs. The performance of these machines nowadays reaches multiple petaflops (see Dongarra and van der Steen (2012) and <http://www.top500.org/>).¹

To demonstrate the ability of the computational method presented in this paper we consider several versions of a standard overlapping generations (OLG) model with neoclassical production. Other stochastic models with infinitely lived consumers or with firms can often be analyzed using similar methods but some of the statements we make are particular to the OLG model. Infinite-horizon general equilibrium models with overlapping generations have many interesting implications that are absent from the standard Arrow–Debreu model. Under certainty, there are well-established methods for approximating equilibria numerically and the model has been fruitfully applied in macroeconomics and public finance (see, e.g., Auerbach and Kotlikoff (1987)). Under uncertainty, steady-state equilibria do not exist in these models, and even when the exogenous aggregate shock can take only finitely many values the equilibrium allocations do not in general have finite support. This feature makes it difficult to approximate equilibria with many agents of different ages and with aggregate uncertainty. These models are therefore an ideal example for demonstrating the advantages of our computational approach.

We consider the simplest possible version of an overlapping generations model with stochastic production: there is a single agent per generation and the only asset available for trade is risky capital. We show that with the help of adaptive sparse grids and high-performance computing the simple collocation method from Krueger and Kubler (2004) can be extended to handle models where agents live for 60 periods—that is to say, models that can be calibrated to yearly data. In the presence of disaster shocks fairly large shifts in the intergenerational wealth distribution can occur endogenously and can have large effects on aggregate variables. The possibility of large negative shocks that occur with small probability has been made popular by Barro (2006), although in a different framework. Our sparse grid collocation method is ideally suited to be applied to this problem. We also show how the problem scales nicely with the number of processors. Moreover, we illustrate the performance of the method with respect to the refinement criterion used in the adaptation procedure.

We also examine an OLG model with a continuum of ex ante identical agents within each generation that features, in addition to aggregate shocks, idiosyncratic shocks. It is now well understood that uninsurable idiosyncratic risk often plays an important role in macroeconomic dynamics (see,

¹The Swiss National Supercomputer Centre’s “Piz Daint” Cray XC30, which is used in the numerical experiments in Section 6 below consists of Intel Xeon E5 processors with NVIDIA Tesla K20X GPUs attached; its peak performance is 7.7 petaflops. It can perform, in a single day, the same number of floating point operations as an off-the-shelf laptop can carry out in a millennium.

e.g., Krueger et al. (2015) for a survey). In the standard Bewley model there is a continuum of infinitely lived, ex ante identical agents that differ ex post by the realization of their idiosyncratic shocks. In this model, the wealth distribution is an infinite-dimensional object and standard collocation methods do not seem applicable. Krusell and Smith (1997, 1998) have the seminal insight to focus on what we call “approximately self-confirming equilibria”. They postulate that along the equilibrium path agents’ forecasts of future prices are simple functions of some key moments of the wealth distribution. In fact, in the simple model they consider, the mean capital holding turns out to be sufficient for achieving very accurate forecasts. Subsequently, Storesletten et al. (2007) and Harenberg and Ludwig (2014), among others, have applied this method in a framework with overlapping generations. For many purposes (in particular for the original contribution in Krusell and Smith (1998)) the method is well suited to approximate equilibrium numerically. However, for others it does not. In particular, in a model with overlapping generations, shifts in the intergenerational wealth distribution have large effects on aggregate outcomes, and if large shocks to fundamentals lead to large shifts in asset holdings across generations, moments of the wealth distribution are not well suited to describe the state of the economy. Moreover, there is no easy way of using this method to investigate the effects of unanticipated policy changes: even if the method provides good approximations for the (long-run) behavior in the two different regimes, there is no straightforward way of approximating the policies along the transition path. Using ideas from Reiter (2010) and from Algan et al. (2014) we suggest that these challenges can be tackled with an “extended” projection method where the wealth distribution within each generation is approximated by a simple low-dimensional scheme. However, the method can only be employed for models with a (relatively) small number of generations. In order to tackle large-scale models, the adaptive sparse grid environment should be coupled with high-dimensional model representation techniques (HDMR; see, e.g., Rabitz and Alis (1999) and Ma and Zabaras (2010)). The combination of these two methods—a linear combination of lower-dimensional (adaptive sparse grid) function approximations that focus on the important dimensions—offers the promise of successfully addressing very-high-dimensional economic problems. How to couple HDRM and adaptive sparse grids in an effective way to address economic problems is the subject of our ongoing research.

Throughout all the examples, we consider a version of the model with a small finite number of shocks. We thus avoid the important additional computational challenge of high-dimensional integration. How to optimally combine sparse grid interpolation with sparse grid integration is the subject of further research; see Brumm and Scheidegger (2014) for some discussion on the issue. In the papers surveyed in Kollmann et al. (2011) the issue of high-dimensional integration is discussed in some detail.

The rest of this paper is organized as follows. In Section 2 we briefly review some of the existing literature. In Section 3 we introduce a general OLG model. In Section 4 we describe the basic

idea behind collocation methods and explain how the method can be used both for models with a continuum of agents and for models with finitely many agents. In Section 5 we introduce sparse grids. In Section 6 we explain in detail how the model without idiosyncratic shocks can be solved and we present the computational and economic results obtained.

2 A brief literature review

There exist several survey papers that review the advantages and disadvantages of existing computational methods in detail. Judd et al. (2003) provide an in depth survey on the development of computational methods in economics until the late 1990's. In particular they explain many methods for the computation of equilibria in deterministic models and in stochastic models with a representative agent. For models with finitely many agents, Maliar and Maliar (2014) and Kollmann et al. (2011) provide excellent and very detailed overviews of recent advances in computational methods. For models with a continuum of ex ante (but not ex post) identical agents, Den Haan (2010), Algan et al. (2014), and—for the case of heterogeneous firms—Terry (2014) provide excellent reviews of existing algorithms.

Which computational strategy should be employed obviously depends on a variety of considerations. A first important distinction to be made is whether one models a “distribution economy” with a continuum of agents and hence with an infinite-dimensional wealth distribution or an economy with a finite number of agents. In the latter case the choice of the “right” computational method would depend on the number of economic agents and the dimension of the state space in the recursive formulation of the model. Other considerations would be whether there are occasionally binding constraints, whether there are large shocks to fundamentals, which can be expected to generate large movements in the wealth distribution, and whether—in general—nonlinearities are expected to play an important role in the economic model and for the assumed values of parameters. Most importantly there is an important trade-off between high-productivity computing with high accuracy and high reliability of the solution on one side and high-productivity coding on the other. For many purposes it might suffice to employ linearization, or perturbation techniques to obtain a relatively accurate solution at a very low cost. The software package DYNARE (see, e.g., Juillard (2003)) provides the user with a simple and user-friendly framework for this. Several recent papers use local techniques repeatedly to obtain solutions that are globally accurate (see, e.g., Evans (2015) and Proehl (2015)).

For asset pricing models with a few heterogeneous agents the method from Dumas and Lyasoff (2012) proves very convenient and can be used for a variety of applications. Models with agents who face trading constraints might be best solved with tessellation techniques (see, e.g., Brumm and Grill (2014)); models with a moderate number of heterogeneous agents can be solved with a Galerkin approach as in Pichler (2011), with a simulation-based approach as in Maliar et al. (2011),

or with a collocation method as in Krueger and Kubler (2004). Maliar and Maliar (2014) provide an excellent discussion of these methods, and MATLAB code for versions of the method from Maliar et al. (2011). Hasanhodzic and Kotlikoff (2013) apply the method to a model with overlapping generations, similar to the one considered in this article.

The Smolyak method in Krueger and Kubler (2004) is embedded in a Bayesian estimation framework by Winschel and Kraetzig (2010) and extended in Judd et al. (2014) by introducing substantial speedups in interpolation and the evaluation of the Smolyak polynomials. These papers are the most closely related to the basic approach used in this paper. The main difference is that we use sparse grids with hierarchical basis functions, which makes it possible to use adaptive sparse grids and has important advantages regarding the data structure for parallel computing. These features allow us, in contrast to the abovementioned Smolyak methods, to obtain global solutions that feature high local resolution where needed—for instance when functions exhibit non-differentiabilities or steep gradients.

The computation of equilibria in models with a continuum of heterogeneous agents (distribution economies) and aggregate shocks is less well understood. The problem can often be tackled following the seminal approach of Krusell and Smith (1997, 1998). In this approach it is postulated that the future evolution of the wealth distribution can be approximated well by a function that depends only on a few moments of the current wealth distribution. Agents forecast prices using this approximation. In an approximately self-confirming equilibrium the agents make only small forecasting errors.

It turns out that a method that uses only the first moment of the wealth distribution provides reasonable approximations in many models, both those with infinitely lived agents and those with overlapping generations. In asset pricing models without a physical state, it is useful to include either prices (as in Storesletten et al. (2007)) or past aggregate shocks as variables that forecast future prices. Using only past shocks (and no information about the wealth distribution) obviously simplifies the problem hugely since forecasts can be summarized in a finite vector. This approach has been fruitfully used by Chien and Lustig (2010) and Chien et al. (2011). While in these papers the authors explain their method in the framework of a multiplier approach and forecast a crucial moment of the cross-sectional distribution of multipliers by lagged shocks, it is easy to verify that there is a one-to-one map between this moment and equilibrium prices.

Nevertheless, as Algan et al. (2014) point out, depending on the situation, more sophisticated—yet also more complicated—computational methods might be more suitable. Young (2010) shows that the basic idea of Krusell and Smith (1997, 1998) can be employed more efficiently by replacing the simulation step in their method by a non-stochastic algorithm. Algan et al. (2010) and Reiter (2010) extend the basic ideas in Krusell and Smith (1997, 1998) and Young (2010) and allow for more flexible functional forms for both the wealth distributions and the transition functions. Promising

approaches have also been developed by Den Haan and Rendahl (2010), Reiter (2009), Mertens and Judd (2013), and Sager (2014). These approaches are ideally suited for models with infinitely lived agents but it is less clear how to apply them to economies with overlapping generations.

There are many more interesting papers (many of them cited in the surveys we mention above) that introduce useful methods for solving dynamic stochastic models with heterogeneous agents. Rather than discuss them all, we will mention some in our explanation of our methods below and make some comparisons.

3 Model

We consider an Aiyagari–Bewley-style overlapping generations model with incomplete financial markets. The model is a simple extension of the model in Krueger and Kubler (2006) to economies with a continuum of heterogeneous agents.

3.1 The physical economy

Time is indexed by $t \in \mathbb{N}_0$. Aggregate shocks z_t realize in a finite set \mathbf{Z} , and follow a first-order Markov process with transition probability $\pi(z'|z)$. A history of aggregate shocks up to some date t is denoted by $z^t = (z_0, z_1, \dots, z_t)$. At each date-event a continuum of ex ante identical agents enter the economy and live for A periods. Within each cohort agents differ ex post by the realization of their idiosyncratic shocks. We assume that idiosyncratic shocks follow a Markov chain and have support in a finite set \mathbf{Y} . We denote by $\eta(y'|y)$ the conditional probability of y' given y . For simplicity we assume that at $a = 1$ all agents are assumed to be at an initial shock y_1 and we use $\eta(y^a)$ to denote the probability of a history y^a of idiosyncratic shocks until age a . Note that in this setup $\eta(y^a)$ is time invariant and does not depend on the aggregate shock.

At a given date-event z^t we can uniquely identify agents who consume at that date-event by their age and history of idiosyncratic shocks, (a, y^a) . For each (a, y^a) there is a continuum of identical agents, but we will refer to them as “one agent” since at z^t they all take identical actions. We often simply write y^a since the age is implicit in the length of the history of shocks. We denote the set of all these “agents” by $\mathbf{A} = \{y^a : 1 \leq a \leq A, y^a \in \mathbf{Y}^a\}$ and the set of all agents except for generation i by $\mathbf{A}_{-i} = \mathbf{A} \setminus \{y^a : a = i, y^a \in \mathbf{Y}^a\}$.

At each z^t we denote the fraction of agents of type h that have an idiosyncratic shock history y^a by $\nu_{y^a}(z^t)$. We assume that the joint distribution of idiosyncratic shocks within a type ensures that at each history of aggregate shocks, z^t , for any $y^a \in \mathbf{Y}^a$ the fraction of agents with history $y^a = (y_1, \dots, y_a)$ is $\nu_{y^a}(z^t) = \eta(y^a)$ (see, e.g., Feldman and Gilles (1985) for a simple construction of such a process). This allows us to focus on equilibria where prices and aggregate quantities only depend on the history of aggregate shocks, z^t . In a slight abuse of language we will refer to these

as nodes of the event tree.

At node z^t an agent y^a has a non-negative labor endowment $l_{y^a}(z^t) = l^a(y_a, z_t)$, which depends on his or her idiosyncratic shock, y_a , and possibly on the aggregate shock, z_t . The price of the consumption good at each date-event is normalized to one and at each date-event z^t the household supplies its labor endowment inelastically for a market wage $w(z^t)$.

Individuals have preferences over consumption streams representable by a recursive utility function (see Kreps and Porteus (1978) and Epstein and Zin (1989)). At node z^t all agents of shock history y^a have a value function

$$U_{y^a}(z^t) = \left\{ [c_{y^a}(z^t)]^\rho + \beta_a \left[\sum_{z_{t+1}} \pi(z_{t+1}|z_t) \sum_{y_{a+1}} \eta(y_{a+1}|y_a) (U_{y^{a+1}}(z^{t+1}))^\sigma \right]^{\frac{\rho}{\sigma}} \right\}^{\frac{1}{\rho}}, \quad (1)$$

where $\frac{1}{1-\rho}$ is the intertemporal elasticity of substitution and $1 - \sigma$ measures the risk aversion of the consumer with respect to atemporal wealth gambles. We allow the discount factor to depend on age and denote it by $\beta_a > 0$. We assume $\sigma < 1$ and $\rho < 1, \rho \neq 0$. Note that if $\rho = \sigma$, then households have standard constant relative risk aversion (CRRA) expected utility, with a CRRA of $1 - \sigma$ if the final continuation utility function is given by $U_{y^A}(z^t) = c_{y^A}(z^t)$, which we assume.

To make the notation and the model as simple as possible, we assume that agents can only trade risky capital. There is a storage technology that uses one unit of the consumption good today to produce one unit of the capital good for next period. We denote the investment of household y^a in this technology by $k_{y^a}(z^t)$. At time t the household sells its capital goods accumulated from last period, $k_{y^a}(z^{t-1})$, to the firm for a market price $1 + r(z^t) > 0$.

The budget constraint of household y^a at node z^t therefore reads as

$$c_{y^a}(z^t) + k_{y^a}(z^t) = (1 + r(z^t))k_{y^a}(z^{t-1}) + l_{y^a}(z^t)w(z^t), \quad (2)$$

where $k_{y^0} = k_{y^A} = 0$.

To start off the economy we assume that in period zero there are $A - 1$ households of ages $a = 1, \dots, A$ that enter the period with given capital holdings $k_{-1}(z_0), \dots, k_{-A+1}(z_0)$. We sometimes refer to this as the “initial conditions”. For simplicity we do not consider initial conditions where there is a non-degenerate distribution of capital within some generation that is alive at $t = 0$. This would just introduce some additional notation and add nothing to the economics.

3.2 Firms

There is a single representative firm, which in each period t uses labor and capital to produce the consumption good according to a constant-returns-to-scale production function $f(K, L; z_t)$. Since firms make decisions on how much capital to buy and how much labor to hire after the realization

of the shock z_t they face no uncertainty and simply maximize current profits.²

In our quantitative work below we will always use the following parametric form for the production function:

$$f(K, L; z_t) = \xi(z_t)K^\alpha L^{1-\alpha} + K(1 - \delta(z_t)), \quad (3)$$

where $\xi(\cdot)$ is the stochastic shock to productivity and where $\delta(\cdot)$ can be interpreted as the (possibly) stochastic depreciation rate.

Note that since firms maximize profits, the rate of return on capital, $1 + r(z_t)$, will always equal the marginal product of capital, $f_K(K, L, z_t)$, and the wage, $w(z^t)$, will equal the marginal product of labor.

3.3 Markets

In this simple economy the only markets are spot markets for consumption, labor, and capital, all of which are assumed to be perfectly competitive. To simplify the exposition, we do not include a bond market in our basic model. In Section 6.4 below we explain how the presence of a risk-free bond can be handled using the same computational strategy.

For given initial conditions $z_0, (k_i(z_0))_{i=-A+1}^{-1}$ a competitive equilibrium is a collection of choices for households $(c_{y^a}(z^t), k_{y^a}(z^t))_{y^a \in \mathbf{A}}$, for the representative firm $\{K(z^t), L(z^t)\}$, and for prices $\{r(z^t), w(z^t)\}$ such that households and the firm maximize and markets clear: for all t, z^t

$$L(z^t) = \sum_{a=1}^A \sum_{y^a \in \mathbf{Y}^a} \nu(y^a) l_{y^a}(z^t) \quad (4)$$

$$K(z^t) = \sum_{a=1}^A \sum_{y^a \in \mathbf{Y}^a} \nu(y^a) k_{y^a}(z^{t-1}) \quad (5)$$

By Walras's law, market clearing in the labor and capital markets imply market clearing in the consumption goods market in general equilibrium.

Note that in our construction, there are strictly speaking only finitely many different agents active at all nodes. As long as one considers a finite number of idiosyncratic shocks this is naturally the case in overlapping generation models (unless one assumes that agents enter the economy at age $a = 1$ with a continuous wealth distribution). However, the number of agents will typically be so large that it is useful to use methods from the case of infinitely lived agents to perform computations. We will clarify this below.

²We assume that households cannot convert capital goods back into consumption goods at the beginning of the period. This assumption is necessary to prevent households from consuming the capital at the beginning of the period instead of selling it to the firm in states where the net return to capital is negative.

4 Projection methods

Projection methods (most importantly collocation methods and Galerkin’s method) are well-established numerical methods for the solution of functional equations and they are now a standard method for analyzing dynamic economies. Projection methods were introduced to economics by Judd (1992) and a description of these methods can now be found in most textbooks on numerical methods in economics (see, e.g., Judd (1998) or Heer and Maussner (2009)). As Krueger and Kubler (2004) point out, these methods are ideally suited to approximating equilibria in overlapping generations models with aggregate uncertainty and a finite (moderate) number of agents. In this paper, we use sparse grids and parallel computing in order to employ collocation methods to solve models with up to 60 agents. We also argue that projection methods can be used to solve Bewley-style models with a continuum of ex ante identical agents which differ ex post by the realization of idiosyncratic shocks (we refer to these models as “distribution economies”). Following an idea of Reiter (2010), we develop a collocation method to approximate equilibria in distribution economies with overlapping generations.

In this section we explain, in abstract terms, the basic methods and argue that, depending on the circumstances, it is advantageous to use projection methods rather than simulation-based approaches as developed in Krusell and Smith (1998) or in Maliar et al. (2011).

4.1 Economies with a finite number of agents

We consider the basic model where $A - 1$ heterogeneous agents are active at each date-event. Following Judd (1992) we use a projection method to approximate equilibria numerically. In order to do so we first need to describe equilibrium as a system of operator equations.

4.1.1 Functional Rational Expectations Equilibrium

We assume in this subsection that there are no idiosyncratic shocks. Therefore we can identify agents simply by his or her age, a . Our computational strategy searches for a recursive equilibrium where the distribution of capital holdings across agents constitutes a sufficient endogenous state, and where the endogenous state lies in a known compact set. Following Spear (1988) we call this “functional rational expectations equilibrium” (FREE). In our specification, we require the compact endogenous state space to be a hyperrectangle, also simply called a box. A natural endogenous state would be the vector of individual asset holdings. However, this turns out to be impractical in our setting. The lower bound of individuals’ equilibrium asset holdings is not guaranteed to be positive. The sum of all lower bounds being negative results in a negative aggregate capital stock and, thus, returns to capital and wages that are not well defined. Instead we choose as an endogenous state the aggregate capital stock together with the financial wealth of all generations except the youngest and

the oldest—the youngest have no financial wealth and the oldest simply consume all their financial wealth. The precise definition of a functional rational expectations equilibrium is as follows:

DEFINITION 1 A *FREE* consists of an $A - 1$ -dimensional box $\mathbf{B} \subset \mathbb{R}^{A-1}$, asset demand functions $k_a : \mathbf{Z} \times \mathbf{B} \rightarrow \mathbb{R}$, and value functions $v_a : \mathbf{Z} \times \mathbf{B} \rightarrow \mathbb{R}$, $i = 1, \dots, A - 1$, such that for all shocks $z \in \mathbf{Z}$ and all states $s = (s_1, \dots, s_{A-1}) \in \mathbf{B}$

$$\begin{aligned} & \bar{c}_a(z, s)^{\rho-1} + \beta_a \left[\sum_{z'} \pi(z'|z) (v_{a+1}(z', s'))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}-1} \\ & \sum_{z'} \pi(z'|z) (1 + r(z', s'_1)) v_{a+1}(z', s')^{\frac{\sigma}{\rho}-1} \bar{c}_{a+1}(z', s')^{\rho-1} = 0, \quad a = 1, \dots, A - 1 \end{aligned} \quad (6)$$

and

$$v_a(z, s) = \bar{c}_a(z, s)^\rho + \beta_a \left[\sum_{z'} \pi(z'|z) (v_{a+1}(z', s'))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}}, \quad a = 1, \dots, A - 1, \quad (7)$$

where $v_A(z, s) = \bar{c}_A(z, s)^\rho$, $s'_1 = \sum_{i=1}^{A-1} k_i(z, s)$,

$$s' = (s'_1, k_1(z, s)(1 + r(z', s'_1)), \dots, k_{(A-2)}(z, s)(1 + r(z', s'_1))) \in \mathbf{B}$$

is the state tomorrow, and

$$\begin{aligned} (1 + r(z, s_1)) &= f_K(s_1, \sum_{i=1}^A l^i(z), z) \\ w(z, s_1) &= f_L(s_1, \sum_{i=1}^A l^i(z), z) \\ \bar{c}_1(z, s) &= l^1(z)w(z, s_1) - k_1(z, s) \\ \bar{c}_i(z, s) &= s_i + l^i(z)w(z, s_1) - k_i(z, s) \text{ for } i = 2, \dots, A - 1 \\ \bar{c}_A(z, s) &= \left(s_1(1 + r(z, s_1)) - \sum_{i=2}^{A-1} s_i \right) + l^A(z)w(z, s_1). \end{aligned}$$

Note that in this definition the financial wealth of the oldest generation is given by aggregate financial wealth minus the financial wealth of all other generations. Since all first order conditions are necessary and sufficient it is clear that any FREE induces a competitive equilibrium in the sense of Definition 1.

Unfortunately, there are no guarantees that a FREE always exists. The existence of recursive equilibria in stochastic models with overlapping generations can only be guaranteed under very restrictive assumptions (see, e.g., Brumm and Kubler (2014)). Moreover, our computational strategy below assumes the existence of a FREE with continuous asset-policies. It is clear that these will not always exist. However, in the calibrations we use in the example below, there turn out to be ϵ -FREE—that is to say, smooth policy-functions that solve (6) and (7) with a small error $\epsilon > 0$. Kubler and Schmedders (2005) give a simple interpretation of how to relate them to exact equilibria.

4.1.2 Collocation

We approximate the unknown equilibrium asset-demand and value functions $k_i(z, \cdot)$, $v_i(z, \cdot)$, $i = 1, \dots, A-1$, $z \in \mathbf{Z}$ by piecewise multi-linear functions $\hat{k}_i(z, \cdot | \alpha^k)$, $\hat{v}_i(z, \cdot | \alpha^v)$ that are uniquely defined by finitely many coefficients α^k, α^v . In order to solve for the unknown coefficients, we require that the functional equation (6)–(7) holds exactly at M collocation points $\tilde{s} \in \mathbf{H} \subset \mathbf{B}$. We therefore transform the infinite-dimensional functional equation into the following finite-dimensional (nonlinear) system of equations in the $2 \cdot M \cdot S \cdot (A-1)$ unknown coefficients α^k, α^v .

We require that for all $z \in \mathbf{Z}$ and all $\tilde{s} \in \mathbf{H}$ we have

$$\begin{aligned} & \hat{c}_a(z, \tilde{s} | \alpha)^{\rho-1} + \beta_a \left[\sum_{z'} \pi(z' | z) (\hat{v}_{a+1}(z', s' | \alpha))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}-1} \\ & \sum_{z'} \pi(z' | z) (1 + r(z', s')) \hat{v}_{a+1}(z', s' | \alpha)^{\frac{\sigma}{\rho}-1} \hat{c}_{a+1}(z', s' | \alpha)^{\rho-1} = 0, \quad a = 1, \dots, A-1 \end{aligned}$$

and

$$\hat{v}_a(z, \tilde{s} | \alpha) = \hat{c}_a(z, \tilde{s} | \alpha)^\rho + \beta_a \left[\sum_{z'} \pi(z' | z) (\hat{v}_{a+1}(z', s' | \alpha))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}}, \quad a = 1, \dots, A-1, \quad (8)$$

where $\hat{v}_A(z, s | \alpha) = \hat{c}(z, s | \alpha)^\rho$, $s'_1 = \sum_{i=1}^{A-1} \hat{k}_i(z, \tilde{s})$, and

$$\begin{aligned} s' &= \left(s'_1, \hat{k}_1(z, \tilde{s})(1 + r(z', s'_1)), \dots, \hat{k}_{(A-2)}(z, \tilde{s})(1 + r(z', s'_1)) \right) \in \mathbf{B} \\ \hat{c}_1(z, s | \alpha) &= l^1(z)w(z, s_1) - \hat{k}_1(z, s; \alpha) \\ \hat{c}_i(z, s; \alpha) &= s_i + l^i(z)w(z, s_1) - \hat{k}_i(z, s; \alpha) \text{ for } i = 2, \dots, N-1 \\ \hat{c}_N(z, s; \alpha) &= \left(s_1(1 + r(z, s_1)) - \sum_{i=2}^{A-1} s_i \right) + l^i(z)w(z, s_1) \end{aligned}$$

and where the prices $1 + r$ and w are as above in the definition of FREE.

The main computational challenges are caused by the fact that the endogenous state space has dimension $A-1$. In a model where a period corresponds to a year and agents live (or are active in the economy) for 60 years, one therefore has to approximate 59-dimensional functions. In order to do so with some accuracy one needs a large number of collocation points, which can result in a very large nonlinear system in the unknown coefficients, α . We will explain in Section 6 how we solve this system after we have introduced sparse grid methods in Section 5. The sparse grid methods allow us to approximate very-high-dimensional functions with a reasonable number of points.

Clearly, the resulting functions \hat{k}, \hat{v} will not solve the functional equation (6)–(7) exactly at all $(s, z) \in \mathbf{B} \times \mathbf{Z}$. One could argue that a candidate solution is a good approximation if the maximal error in these equations over all of $\mathbf{B} \times \mathbf{Z}$ is small. For high dimensions this is obviously impossible to verify and one needs to find other ways to determine the quality of a candidate solution. We return to this issue below when we present details of our computational strategy and a concrete example.

4.2 A collocation method for distribution economies

In order to use projection methods for solving the model with idiosyncratic shocks we follow ideas from Reiter (2010) and Algan et al. (2010). The main difference between our approach and theirs is that we consider an OLG economy and keep track of each generation separately and they consider a model with infinitely lived agents. Moreover, both their approaches rely on simulation techniques in order to provide a link between the moments of the wealth distribution that are used as an endogenous state and the actual wealth distribution. We use the same object as a state that we later use to represent the wealth distribution, and hence do not need simulation at any point in the method.

We now assume that within each generation there is a continuum of agents that face idiosyncratic shocks. For simplicity we assume that the idiosyncratic shocks are i.i.d. within each generation and shocks are independent across different cohorts. This simplification is not strictly necessary for our approach but it simplifies notation and with persistence in the idiosyncratic shock our computational methods become infeasible rather quickly.

Obviously, the number of different agent types at each node, while finite, is so large that projection methods as explained above can no longer be used (even in the simplest case where the idiosyncratic shock can take just two different values there are more than 2^A agents). However, within each generation agents only differ by their asset holdings. We approximate this intragenerational wealth distribution by discretizing the inverse distribution function.

As above, we keep track of aggregate capital and describe the wealth distribution across individuals by their financial wealth. We allow for N different wealth levels for each generation $a = 2, \dots, A - 1$.³ We denote by $\omega_{a1}, \dots, \omega_{aN}$ the possible wealth levels used for the description of the wealth distribution within generation a . For concreteness and simplicity, assume that each ω_{ai} , $a = 2, \dots, A - 1$ is the average capital holding of all agents of age a that are richer than the $(i - 1)/N$ percentile of the wealth distribution yet poorer than the i/N percentile. This choice for representing the wealth distribution only works well in a model without borrowing constraints. Other methods should be employed if it can be expected that a large fraction of the population holds zero wealth.

The endogenous *aggregate* state is then given by

$$s = (K, (\omega_{a1}, \dots, \omega_{aN})_{a=2, A-1}).$$

We denote by $k_a(z, s, \theta)$ the savings function of an individual of age a given the aggregate state (z, s) , and the individual's cash-at-hand by θ . Since we assume that the individual is "atomless", there does not need to be any relation between s and θ .

³All agents of age $a = 1$ enter the economy with zero wealth and the distribution of wealth among agents of age $a = A$ is irrelevant. Moreover, since there are only Y possible wealth levels for agents of age $a = 2$ it is often convenient to treat this separately but for simplicity we do not do so here

As before we, postulate the existence of a box that always contains the endogenous state $(s, \theta) \in \mathbf{B} \subset \mathbb{R}^{N(A-2)+2}$. It is useful to denote by $\tilde{\mathbf{B}} \subset \mathbb{R}^{N(A-2)+1}$ the range of only the aggregate state, s . We need to determine asset demand functions $k_a : \mathbf{Z} \times \mathbf{B} \rightarrow \mathbb{R}$ and value functions $v_a : \mathbf{Z} \times \mathbf{B} \rightarrow \mathbb{R}$, $a = 1, \dots, A-1$, such that for all shocks $z \in \mathbf{Z}$ and all states $(s, \theta) \in \mathbf{B}$ for each $a = 1, \dots, A-1$,

$$\bar{c}(z, s, \theta)^{\rho-1} + \beta_a \left[\sum_{z'} \pi(z'|z) \sum_{y'} \eta(y') (v_{a+1}(z', s', \theta'(y', z'))) \right]^{\frac{\sigma}{\rho}} \\ \sum_{z'} \pi(z'|z) \sum_{y'} \eta(y') (1 + r(z', s')) v_{a+1}(z', s', \theta'(y', z'))^{\frac{\sigma}{\rho}-1} \bar{c}_{a+1}(z', s', \theta'(y', z'))^{\rho-1} = 0,$$

and

$$v_a(z, s, \theta) = \bar{c}_a(z, s, \theta)^{\rho} + \beta_a \left[\sum_{z'} \pi(z'|z) \sum_{y'} \eta(y') (v_{a+1}(z', s', \theta'(y', z'))) \right]^{\frac{\sigma}{\rho}},$$

with $\theta'(y', z') = k_a(z, s, \theta)(1 + r(z', s')) + l_{a+1}(y', z')w(z', s')$.

Since we assume that all agents of age $a = 1$ are identical, we can write $l_1(y, z) = l_1(z)$ as we have the following consumption functions:

$$\begin{aligned} \bar{c}_1(z, s, \theta) &= l_1(z)w(s, z) - k_1(z, s, \theta) \\ \bar{c}_i(z, s, \theta) &= \theta - k_i(z, s, \theta) \text{ for } i = 2, \dots, A-1 \\ \bar{c}_A(z, s, \theta) &= \theta. \end{aligned}$$

As above, for $a = A$ we set $v_A(z, s, \theta) = c_A(z, s, \theta)^{\rho}$.

The aggregate state evolves as follows:

$$s' = \left(\left(\sum_{i=1}^{A-1} \sum_{n=1}^N \frac{1}{N} \sum_{y \in \mathbf{Y}} \eta(y) k_i(z, s, \omega_n + w(z, s) l^i(y, z)) \right), (\omega'_{an})_{a=2, \dots, A-1, n=1 \dots N} \right) \in \tilde{\mathbf{B}},$$

where for each $a = 2, \dots, A-1$, ω'_{ai} is the average over financial wealth,

$$k_{a-1}(z, s, \omega_{a-1, n} + w(z, s) l^a(y, z)) (1 + r(z', s')),$$

among agents of that generation that are richer than the $(i-1)/N$ percentile of the wealth distribution yet poorer than the i/N percentile (for $a=2$ we only take one possible wealth level, $\omega_1 = 0$, since agents are assumed to enter the economy with no wealth). Finally, we have

$$\begin{aligned} (1 + r(z, s)) &= f_K(s_1, \sum_{i=1}^A \sum_{y \in \mathbf{Y}} \eta(y) l^i(y, z), z) \\ w(z, s) &= f_L(s_1, \sum_{i=1}^A \sum_{y \in \mathbf{Y}} \eta(y) l^i(y, z), z). \end{aligned}$$

Although slightly tedious to write out, this is a standard functional equation that can be solved by a collocation method. Note that we have explicitly two stages of approximations. First we

approximate the infinite-dimensional problem by a finite-dimensional one. Choosing sufficiently large N is presumably crucial for this approximation to be viable. Some knowledge about the actual intragenerational wealth distribution might be useful in making this approximation more accurate. Algan et al. (2010) and Reiter (2010) develop some methods for this but as we argue above they need to rely on simulations in order to do so. We present results for the simplest possible approach to this problem. Second we need to use collocation methods to approximate the solution to the functional equation. Here a large N (together with a large A) poses computational problems since it leads to a high-dimensional endogenous state space.

In conducting error analysis to assess the quality of a candidate solution, one can either just consider the error in the solution of the functional equation (which itself might only be a rough approximation of the actual recursive equilibrium), or one can try to also determine the quality of this first approximation.

Clearly this approach to solving Bewley-style models is much more demanding than the standard simulation approach pioneered by Krusell and Smith (1998). As we already mentioned in the introduction to this paper, it is often not a priori clear which method is superior. For many applications, simulation-based methods obviously have the huge advantage that they are easy to implement and not too costly to use. However, once one has a reliable suite of subroutines for a time-iteration collocation method, it can be easily adapted to also handle distribution economies as long as one is content with a rather rough approximation to the intragenerational wealth distribution.

5 Sparse Grids

In this section we review recent developments in numerical analysis on the approximation of high-dimensional functions. Our description closely follows Brumm and Scheidegger (2014). After introducing some notation, this section proceeds in four main steps. First, we present piecewise linear hierarchical basis functions in one dimension. Second, we extend such bases to multiple dimensions via a tensor product construction. Third, we show how *classical* sparse grids alleviate the curse of dimensionality to some extent (see, e.g., Bungartz and Griebel (2004); Garcke and Griebel (2012), and references therein). Fourth, we explain how the hierarchical structure of the basis functions and the associated sparse grid can be used to build an adaptation procedure that can better capture the local behavior of the functions to be interpolated. Finally, we provide an example of a function with steep gradients and non-differentiabilities where adaptive sparse grids outperform *classical* sparse grids by far.

5.1 Notation

We will focus on the domain $\Omega = [0, 1]^d$, where d is the dimensionality of the problem. Many, but not all domains occurring in economics can be transformed into such a cube by proper rescaling.

Let $\vec{l} = (l_1, \dots, l_d) \in \mathbb{N}^d$ and $\vec{i} = (i_1, \dots, i_d) \in \mathbb{N}^d$ denote multi-indices representing the grid refinement level and the spatial position of a d -dimensional gridpoint $\vec{x}_{\vec{l}, \vec{i}}$. Using this notation, we can define the full grid $\Omega_{\vec{l}}$ on Ω with mesh size

$$h_{\vec{l}} := (h_{l_1}, \dots, h_{l_d}) = 2^{-\vec{l}} := (2^{-l_1}, \dots, 2^{-l_d}), \quad (9)$$

and generic gridpoint

$$\vec{x}_{\vec{l}, \vec{i}} := (x_{l_1, i_1}, \dots, x_{l_d, i_d}), \quad (10)$$

where $x_{l_t, i_t} := i_t \cdot h_{l_t} = i_t \cdot 2^{-l_t}$, $i_t \in \{0, 1, \dots, 2^{l_t}\}$, and $t \in \{1, \dots, d\}$.⁴ Along each dimension, the grid is equidistant. However, the mesh sizes, h_{l_t} , may differ across dimensions.

In addition, when dealing with d -dimensional multi-indices such as \vec{l} , we use relational operators component-wise,

$$\vec{l} \leq \vec{k} \Leftrightarrow l_t \leq k_t, \forall t \in \{1, \dots, d\}. \quad (11)$$

Finally, we use the l_1 -norm, $|\vec{l}|_1$, and the maximum norm, $|\vec{l}|_\infty$, given by

$$|\vec{l}|_1 := \sum_{t=1}^d l_t, \quad |\vec{l}|_\infty := \max_{1 \leq t \leq d} l_t. \quad (12)$$

5.2 Hierarchical Basis Functions in One Dimension

The sparse grid method introduced below is based on a hierarchical decomposition of the underlying approximation space. Such a hierarchical structure is convenient both for local adaptivity (see Sec. 5.6) and for the use of parallel computing (see Sec. 6.2). We explain this hierarchical structure starting with the one-dimensional case, $\Omega = [0, 1]$. Afterwards, we extend it to the multivariate case using tensor products.

Let us assume that a function $f : \Omega \rightarrow \mathbb{R}$ of interest is sufficiently smooth (see, e.g., Bungartz and Griebel (2004)). For the time being we also assume that the function f vanishes at the boundary (i.e. $f|_{\partial\Omega} = 0$).⁵ An interpolation formula is then given by

$$f(x) \approx u(x) := \sum_i \alpha_i \phi_i(x) \quad (13)$$

with coefficients α_i and a set of appropriate piecewise linear basis functions $\phi_i(x)$. One standard approach is to use hat functions

$$\phi(x) = \begin{cases} 1 - |x| & \text{if } x \in [-1, 1] \\ 0 & \text{else} \end{cases} \quad (14)$$

⁴A concrete example for $d = 3$ is the following: let $\vec{l} = (2, 2, 3)$, $\vec{i} = (2, 1, 2)$. Then, $\vec{x}_{\vec{l}, \vec{i}} = (2 \cdot 0.25, 1 \cdot 0.25, 3 \cdot 0.125) = (0.5, 0.25, 0.375)$.

⁵We delegate the treatment of non-zero boundaries to Sec. 5.5.

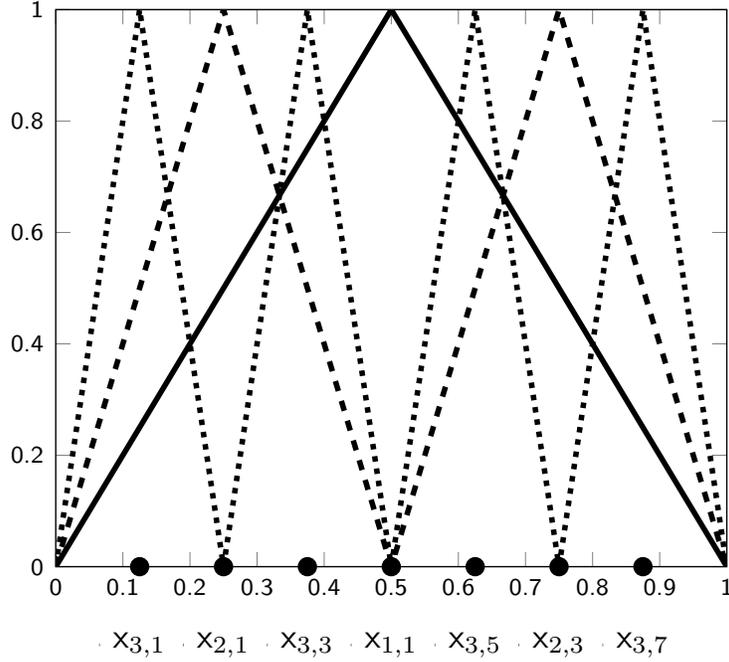


Figure 1: Hierarchical basis functions of V_3 ; level 1 (solid), level 2 (dashed), level 3 (dotted).

to generate a family of basis functions $\phi_{l,i}$ with support $[x_{l,i} - h_l, x_{l,i} + h_l]$ by defining

$$\phi_{l,i}(x) := \phi\left(\frac{x - i \cdot h_l}{h_l}\right). \quad (15)$$

This basis is called *nodal basis*, and the respective nodal function spaces are

$$V_l := \text{span}\{\phi_{l,i} : 1 \leq i \leq 2^l - 1\}. \quad (16)$$

The hierarchical increment spaces W_l are defined by

$$W_l := \text{span}\{\phi_{l,i} : i \in I_l\}, \quad (17)$$

using the index set

$$I_l = \{i \in \mathbb{N}, 1 \leq i \leq 2^l - 1, i \text{ odd}\}. \quad (18)$$

The nodal spaces V_l are the direct sum of the hierarchical increment spaces W_l

$$V_l = \bigoplus_{k \leq l} W_k. \quad (19)$$

Fig. 1 shows the first three levels of these hierarchical, piecewise linear basis functions. Using this basis, a function f is approximated by a unique $u \in V_l$ with coefficients $\alpha_{k,i} \in \mathbb{R}$:

$$f(x) \approx u(x) = \sum_{k=1}^l \sum_{i \in I_k} \alpha_{k,i} \cdot \phi_{k,i}(x). \quad (20)$$

Note that the basis functions $\phi_{k,i}$ spanning W_k have mutually disjoint support, as can be seen in Fig. 1. The coefficients $\alpha_{k,i}$ in Eq. 20 can easily be determined due to the nested structure of the

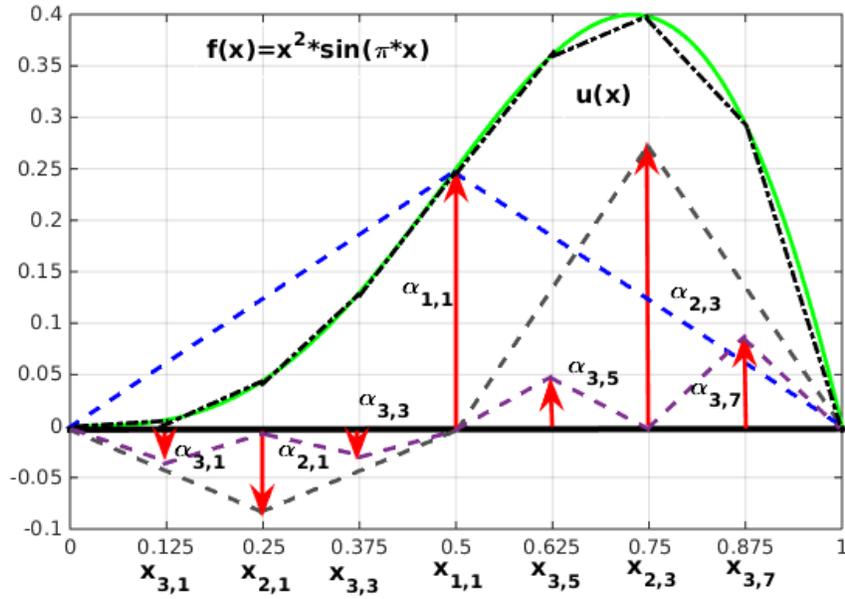


Figure 2: Construction of $u(x)$ interpolating $f(x) = x^2 \cdot \sin(\pi \cdot x)$ with hierarchical linear basis functions of levels 1, 2 and 3. The hierarchical surpluses $\alpha_{l,i}$ associated with the respective basis functions are indicated by arrows (cf. Eq. 21). They are simply the difference between the function values at the current and the previous interpolation levels.

hierarchical grid: The set of points X^{l-1} at level $l-1$ is contained in X^l (i.e. $X^{l-1} \subset X^l$). The hierarchical coefficients $\alpha_{l,i}$, $l \geq 1$, i odd are given by:

$$\begin{aligned} \alpha_{l,i} &= f(x_{l,i}) - \frac{f(x_{l,i} - h_l) + f(x_{l,i} + h_l)}{2} \\ &= f(x_{l,i}) - \frac{f(x_{l-1,(i-1)/2}) + f(x_{l-1,(i+1)/2})}{2}. \end{aligned} \quad (21)$$

In operator form, Eq. 21 can conveniently be rewritten as

$$\alpha_{l,i} = \begin{bmatrix} -\frac{1}{2} & 1 & -\frac{1}{2} \end{bmatrix}_{l,i} f. \quad (22)$$

Note that the coefficients $\alpha_{l,i}$ are called *hierarchical surpluses* (Bungartz and Griebel, 2004) since they correct the interpolant of level $l-1$ at the point $x_{l,i}$ to the actual value of $f(x_{l,i})$, as displayed in Fig. 2.

5.3 Hierarchical Basis Functions in Multiple Dimensions

The one-dimensional hierarchical basis from above can be extended to a d -dimensional basis on the unit cube $\Omega = [0, 1]^d$ by a tensor product construction. Our notation naturally extends to the d -dimensional case as well.

For each gridpoint, $\vec{x}_{l,i}$, an associated piecewise d -linear basis function $\phi_{\vec{l},i}(\vec{x})$ is defined as the

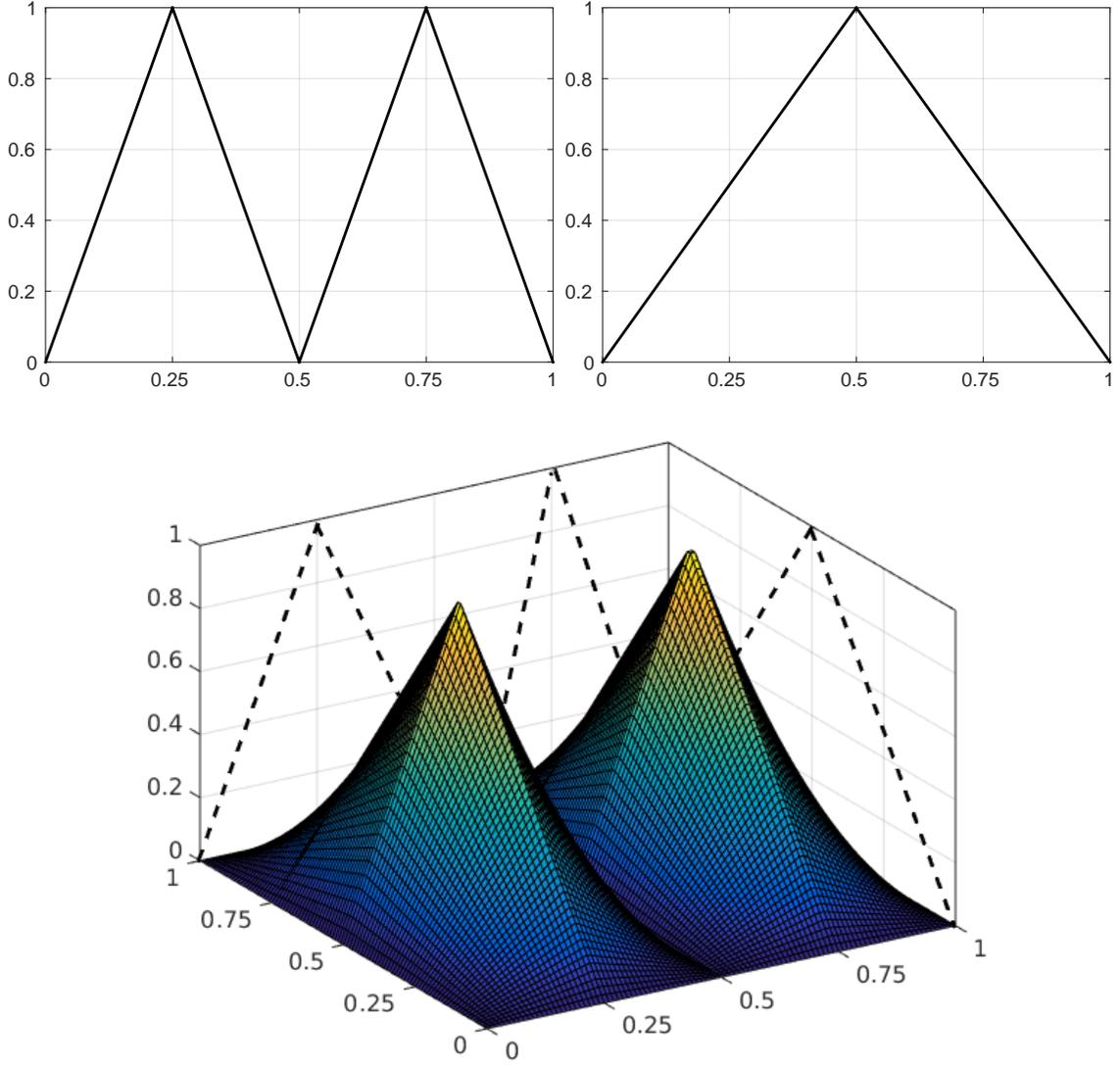


Figure 3: The one-dimensional basis functions of level 2 (left top panel) and level 1 (right top panel) that form the two-dimensional basis function of the hierarchical increment space $W_{(2,1)}$ (lower panel) via a tensor product construction (see Eq. 23).

product of the one-dimensional basis functions (see Eq. 14):

$$\phi_{\vec{l}, \vec{i}}(\vec{x}) := \prod_{t=1}^d \phi_{l_t, i_t}(x_t). \quad (23)$$

These basis functions are then used to define the function spaces $V_{\vec{l}}$ consisting of piecewise linear functions on Ω (with $f|_{\partial\Omega} = 0$):

$$V_{\vec{l}} := \text{span}\{\phi_{\vec{l}, \vec{i}} : \vec{l} \leq \vec{i} \leq 2^{\vec{l}} - \vec{l}\}. \quad (24)$$

The index set $I_{\vec{l}}$ is given by

$$I_{\vec{l}} := \{\vec{i} : 1 \leq i_t \leq 2^{l_t} - 1, i_t \text{ odd}, 1 \leq t \leq d\}. \quad (25)$$

and the hierarchical increment spaces are defined as

$$W_{\vec{l}} := \text{span}\{\phi_{\vec{l},i} : i \in I_{\vec{l}}\}. \quad (26)$$

An example of such an increment space is shown in Fig. 3. These hierarchical increment spaces now allow us to define a multilevel space decomposition. In line with the sparse grid literature (see, e.g., Pflüger (2010); Garcke and Griebel (2012); Bungartz and Griebel (2004)), we define $V_n := V_{(n,\dots,n)}$ as a direct sum of spaces. Consequently, the hierarchical increment spaces $W_{\vec{l}}$ are related to the nodal spaces $V_{\vec{l}}$ of piecewise d -linear functions with mesh width h_l in each dimension by

$$V_n := \bigoplus_{l_1=1}^n \cdots \bigoplus_{l_d=1}^n W_{\vec{l}} = \bigoplus_{|\vec{l}|_{\infty} \leq n} W_{\vec{l}}, \quad (27)$$

leading to a full grid with $(2^n - 1)^d$ gridpoints⁶. The interpolant of f , namely $u(\vec{x}) \in V_n$, can uniquely be represented by

$$f(\vec{x}) \approx u(\vec{x}) = \sum_{|\vec{l}|_{\infty} \leq n} \sum_{i \in I_{\vec{l}}} \alpha_{\vec{l},i} \cdot \phi_{\vec{l},i}(\vec{x}) = \sum_{|\vec{l}|_{\infty} \leq n} f_{\vec{l}}(\vec{x}), \quad (28)$$

with $f_{\vec{l}} \in W_{\vec{l}}$ and $\alpha_{\vec{l},i} \in \mathbb{R}$. The hierarchical surpluses are given by

$$\alpha_{\vec{l},i} = \left(\prod_{t=1}^d \left[-\frac{1}{2} \quad 1 \quad -\frac{1}{2} \right]_{l_t, i_t} \right) f. \quad (29)$$

For a sufficiently smooth function f (which we will state more precisely in Sec. 5.4) and its interpolant $u \in V_n$ (Bungartz and Griebel, 2004), we obtain an asymptotic error decay of

$$\|f(\vec{x}) - u(\vec{x})\|_{L_2} \in \mathcal{O}(h_n^2) = \mathcal{O}(2^{-2n}), \quad (30)$$

but at the cost of

$$\mathcal{O}(h_n^{-d}) = \mathcal{O}(2^{nd}) \quad (31)$$

gridpoints, encountering the so-called *curse of dimensionality*. The exponential dependence of the overall computational effort on the number of dimensions is a prohibitive obstacle for the numerical treatment of high-dimensional problems. The curse of dimensionality typically prohibits an accurate solution of problems with more than 4 or 5 dimensions.

5.4 Classical Sparse Grids

To alleviate the curse of dimensionality (see Sec. 5.3) we need to construct approximation spaces that are better than V_n in the sense that the same number of gridpoints leads to higher accuracy (see, e.g., Zenger (1991); Bungartz and Griebel (2004)). The *classical* sparse grid construction arises from a *cost-benefit* analysis (see, e.g., Bungartz and Griebel (2004), and references therein)

⁶Note that $V_n \subset V_{n+1}$.

in function approximation. In particular, we consider the Sobolev space of functions with bounded second-order mixed derivatives

$$H_2(\Omega) := \{f : \Omega \rightarrow \mathbb{R} : D^{\vec{l}}f \in L_2(\Omega), |\vec{l}|_\infty \leq 2, f|_{\partial\Omega} = 0\}, \quad (32)$$

where

$$D^{\vec{l}}f := \frac{\partial^{|\vec{l}|_1}}{\partial x_1^{l_1} \cdots \partial x_d^{l_d}} f. \quad (33)$$

For functions in $H_2(\Omega)$ the hierarchical coefficients $\alpha_{\vec{l}, \vec{i}}$ (see Bungartz and Griebel (2004) and Eq. 28) decay rapidly—namely,

$$|\alpha_{\vec{l}, \vec{i}}| = \mathcal{O}\left(2^{-2|\vec{l}|_1}\right). \quad (34)$$

The strategy for constructing a sparse grid is to leave out those subspaces within the full grid space V_n that only contribute little to the interpolant.

An optimization that minimizes the approximation error given a fixed number of gridpoints (or vice versa) leads to the *sparse grid* space $V_{0,n}^S$ of level n , defined by

$$V_{0,n}^S := \bigoplus_{|\vec{l}|_1 \leq n+d-1} W_{\vec{l}}, \quad (35)$$

where the index 0 in $V_{0,n}^S$ stands for $f|_{\partial\Omega} = 0$ (see Fig. 4 and Bungartz and Griebel (2004), with references therein). Note that the actual choice of subspaces depends on the norm in which we measure the error. The result obtained in Eq. 35 is optimal for the L_2 -norm and the L_∞ -norm.

The number of gridpoints required by the space $V_{0,n}^S$ is given by (see, e.g. Bungartz and Griebel (2004))

$$|V_{0,n}^S| = 2^n \cdot \left(\frac{n^{d-1}}{(d-1)!} + \mathcal{O}\left(n^{d-2}\right) \right) = \mathcal{O}\left(h_n^{-1} \cdot (\log(h_n^{-1}))^{d-1}\right) = \mathcal{O}\left(2^n \cdot n^{d-1}\right), \quad (36)$$

which is a significant reduction of the number of gridpoints, and thus of the computational and storage requirements compared to $\mathcal{O}(2^{nd})$ for the full grid space V_n (see Tab. 1 below in Sec. 5.5).

In analogy to Eq. 28, a function $f \in V_{0,n}^S \subset V_n$ is now approximated by

$$f_{0,n}^S(\vec{x}) \approx u(\vec{x}) = \sum_{|\vec{l}|_1 \leq n+d-1} \sum_{\vec{i} \in I_{\vec{l}}} \alpha_{\vec{l}, \vec{i}} \cdot \phi_{\vec{l}, \vec{i}}(\vec{x}) = \sum_{|\vec{l}|_1 \leq n+d-1} f_{\vec{l}}(\vec{x}), \quad (37)$$

where $f_{\vec{l}} \in W_{\vec{l}}$. As in the one-dimensional case, a hierarchical surplus $\alpha_{\vec{l}, \vec{i}} \in \mathbb{R}$ is simply the difference between the function value at the current and the previous interpolation level (cf. Sec. 5.2). Since the gridpoints are nested (i.e. X^{l-1} at level $l-1$ is contained in X^l) the extension of the interpolation level from level $l-1$ to l only requires the evaluation of the function at gridpoints that are unique to X^l —that is, at $X_\Delta^l = X^l \setminus X^{l-1}$.

The asymptotic accuracy of the interpolant deteriorates only slightly from $\mathcal{O}(h_n^2)$ in case of the full grid (cf. Eq. 30) to

$$\mathcal{O}\left(h_n^2 \cdot \log(h_n^{-1})^{d-1}\right), \quad (38)$$

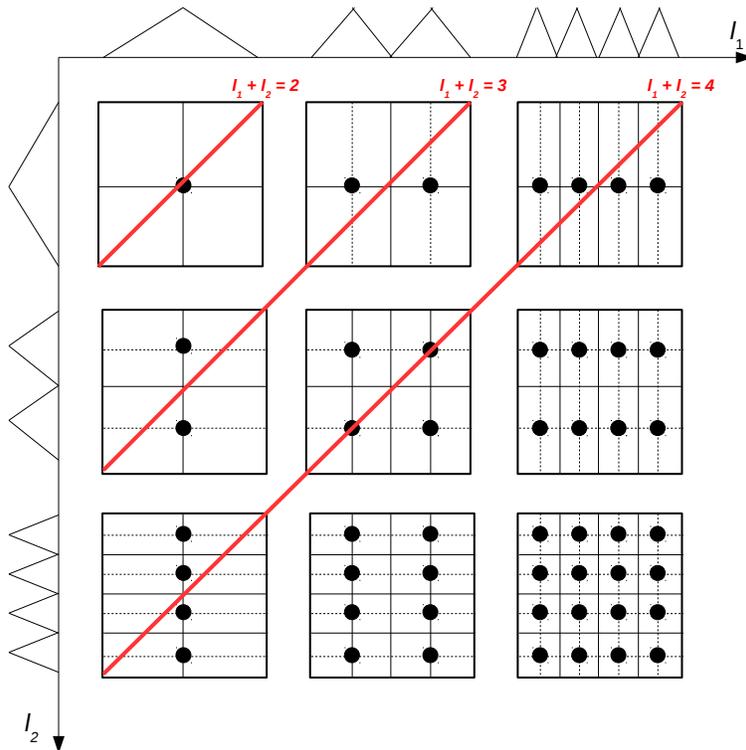


Figure 4: We show the two-dimensional subspaces $W_{\vec{l}}$ up to $l = 3$ ($h_3 = 1/8$) in each dimension. The construction of the sparse grid space $V_{0,3}^S$ in 2 dimensions according to Eq. 35 is indicated by the red lines. This space is the direct sum of all hierarchical subspaces satisfying $|\vec{l}|_1 \leq 4$. Note that it contains only 17 support nodes, whereas a full grid of the same level would consist of 49 points.

as shown in Bungartz and Griebel (2004). Taken together, Eqs. 36 and 38 demonstrate why sparse grids are so well-suited to high-dimensional problems. In contrast to full grids, their size increases a lot slower with dimensions, while they are only slightly less accurate than full grids.

Note that sparse grid methods are not restricted to piecewise linear basis functions; there are several other basis functions possible, including piecewise polynomials (see Bungartz and Griebel (2004); Pflüger (2010), and references therein). However, we focus on linear hat functions, since these are most convenient for adaptive refinement procedures as presented below in Sec. 5.6.

5.5 Sparse Grids with Non-Zero Boundaries

So far, we have assumed that the functions under consideration vanish at the boundary of the domain—that is, $f|_{\partial\Omega} = 0$. To allow for non-zero values at the boundary, the procedure one usually follows is to add additional gridpoints located directly on $\partial\Omega$ (see, e.g., Pflüger (2010); Klimke and Wohlmuth (2005)). Doing this naively, one needs at least 3^d gridpoints, which makes the approach inapplicable to high-dimensional problems. In what follows, we discuss one particular procedure that mitigates this issue (see, e.g., Ma and Zabaras (2009); Klimke and Wohlmuth (2005)). The crucial idea is to have only one gridpoint at the lowest level of approximation. Technically, the

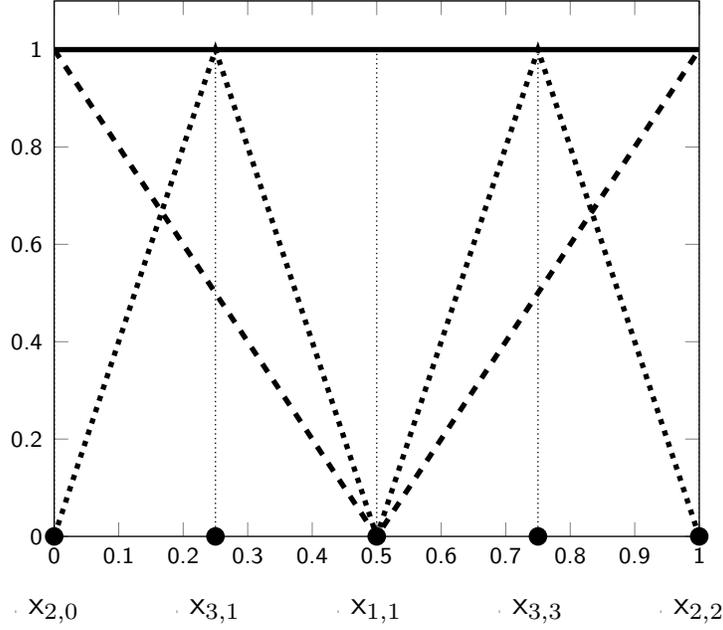


Figure 5: Hierarchical basis functions of the $V_{NZ,n}^S$ sparse grid in one dimension. Level 1 (solid), level 2 (dashed), and level 3 (dotted).

difference to the sparse grid with zero ($V_{0,n}^S$) and one with non-zero ($V_{NZ,n}^S$) boundaries is that the index set of the support nodes is not given by Eq. 25, but rather by

$$I_{\vec{l}} := \begin{cases} \{\vec{i} : i_t = 1, 1 \leq t \leq d\} & \text{if } l = 1 \\ \{\vec{i} : 0 \leq i_t \leq 2, i_t \text{ even}, 1 \leq t \leq d\} & \text{if } l = 2 \\ \{\vec{i} : 1 \leq i_t \leq 2^{l_t-1} - 1, i_t \text{ odd}, 1 \leq t \leq d\} & \text{else,} \end{cases} \quad (39)$$

and the one-dimensional basis functions for each dimension t are given by

$$\phi_{l_t, i_t}^{NZ}(x_t) = \begin{cases} 1 & \text{if } l = 1 \wedge i = 1 \\ \left\{ \begin{array}{l} 1 - 2 \cdot x_t \quad \text{if } x_t \in [0, \frac{1}{2}] \\ 0 \quad \text{else} \end{array} \right\} & \text{if } l = 2 \wedge i = 0 \\ \left\{ \begin{array}{l} 2 \cdot x_t - 1 \quad \text{if } x_t \in [\frac{1}{2}, 1] \\ 0 \quad \text{else} \end{array} \right\} & \text{if } l = 2 \wedge i = 2 \\ \phi_{l,i}(x_t) & \text{else,} \end{cases} \quad (40)$$

where $\phi_{l,i}(x)$ is given by Eq. 15. The first three levels of these basis functions are displayed in Fig. 5. Besides the choice of basis functions the same logic as stated in Eq. 35 applies for construction of the sparse grid space $V_{NZ,n}^S$. Examples for two- and three-dimensional sparse grids with non-zero boundaries of level 4 are given in Fig. 6. It is also worth mentioning that the number of gridpoints of the non-zero boundary sparse grid $|V_{NZ,n}^S|$ grows slightly slower than $|V_{0,n}^S|$ (cf. Tab. 1).

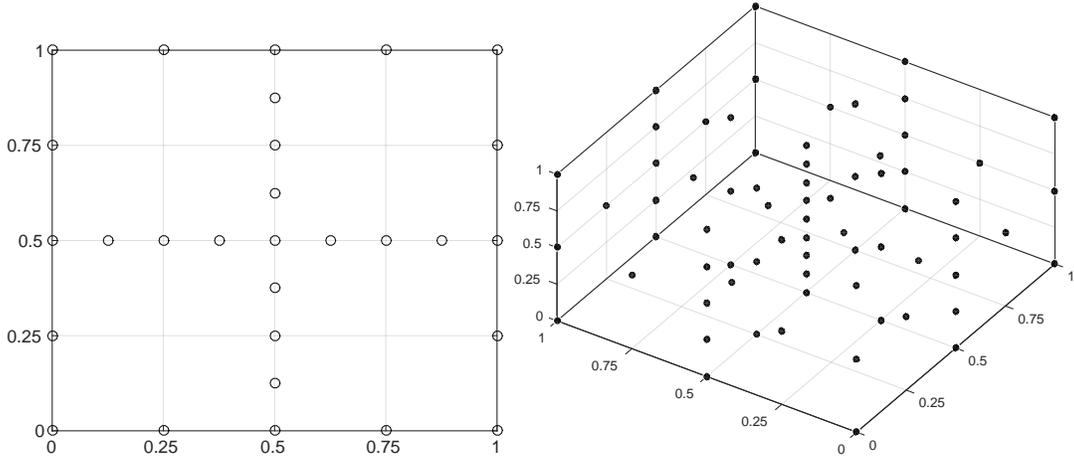


Figure 6: Two-dimensional (left) and three-dimensional (right) sparse grids, $V_{NZ,n}^S$, each of level $n = 4$.

d	$ V_4 $	$ V_{0,4}^S $	$ V_{NZ,4}^S $
1	15	15	9
2	225	49	29
3	3,375	111	69
4	50,625	209	137
5	759,375	351	241
10	$5.77 \cdot 10^{11}$	2,001	1,581
20	$3.33 \cdot 10^{23}$	13,201	11,561
50	$6.38 \cdot 10^{58}$	182,001	171,901
100	>Googol	1,394,001	1,353,801

Table 1: Number of gridpoints for several different grid types of level 4. First column—dimension; second column—full grid; third column—“classical” sparse grid with no points at the boundaries; last column—non-zero boundary sparse grid.

5.6 Refining Sparse Grids Adaptively

Functions that do not meet the smoothness requirements or that show significantly varying characteristics across the domain of interest (see Brumm and Scheidegger (2014) for examples of economic models with occasionally binding constraints) can still be tackled with sparse grids if adaptivity is used. The sparse grid structure introduced in Eq. 35 defines an a priori selection of grid points that is optimal for functions with bounded second-order mixed derivatives. An adaptive (a posteriori) refinement can additionally, based on local features of the function, select which grid points in the sparse grid structure should be refined (see, e.g., Pflüger (2012), Bungartz and Dirnstorfer (2003), and Ma and Zabararas (2009)). The most common way of doing so is the following heuristics: When approximating a function as a sum of piecewise linear basis functions, the main contributions to

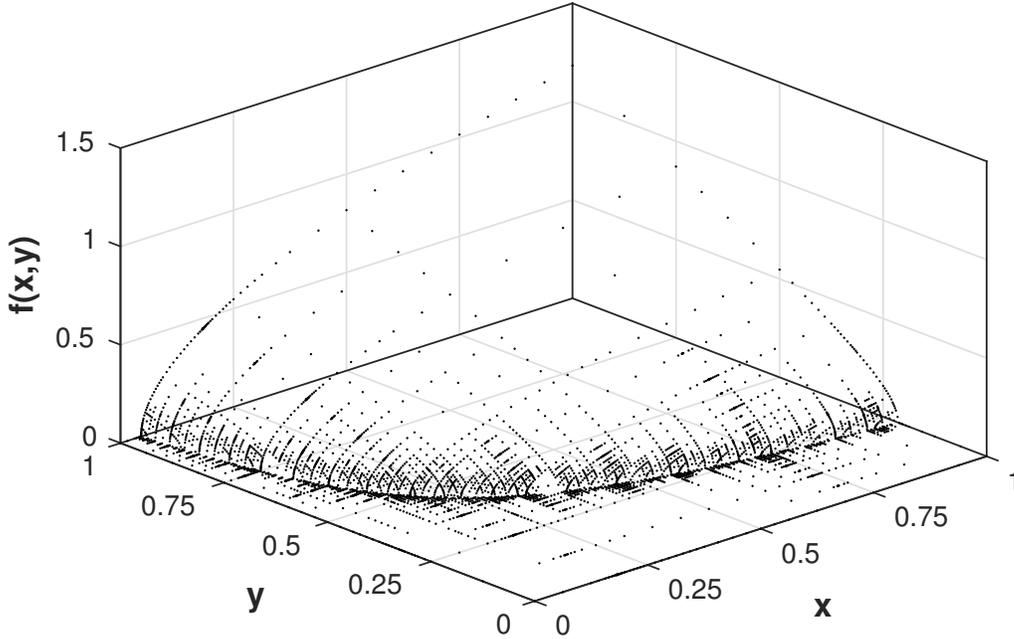


Figure 7: Evaluation of Eq. 42 at the gridpoints obtained by the adaptive sparse grid algorithm run with refinement threshold $\epsilon = 10^{-3}$ and basis functions for non-vanishing boundary conditions (cf. Sec. 5.5) at refinement level 15.

the interpolant (most likely) stem from comparatively few terms with big surpluses (cf. Eq. 37). The logic of the refinement strategies is therefore to monitor the size of the hierarchical surpluses. The magnitude of the hierarchical surplus reflects the local irregularity of the function and thus serves as a natural error indicator. *2d children* in the hierarchical structure are added locally to the current grid for those hierarchical basis functions, $\phi_{\vec{l}, \vec{j}}$, that have a hierarchical surplus, $\alpha_{\vec{l}, \vec{j}}$, that satisfies $|\alpha_{\vec{l}, \vec{j}}| \geq \epsilon$ for a so-called refinement threshold $\epsilon \geq 0$.⁷ Whenever this criterion is satisfied, the *children* of the current point are added to the sparse grid. If additional knowledge about the problem at hand is available, it can be used in the criterion for adaptive refinement, allowing to better adapt the problem specific characteristics (see, e.g., Brumm and Scheidegger (2014)). For more details regarding adaptive sparse grids, we refer the reader to Bungartz and Dirnstorfer (2003), Ma and Zabaras (2009), Brumm and Scheidegger (2014), and Pflüger (2012).

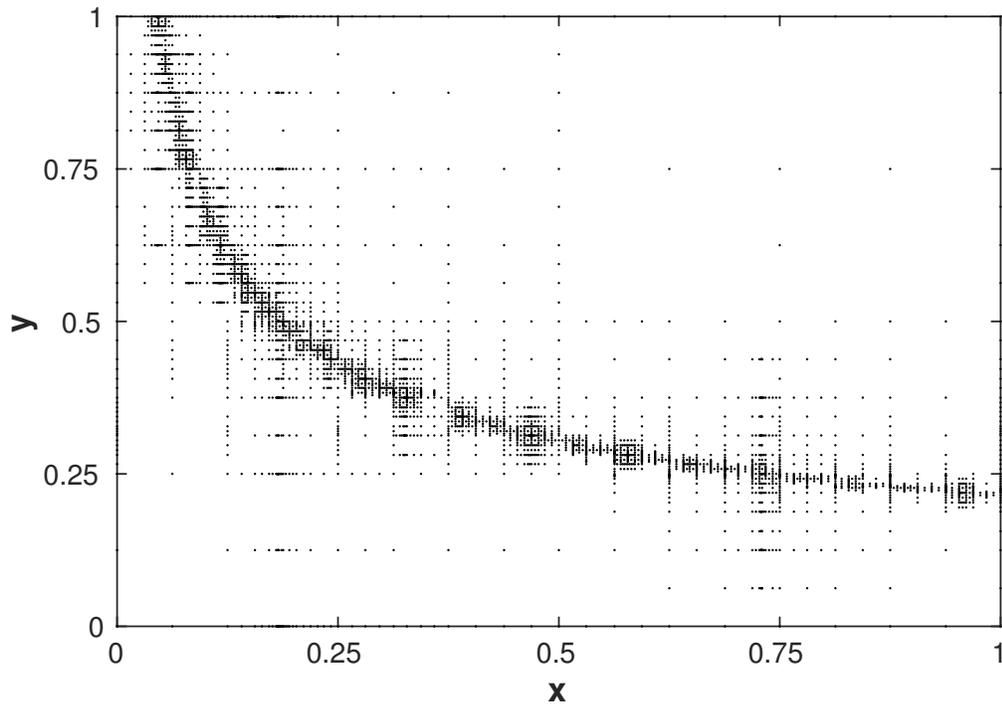


Figure 8: Adaptive sparse grid on $[0, 1]^2$, constructed with a refinement threshold $\epsilon = 10^{-3}$ and basis functions for non-vanishing boundary conditions (cf. Sec. 5.5). Refinement level 15, consisting of 4087 points, is shown.

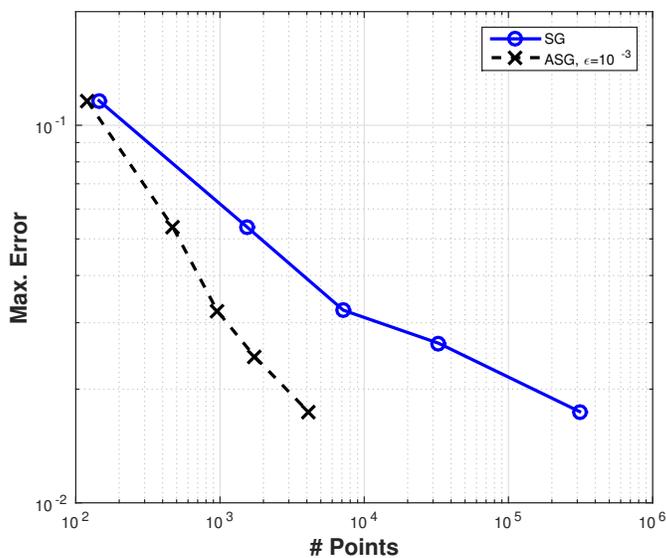


Figure 9: Comparison of the interpolation error (cf. Eq. 41) for conventional and adaptive sparse grid interpolation at different refinement levels. Note that the adaptive sparse grid algorithm labelled “ASG”, while its corresponding *classical* sparse grid version is denoted by “SG”. The adaptive grids was obtained by applying a threshold $\epsilon = 10^{-3}$.

Analytical Example

We now demonstrate the ability of the adaptive sparse grid algorithm to efficiently interpolate functions that exhibit steep gradients and kinks. We present an analytical example in two dimensions. This should foster an understanding of the adaptive sparse grid algorithm in use.

For the testing, we proceed as follows: We pick a (non-smooth) function $f : [0, 1]^2 \rightarrow \mathbb{R}$, construct the interpolant $u(\vec{x})$ of $f(\vec{x})$, then randomly generate 1,000 test points from a uniform distribution on $[0, 1]^d$, and finally compute the maximum error given by

$$e = \max_{i=1, \dots, 1000} |f(\vec{x}_i) - u(\vec{x}_i)|. \quad (41)$$

As example, we apply the adaptive sparse grid algorithm to a problem with non-vanishing boundaries and a threshold of $\epsilon = 10^{-3}$ to the two-dimensional test function

$$f(x, y) = \sqrt{\max \left[\left(xy^2 - \frac{1}{\pi} \right) \cdot \frac{\pi}{\pi - 1} + 0.4, 0 \right]} \quad (42)$$

plotted in Fig. 7. We chose this function because this type of shape is typical for a policy function in a model with an occasionally binding constraint (and two continuous state variables in this case). Note that the line of non-differentiability is automatically detected by the adaptive sparse grid algorithm (cf. Figs. 7 and 8). In Fig. 9, we provide the convergence rate of the adaptive sparse grid method. The data points shown in Fig. 9 were obtained by computing the errors (cf. Eq. 41) at levels 5, 8, 10, 12, and 15. These results are contrasted with the “classical” sparse grid counterparts of the respective refinement level.

Strikingly, the solution obtained by the adaptive sparse grid reaches the same accuracy as the “classical” sparse grid, however with considerable fewer points. A refinement level 15 adaptive sparse grid requires 4,087, opposed to 311,297 points using the same level of refinement in the conventional sparse grid (or $\mathcal{O}(10^9)$ points in case of a Cartesian grid).

6 A time-iteration collocation method for models without idiosyncratic risk

To illustrate the computational performance of time-iteration collocation we consider the version of the model without idiosyncratic shocks. Except for the fact that we allow for recursive utility, this is exactly the model considered in Krueger and Kubler (2004). We illustrate that the use of adaptive sparse grids together with the use of parallel computing enables us to solve models with 60 generations, i.e. models that are calibrated to yearly data. We also use this simple setup to demonstrate possible advantages of our method relative to simulation-based methods.

⁷However, depending on the application, more sophisticated criteria might need to be imposed for an efficient approximation. Thus, we have to replace the trivial refinement criterion by some $g(\alpha_{i,\vec{x}}) \geq \epsilon$, where the refinement choice is governed by a function $g : \mathbb{R} \rightarrow \mathbb{R}$.

6.1 Time-iteration collocation

As in Krueger and Kubler (2004) we use an iterative scheme to solve for the unknown coefficients of the piecewise multi-linear functions that are solutions to our collocation equations. For high-dimensional problems, even when using sparse grids, there are so many coefficients to solve for that simple Newton methods quickly become infeasible.⁸ Moreover, there is no straightforward way to combine such a direct collocation approach with adaptive sparse grids. Therefore we use a time-iteration algorithm, which can be interpreted as solving finite-horizon approximations to our infinite-horizon model. The main idea of our algorithm is similar to a basic policy function iteration as proposed, for example, by Coleman (1990). The main innovation, yet also complication, of our approach stems from the adaptive part of the sparse grid, which operates within each time iteration step. The details of the algorithm are as follows:

1. Make an initial guess for next period's policy function:

$$p' : \mathbf{Z} \times \mathbf{B} \rightarrow \mathbb{R}^{2(A-1)}, p'(z', s') = (k'_1(z', s'), \dots, k'_{A-1}(z', s'), v'_1(z', s'), \dots, v'_{A-1}(z', s')).$$

Choose an approximation accuracy $\bar{\eta}$.

2. Make one time iteration step:

- (a) For each $z \in \mathbf{Z}$, start with a coarse grid $G_{old}^z \subset S$ (a “classical” sparse grid of a “low” level L_0), and generate G^z by adding for each $x \in G_{old}^z$ all $2d$ neighboring points. Choose a refinement threshold ϵ and a maximal level $L_{max} > L_0$ and set $l = 1$.
- (b) For each $z \in \mathbf{Z}$ and each gridpoint

$$s \in \begin{cases} G^z & \text{if } l = 1 \\ G^z \setminus G_{old}^z & \text{if } l > 1 \end{cases}$$

solve for the optimal policies

$$p(z, s) = (k_1(z, s), \dots, k_{A-1}(z, s), v_1(z, s), \dots, v_{A-1}(z, s))$$

at (z, s) by solving the system of equilibrium conditions given next period's policy

$$p'(z', s') = (k'_1(z', s'), \dots, k'_{A-1}(z', s'), v'_1(z', s'), \dots, v'_{A-1}(z', s')).$$

⁸When agents live for 60 periods, even when there are only 4 exogenous shocks and a “classical” sparse grid of only level 3 is used, a nonlinear equation system with $59 \cdot 4 \cdot 7,081 \cdot 2 = 3,342,232$ unknown coefficients has to be solved—for each of the 59 (active) agents and each of the 4 shocks, we have 7,081 coefficients both for the capital policy function and the value function.

More precisely, the system of equilibrium conditions can be written as

$$\begin{aligned} & \hat{c}_a(z, \tilde{s}|\alpha)^{\rho-1} + \beta_a \left[\sum_{z'} \pi(z'|z) (\hat{v}'_{a+1}(z', s'|\alpha))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}-1} \\ & \sum_{z'} \pi(z'|z) (1+r(z', s')) \hat{v}'_{a+1}(z', s'|\alpha)^{\frac{\sigma}{\rho}-1} \hat{c}'_{a+1}(z', s'|\alpha)^{\rho-1} = 0, \quad a = 1, \dots, A-1 \end{aligned}$$

and

$$\hat{v}_a(z, \tilde{s}|\alpha) = \hat{c}_a(z, \tilde{s}|\alpha)^\rho + \beta_a \left[\sum_{z'} \pi(z'|z) (\hat{v}'_{a+1}(z', s'|\alpha))^{\frac{\sigma}{\rho}} \right]^{\frac{\rho}{\sigma}}, \quad a = 1, \dots, A-1. \quad (43)$$

where $\hat{v}_A(z, s|\alpha) = \hat{c}(z, s|\alpha)^\rho$, $s'_1 = \sum_{i=1}^{A-1} \hat{k}_i(z, \tilde{s})$, and

$$\begin{aligned} s' &= \left(s'_1, \hat{k}_1(z, \tilde{s})(1+r(z', s'_1)), \dots, \hat{k}_{(A-2)}(z, \tilde{s})(1+r(z', s'_1)) \right) \in \mathbf{B} \\ \hat{c}_1(z, s|\alpha) &= l^1(z)w(z, s_1) - \hat{k}_1(z, s) \\ \hat{c}_i(z, s; \alpha) &= s_i + l^i(z)w(z, s_1) - \hat{k}_i(z, s; \alpha) \text{ for } i = 2, \dots, N-1 \\ \hat{c}_N(z, s; \alpha) &= \left(s_1(1+r(z, s_1)) - \sum_{i=2}^{A-1} s_i \right) + l^i(z)w(z, s_1) \end{aligned}$$

and where $\hat{c}'_1, \dots, \hat{c}'_N$ are given analogously to $\hat{c}_1, \dots, \hat{c}_N$ and the prices $1+r$ and w are as above in the definition of FREE.

- (c) For each z , generate G_{new}^z from G^z by adding for each $s \in G^z \setminus G_{old}^z$ its $2d$ neighboring points if

$$g(p(z, s) - \tilde{p}(z, s)) > \epsilon,$$

where the policy $\tilde{p}(z, s)$ is given by interpolating between $\{p(z, s)\}_{s \in G_{old}^z}$ (thus $p(z, s) - \tilde{p}(z, s)$ is the hierarchical surplus at s), and $g : \mathbb{R}^{2(A-1)} \rightarrow \mathbb{R}$ is chosen to be $g(x) = \sum_{i=1}^{A-1} |x_i|$ or $g(x) = \max_{1 \leq i \leq A-1} |x_i|$.

- (d) Set $G_{old}^z = G^z$ and $G^z = G_{new}^z$.
(e) If $G^z = G_{old}^z$ or $L_0 + l = L_{max}$, then go to (f), otherwise set $l = l + 1$ and go to (b).
(f) Define the policy function $p(z, \cdot)$ as the (adaptive sparse grid) interpolation of $\{p(z, s)\}_{s \in G^z}$.
(g) Calculate (an approximation for) the error, for example,

$$\eta = \|p - p'\|_\infty.$$

If $\eta > \bar{\eta}$, set $p' = p$ and go to step 2, otherwise go to step 3.

3. The (approximate) equilibrium policy function is given by p .

Note that in step 2.(b) of the method we solve all Euler equations simultaneously. For this particular, simple model one can often use the previous period's policy function to solve for today's policy much more efficiently. However, for a model with assets in fixed supply (e.g. in the presence of a risk-free asset) there is no simple short-cut and all Euler equations have to be solved as one system – we explain this briefly in Section 6.4 below.

6.2 Using high-performance computing

In macroeconomics—as in other sciences—the quest for scientific discovery often has to rely on *in-silico* experiments: real experiments are often infeasible. Moreover, the theoretical models are hideously difficult and most of the time lack an analytical solution. In order to obtain a (timely) answer to the research question imposed, a way out is to use HPC systems. In many fields of science, computational research—and in particular HPC—has already become a strong, well-established “third pillar”, alongside theory and experimentation.⁹

HPC systems can probably be considered as the most powerful and flexible research instruments available today (Dongarra and van der Steen (2012)), and allow us to ask questions that would otherwise be impossible to address. Their computational power nowadays often reaches multiple petaflops, with an eleven-year cycle of achieving a three-orders-of-magnitude increase in performance (see <http://www.top500.org>). Their predominant design currently is—according to the taxonomy of Flynn (1972)—“distributed memory multiple instruction, multiple data (DM-MIMD)”, better known as so-called “clusters”. A schematic diagram is given in Fig. 10, displaying a number of processors (eight in this case) drawing on the same local memory, the nodes being connected by some (low latency) network. Thus, when a processor in node *A* needs data present in node *B*, this has to be accessed through the network; hence the characterization of the system. In addition, one or multiple general-purpose graphics processing units (GPGPUs—GPUs for short) are often attached to the nodes, forming so-called hybrid compute nodes (see <http://www.top500.org> for a detailed description of the architecture of the world’s most advanced systems). Such systems are difficult to deal with. To use them efficiently, one has to design the software carefully, taking into account the individual advantages that the various (off-the-shelf) hardware components of such machines offer. To do so, distributed memory, shared memory, and very recently even GPU programming paradigms have to be combined. Despite the promise of HPC facilities to solve complex models, the latter point is probably the reason why economists have, in the past, have not used them to the full degree possible.

Single GPUs attached to a single node were used in several applications in order to accelerate computations (see, e.g., Aldrich (2014) and Aldrich et al. (2011)). However, GPUs are still special-purpose processors—that is to say, accelerators. One has to realize that they are good at some specialized computations, but totally unable to perform others. Therefore, not all applications can benefit from them, and of those that can not all can benefit to the same degree. Cai et al. (2013) have used the high latency “Condor” paradigm to solve dynamic programming problems in parallel. Their approach is scalable—however, since it operates within high latency, their paradigm is probably best used in situations where there are multiple, totally independent problems that need to be solved and that do not require any sort of synchronization.

⁹For an in-depth description of machines that can be called “high-performance”, see Culler et al. (1997).

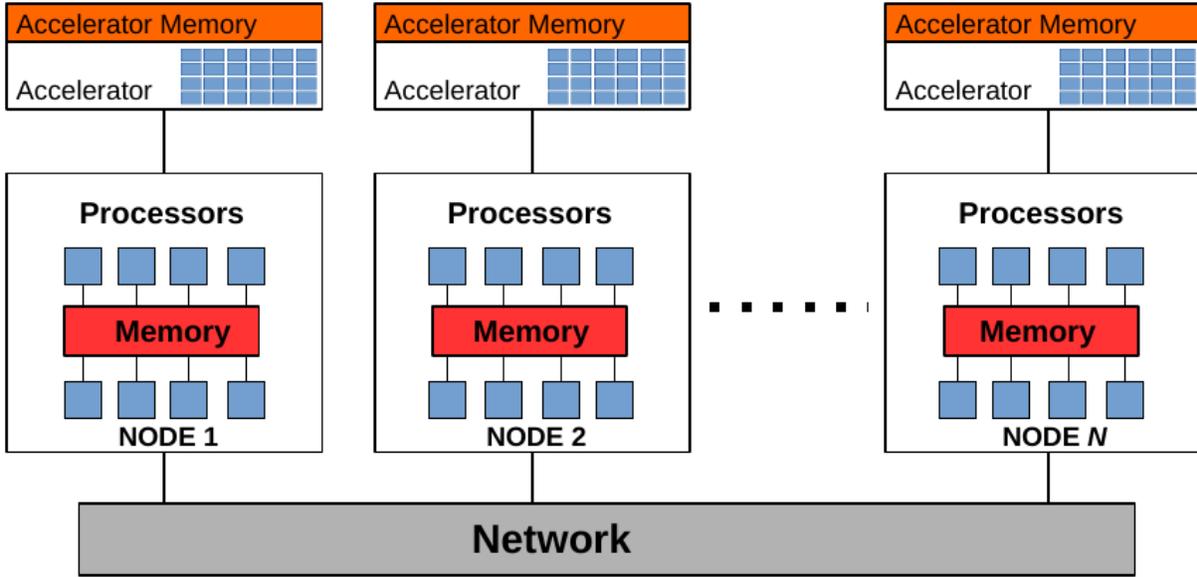


Figure 10: Diagram of a generic DM-MIMD machine.

However, apart from Cai et al. (2015)¹⁰ and Brumm and Scheidegger (2014); Brumm et al. (2015), who recently exploited highly parallel, low-latency systems, no one in computational economics has so far attempted to make efficient use of the most advanced contemporary HPC systems.

In order to solve “large” problems in a reasonable amount of time, we outline in the following the state-of-the art parallelization that Brumm and Scheidegger (2014); Brumm et al. (2015) introduced and how it extends to the OLG model discussed in this paper.

In each step of the above time iteration procedure the updated policy function is determined using a hybrid MPI/threads parallel algorithm, as illustrated in Fig. 11. Within each refinement step, we first distribute the different, independent discrete states of the problem to different MPI groups (see, e.g. Skjellum et al. (1999)). Then, within each individual MPI group, the newly generated gridpoints are split among multiple (multi-threaded) processes by MPI. The points that are sent to one particular compute node are further distributed among different threads. Each thread then solves a set of nonlinear equations for every single gridpoint assigned to it.¹¹ On top of this, we add an additional level of parallelism. When searching for the solution to the equation system at a given point, the algorithm has to frequently interpolate the function computed in the previous iteration step. These interpolations take up 99 percent of the computation time needed to solve the equation system. As they have a high arithmetic intensity they are perfectly suited for GPUs (see, e.g., Murarasu et al. (2011) and Heinecke and Pflueger (2013)). We therefore offload

¹⁰Cai et al. (2015) use an MPI-based implementation of a dynamic programming code.

¹¹The set of nonlinear equations in this example is solved with `Ipopt` (Wachter and Biegler (2006)) (<http://www.coin-or.org/Ipopt/>). Note, however, that the code framework is designed such that any solver could be added.

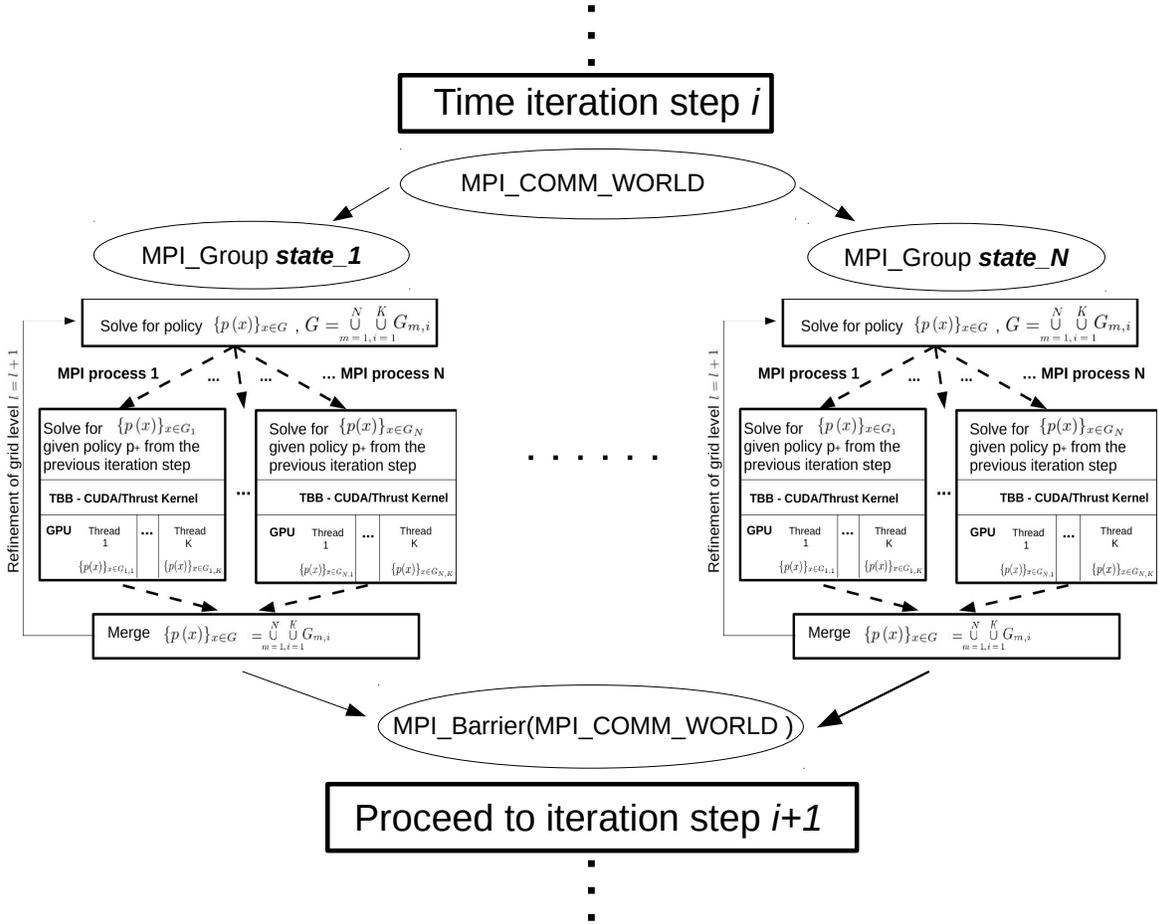


Figure 11: Schematic representation of the hybrid parallelization of a time iteration step. Each MPI process is using TBB and a CUDA/Thrust kernel for the function evaluation.

parts of the interpolation from the compute nodes to their attached accelerators. When comparing a single-threaded CPU with the case in which a GPU is also employed, we observe a speedup of up to one order of magnitude for the interpolation. In the case where we use the entire node, the multi-threading is implemented with thread building blocks (TBB; see Reinders (2007)). One of the TBB-managed threads exclusively uses the GPU, reducing the overall computation time by roughly 50 percent. Moreover, CPU and GPU threads leverage TBBs' automatic workload balancing based on stealing tasks from slower threads, as shown in Fig. 12. More details regarding the parallel implementation and optimization of our code can be found in Brumm et al. (2015).

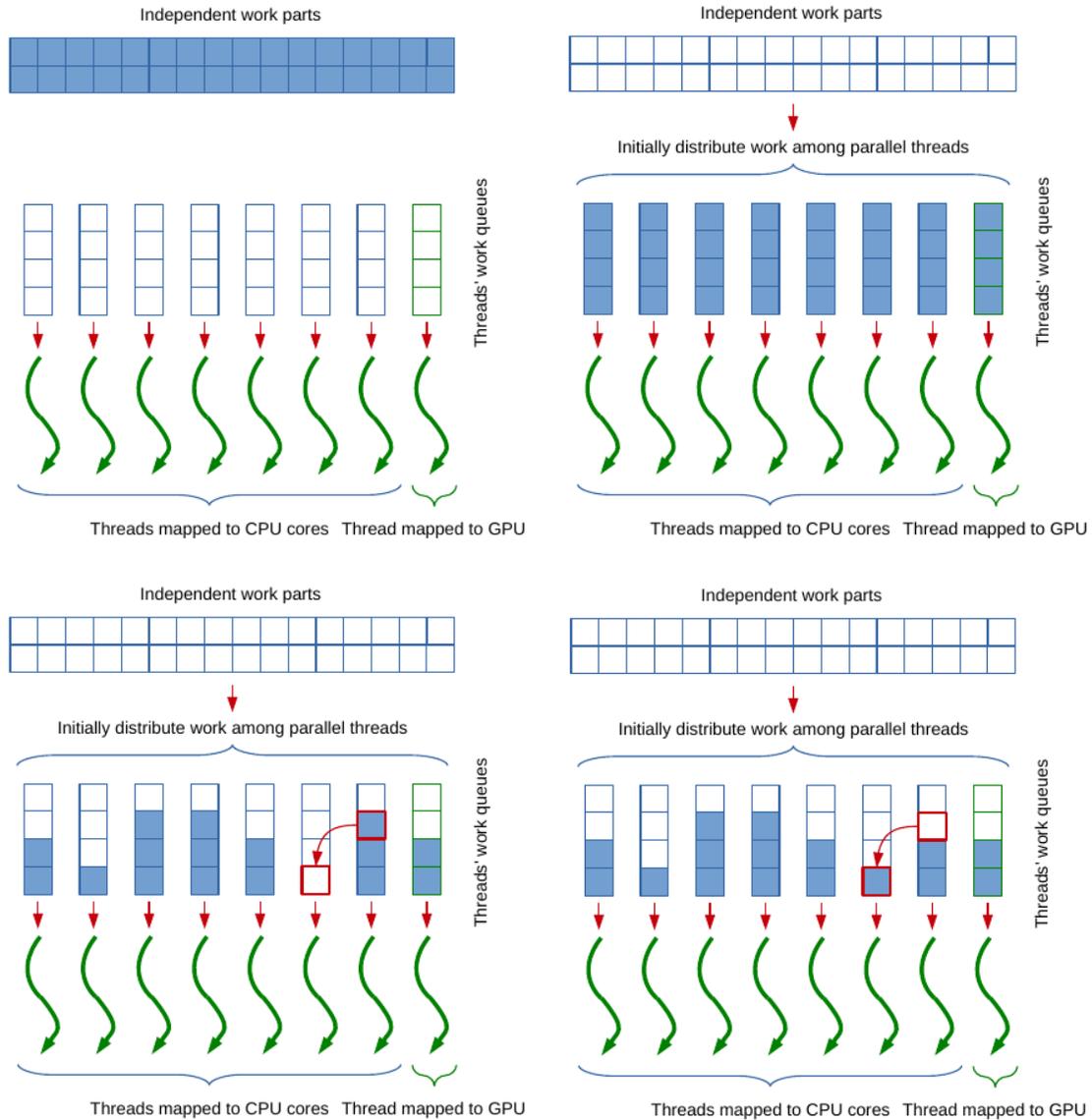


Figure 12: Hybrid multi-threading with Intel TBB: $(N - 1)$ threads on the CPU, 1 thread for the GPU; TBB balances workloads automatically using “work stealing”, as schematically indicated by the third thread from the right.

6.3 Results

We now present results for the model with a single (type of) agent per generation and a yearly, as well as a 5-yearly, calibration.

6.3.1 Parameters

To focus on the computational method instead of the calibration exercise, we take a simple parameterization that is, nevertheless, not too far away from standard calibrations. The chosen parameters for the yearly calibration are summarized in Tab. 2. Agents have a model lifetime of 60 periods, each corresponding to one year of life after the age of 20. We assume a very stylized, piecewise linear lifetime labor endowment profile, which is displayed in Fig. 13. Labor income doubles from age 20 to 40, stays constant from 40 to 60, then declines, and finally stays at a replacement rate of 30 percent of average working-age income until agents die at the age of 80. Concerning preferences, risk aversion is 3, the IES is $1/2$, and the discount factor is 0.96 resulting in a yearly equilibrium return to capital of about 4 percent. The capital share is set to 0.3. Depreciation is 8 percent in normal times, yet with 2 percent probability a disaster occurs that causes another 40 percent of the capital stock to depreciate. We assume that disaster events are i.i.d. and are also not correlated with the TFP shocks. The average TFP is set such that the equilibrium aggregate capital stock is about equal to the number of generations (given that the same is true for aggregate labor supply as implied by the above labor profile). The coefficient of variation of TFP equals 3 percent, resulting from two symmetric realizations around the mean. In contrast to the depreciation shock, TFP is persistent, the probability of staying in the good/bad regime from one year to the next being 80 percent.

To illustrate the scaling properties of our computational method below we also consider a version of the model with 12-period-lived agents. For this case, we scale all parameters to fit the 5-yearly calibration—in case of the disaster shock we keep the size as in the yearly calibration but increase the probability fivefold. As we explain below, it is important for the scaling exercise to be able to compute the solution on a single node in reasonable time, which is possible for the 12-generation model, but not for the 60-generation model.

6.3.2 Computational Performance

For the base case of 60-period-lived agents, the main computational challenge lies in the approximation of the policy functions that are defined on the 59-dimensional continuous state space of beginning of period asset holdings. As explained above, using adaptive sparse grids and high-performance computing facilities allows us to solve the basic projection problem. One annoyance known from lower-dimensional problems that becomes even more tedious in such high dimensions is the choice of the right box size, in this case for aggregate capital and the asset holdings of all

Parameter	Symbol	Value
Adult lifetime of agents	A	60
Discount factor	β	0.96
IES	$1/(1 - \rho)$	0.5
Risk aversion	$1 - \sigma$	3
Capital share	α	0.3
Disaster probability	$\mathbb{P}_t(z_{t+1} = 3, 4)$	0.02
Normal depreciation	$\delta(z = 1, 2)$	0.08
Disaster depreciation	$\delta(z = 3, 4)$	0.48
Average depreciation	$\bar{\delta}$	0.088
Persistence of disaster	$\mathbb{P}_t(z_{t+1} = 3, 4 z_t = 3, 4)$	0.02
Average aggregate productivity	$\bar{\xi}$	$(1 - \beta(1 - \bar{\delta})) / \alpha\beta$
Std. of productivity shocks	$STD(\xi(z)) / \bar{\xi}$	0.03
	$\mathbb{P}_t(z = 1, 3) = \mathbb{P}_t(z = 2, 4)$	0.5
	$\xi(z = 1, 3)$	$1.03 \bar{\xi}$
	$\xi(z = 2, 4)$	$0.97 \bar{\xi}$
Persistence of productivity	$\mathbb{P}_t(\xi(z_{t+1}) = \xi(z_t))$	0.8
Labor endowments	$l_a(z_t)$	see Fig. 13

Table 2: Choice of parameters for the OLG model with a yearly calibration, i.e., 60 generations.

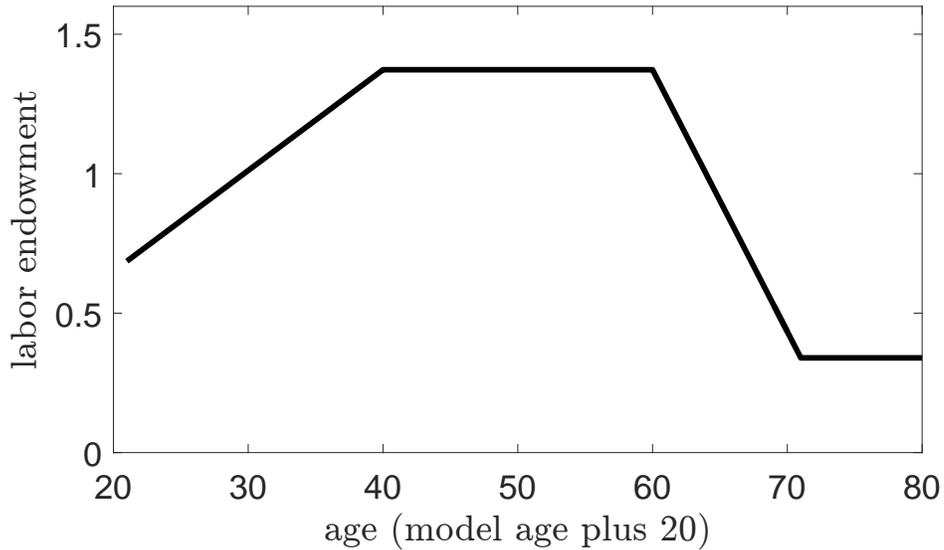


Figure 13: Life cycle labor endowment profile.

Number of gridpoints	Average Euler error
119	-1.79
1,457	-1.84
4,379	-1.88
7,081	-2.18
7,494	-2.20
10,345	-2.21

Table 3: Average Euler error as a function of the number of gridpoints (in log 10 scale) and decreasing ϵ , ranging from 1.25 down to 0.02.

generations. We choose a rather large range for the endogenous state, to ensure that *in simulations* the state always lies in the prescribed boxes. Unfortunately, given the size of the problem, we were unable to determine bounds that ensure that the state remains in the boxes with probability 1. In models with more than 20–30 endogenous continuous states it is also true that the accuracy of the solution can almost only be assessed along a simulated path, as sampling from the entire state space is either too costly or too coarse in such high-dimensional spaces.

As is to be expected, the accuracy of the solution depends crucially on the number of gridpoints used in the approximation scheme. In Tab. 3, we report the average Euler error (over a long simulation) as a function of the number of gridpoints (in log 10 scale) for a decreasing ϵ . The error decays as we increase the number of gridpoints used, with the lowest average error of around 0.61 percent obtained if one employs 10,345 points in each grid. With ten thousand points one therefore reaches acceptable error levels, but more points would be needed to achieve an error size that one would aim for with simpler models.

In the table, we are mixing the results from adaptive sparse grids and classical sparse grids.¹² A classical grid of level 3 has 7,081 points and achieves an error of 0.67 percent. This is not significantly lower than the minimal error achieved with the adaptive grids used. One important reason for this lies in the fact that we work with the same grid for each generation (but different grids across the 4 endogenous shocks). If points were added differently for different generations, adaptive sparse grids would have a stronger comparative advantage in this example.

To achieve the accuracy of a level 3 grid, running times on a single-core machine would be hundreds of days. It is therefore crucial for our approach that the basic time-iteration method can make use of high-performance computing. As it turns out, in our implementation the performance scales strongly with the number of dimensions. To illustrate this we take as a test problem the computation of a single discrete state within one timestep of an OLG model with 12 generations.

¹²As the size of the classical sparse grid grows very fast when the level increases—from 119 (L=2) to 7,081 (L=3) and then already 281,077 (L=4) points—the sparse grid allows us to look at intermediate numbers of gridpoints.

We have chosen this particular test case such that it can be comfortably run on a single node. For the large-scale OLG model with 60 generations, this was no longer feasible anymore. In order to provide a consistent benchmark, we used a nonadaptive (“classical”) sparse grid with non-zero boundary conditions (see Sec. 5.5) and refinement level 6. This instance has a total of 3,080,187 variables and constraints per discrete state. For our scaling experiments, we deploy the OLG model on the 5,272-node Cray XC30 “Piz Daint” system installed at the Swiss National Supercomputing Centre (CSCS). Cray XC30 compute nodes combine 8-core Intel Xeon E5-2670 (SandyBridge) CPUs with 1× NVIDIA Tesla K20X GPU. The OLG model is compiled with GNU compilers and CUDA Toolkit 5.0. The economic test case was solved with increasingly larger numbers of nodes (from 1 to 1,024 nodes). Fig. 14 shows the execution times and scaling on different levels and their ideal speedups. We used one MPI process per multi-threaded Intel SandyBridge node, of which each offloads part of the function evaluation to the K20x GPU. For this benchmark, the code scales nicely up to the order of 1,024 nodes, implying that the code will scale up to at least 4,096 nodes, as all four discrete states are independent (see Fig. 10). The dominant limitation to the strong scaling stems from the fact that with increasing node numbers the ratio of “points to be evaluated to MPI processes” is often smaller than one in the first few refinement levels. Moreover, the workload may be unbalanced in the case of large node numbers—for example, one MPI process gets two points to work on, while a second one obtains only one point to work on. The better parallel efficiency on the higher refinement levels is due to the fact that we have many more points available on such levels, so the workload is somewhat more fairly distributed among the different MPI processes. Thus, strong scaling efficiencies will be more evident in higher-dimensional models ($d > 11$), as the number of newly generated gridpoints grows faster with increasing refinement levels.

Note that our code framework, while enabling us to push the frontiers, required substantial development time. However, the trends in hardware design are moving in such a direction that even ordinary desktops will soon host combinations of dozens of general-purpose processors (possibly combined with co-processors and/or GPUs—see, e.g., Intel’s Xeon Phi¹³). Thus, in order to exploit these massive emerging resources, economists should start to think “parallel” more seriously. When porting applications to such hardware platforms, the entry points are still somewhat more cumbersome than using plain vanilla MATLAB. Nevertheless, the computer science community has started to make considerable efforts to facilitate this step. Two such projects, among others, are Julia <http://julialang.org> and Swift <http://swift-lang.org>.

6.3.3 The intergenerational wealth distribution

It is obviously computationally difficult to approximate equilibria for models where the endogenous state space is 59-dimensional. As we know from the work of Krusell and Smith (1997) it is often

¹³<http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html>.

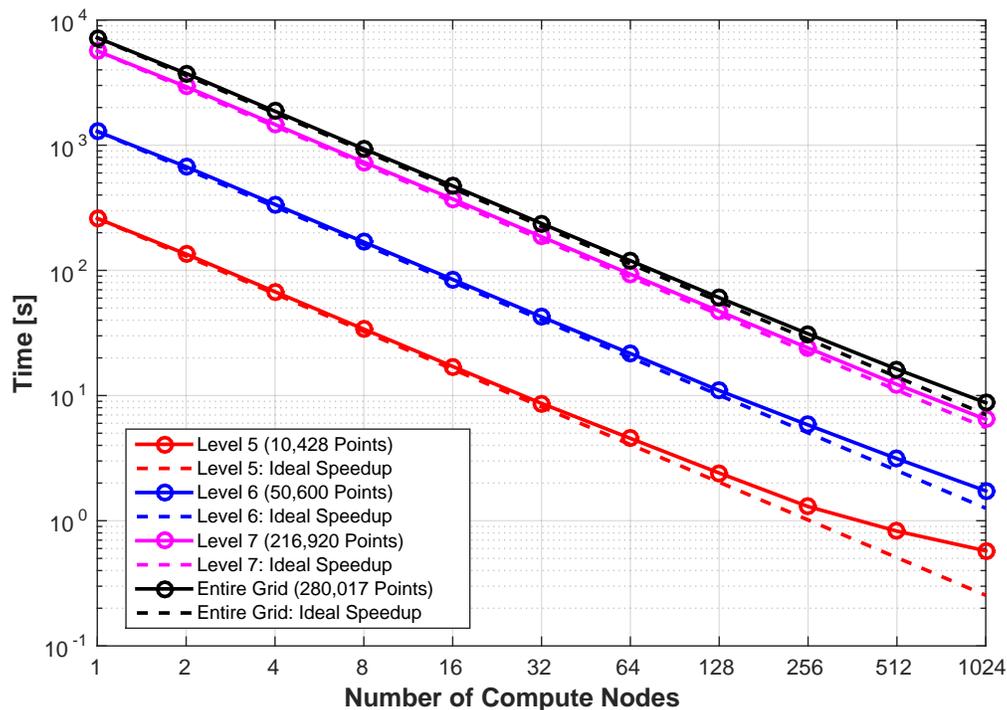


Figure 14: Strong scaling on "Piz Daint" for an OLG model with 12 generations and 4 discrete states using 6 levels of grid refinement (=7 levels) and in total 280,016 points. "Entire Grid" shows the entire computation time for the parallel solution of the nonlinear equations for up to 1,024 nodes. We also show the corresponding execution times for the computational subcomponents on different levels, e.g, for level 7 using 216,920 points, level 6 using 50,600 points, and level 5 using 10,428 points. Dashed lines show ideal speedups.

the case that a high-dimensional (or even infinite-dimensional) state can be reduced to a very-low-dimensional "pseudo-state", which (approximately) provides a sufficient statistic for the future evolution of the economy.

At this point, it is useful to note—however—that, unlike in Krusell and Smith (1997) the mean of capital alone does not provide a good approximation to the high-dimensional state. It is important to understand that movements in the intergenerational wealth distribution will generally have a large effect on prices—the only question is what kind of exogenous shocks lead to large movements of this distribution. Our calibration is obviously chosen in such a way that large movements are potentially possible: a large negative shock to capital reduces the savings of the old and middle-aged but has a relatively small effect on the young. Krueger and Kubler (2004) show that in models where agents live 10–20 periods, disentangling returns to capital and returns to labor in this manner leads to large movements in the intergenerational wealth distribution, which in turn implies that a very-low-dimensional sufficient statistic does not exist.

In our calibration agents live for 60 periods and, despite the fact that the shocks are large, the effect on the intergenerational wealth distribution is modest. Fig. 15 shows the impact of a disaster

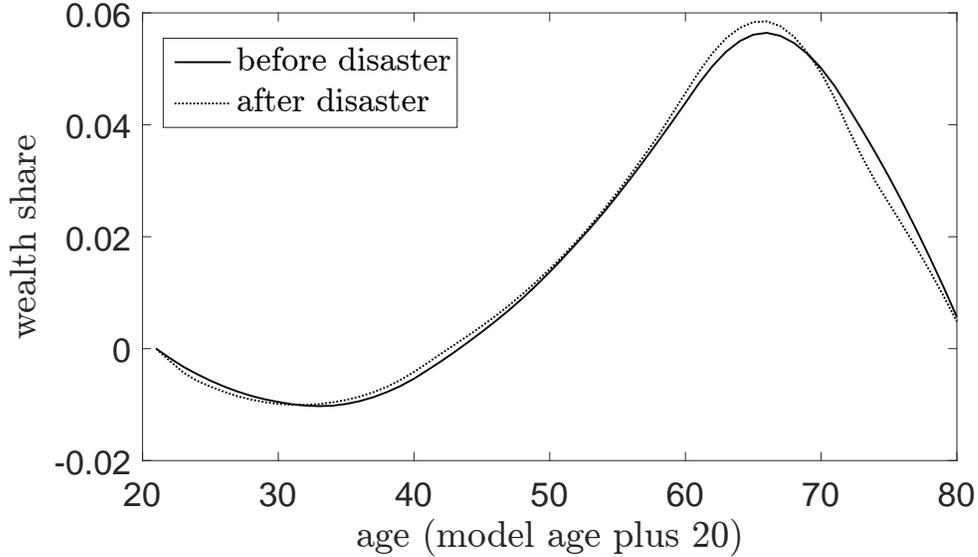


Figure 15: Change in wealth shares through the disaster shock (shock 4). The solid line represents the wealth distribution one period before the shock while the dotted line is the the wealth distribution two periods after the shock.

shock on the wealth distribution.¹⁴ As can be seen in the figure, a disaster shock does shift the wealth distribution, with the relatively old (above 70) losing significantly and the young gaining. However, since the negative shock also leads to high returns to capital, the middle-aged recover quickly and the overall effect is small.

It is for this reason that an approach as in Krusell and Smith (1997) that uses only the aggregate capital to forecast future prices does not perform too bad in this framework. The results are, however, significantly worse than in the original setup of Krusell and Smith (1997). If one runs a linear regression (for each state) to forecast tomorrow’s aggregate capital stock and the return from today’s aggregate capital stock, the R^2 for the four states are 0.998, 0.998, 0.996, and 0.995, respectively¹⁵. These numbers might seem high, yet it has been shown that even numbers much closer to one may be associated with large deviations from more accurate solutions (see Algan et al. (2014), who provide an excellent overview of the topic and explain different error criteria).

For our purposes it suffices to ask how large the forecasting errors are if one actually only uses aggregate capital for the forecast. In Fig. 16, we plot the forecasting error when the economy is in state 1 today as a function of the current capital stock. The forecasting error is highest when capital is low, which is the result of disaster shocks in the recent past. As can be seen in the figure, the errors are not huge, but certainly significant. One would generally think that a forecasting error of 10-20 basis points is difficult to incorporate into a rational expectations framework.

¹⁴Note that we plot the wealth shares, while the impact of a diaster on wealth levels is of course much bigger.

¹⁵Results are very similar (in fact, slightly worse) if one regresses log capital on lagged log capital.

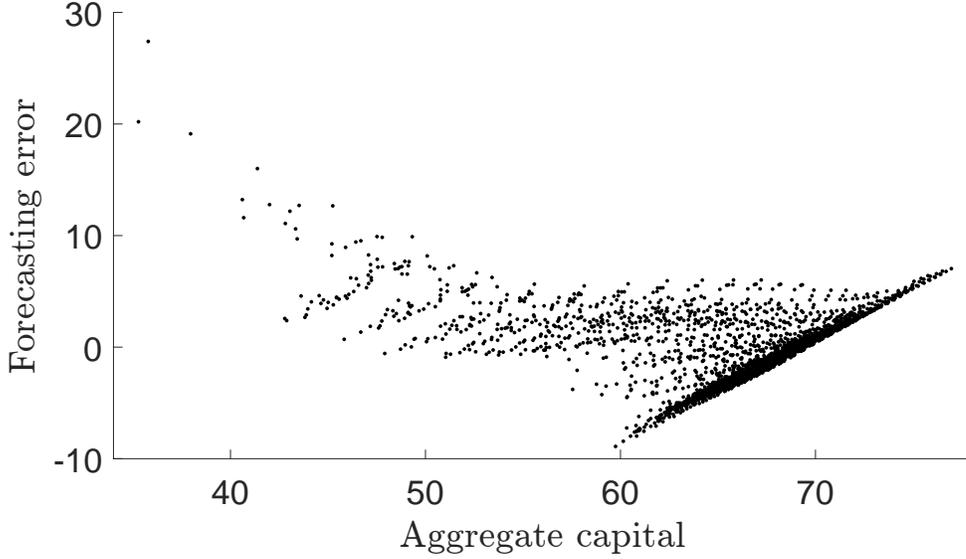


Figure 16: Error that agents make in forecasting the return to capital when they are in state 1. The error is reported in basis points.

6.4 Extensions: Several assets and occasionally binding constraints

For simplicity we focus on the case where agents trade only risky capital. In principal it is easy to extend the algorithm to allow for markets in financial assets, in particular for a bond market. For example, it is natural to assume that households can buy and sell one-period bonds $b_{y^a}(z^t)$ at price $q(z^t)$ today that pay one unit of consumption at each shock tomorrow. The main complication from the introduction of a bond arises from the fact that in most standard calibrations agents will hold unrealistically large asset positions. The Euler equations that pin down agents' portfolio choices tend to be numerically ill-conditioned and the time-iteration method becomes more computationally intensive. Alternatively it might be natural to assume that households cannot short capital, and face collateral constraints for shorting the bond: they can only short the bond if they hold capital as collateral. For simplicity we take the loan-to-value ratio as exogenous and fixed and denote it by LTV . The budget constraint of household y^a at node z^t therefore reads

$$c_{y^a}(z^t) + k_{y^a}(z^t) + q(z^t)b_{y^a}(z^t) = (1 + r(z^t))k_{y^{a-1}}(z^{t-1}) + b_{y^{a-1}}(z^{t-1}) + l_{y^a}(z^t)w(z^t) \quad (44)$$

$$\min(0, b_{y^a}(z^t))q(z^t) + LTV \cdot k_{y^a}(z^t) \geq 0, \quad (45)$$

where $k_{y^0} = b_{y^0} = k_{y^A} = b_{y^A} = 0$. Note that the no-short-sale constraint for capital is implicit in (45).

A competitive equilibrium now also consists of bond prices $q(z^t)$ and bond holdings $(b_{y^a}(z^t))$

that ensure that agents optimize and the bond markets clear

$$\sum_{a=1}^A \sum_{y^a \in \mathbf{Y}^a} \nu(y^a) b_{y^a}(z^{t-1}) = 0.$$

A FREE is defined analogously to the definition above; the Euler equations for the bond choice and market clearing for the bond market are included as additional equations. Time-iteration collocation methods can be used to approximate the FREE numerically. The algorithm is almost identical to the one outlined above, except that the policy functions now also include the bond policy and the bond price. Since we solve the first order conditions simultaneously, adding a bond simply doubles the number of equations at this point—the extra computational burden, while significant, is not prohibitive.

The collateral constraint (45) will occasionally be binding in equilibrium, leading to non-differentiabilities in policy functions. Brumm and Scheidegger (2014) address the consequences of this fact in detail. It seems reasonable to assert that for this case adaptive sparse grids are much more powerful than ordinary sparse grids. Nevertheless, for models where the state space is more than 20-dimensional, Brumm and Scheidegger (2014) find that even if one uses adaptive sparse grids, one would need very large numbers of gridpoints to properly identify all non-differentiabilities in policy and pricing functions. It is a subject of further research how to handle models, with occasionally binding constraints, that have a very-high-dimensional state space (e.g., OLG models with 60 generations).

References

- Aldrich, E. M. (2014). Chapter 10 - GPU computing in economics. In K. Schmedders and K. L. Judd (Eds.), *Handbook of Computational Economics Vol. 3*, pp. 557–598. Elsevier.
- Aldrich, E. M., J. Fernández-Villaverde, A. R. Gallant, and J. F. Rubio-Ramírez (2011). Tapping the supercomputer under your desk: Solving dynamic equilibrium models with graphics processors. *Journal of Economic Dynamics and Control* 35(3), 386–393.
- Algan, Y., O. Allais, and W. J. Den Haan (2010). Solving the incomplete markets model with aggregate uncertainty using parameterized cross-sectional distributions. *Journal of Economic Dynamics and Control* 34(1), 59–68.
- Algan, Y., O. Allais, W. J. D. Haan, and P. Rendahl (2014). Chapter 6 - solving and simulating models with heterogeneous agents and aggregate uncertainty. In K. Schmedders and K. L. Judd (Eds.), *Handbook of Computational Economics Vol. 3*, pp. 277–324. Elsevier.
- Auerbach, A. and L. Kotlikoff (1987). *Dynamic Fiscal Policy*. Cambridge University Press.
- Barro, R. J. (2006). Rare disasters and asset markets in the twentieth century. *The Quarterly Journal of Economics*, 823–866.

- Bellman, R. (1961). *Adaptive Control Processes: A Guided Tour*. Rand Corporation. Research studies. Princeton University Press.
- Bhandari, A., D. Evans, M. Golosov, and T. J. Sargent (2013). Taxes, debts, and redistributions with aggregate shocks. Technical Report 19470, National Bureau of Economic Research.
- Bloom, N., M. Floetotto, N. Jaimovich, I. Saporta-Eksten, and S. J. Terry (2012). Really Uncertain Business Cycles. Technical Report 18245, National Bureau of Economic Research.
- Brumm, J. and M. Grill (2014). Computing equilibria in dynamic models with occasionally binding constraints. *Journal of Economic Dynamics and Control* 38, 142–160.
- Brumm, J., M. Grill, F. Kubler, and K. Schmedders (2015). Collateral requirements and asset prices. *International Economic Review* 56(1), 1–25.
- Brumm, J. and F. Kubler (2014). Applying Negishi’s method to stochastic models with overlapping generations. *Working Paper, University of Zurich*.
- Brumm, J., D. Mikushin, S. Scheidegger, and O. Schenk (2015). Scalable high-dimensional dynamic stochastic economic modeling. *Journal of Computational Science* 11, 12–25.
- Brumm, J. and S. Scheidegger (2014). Using adaptive sparse grids to solve high-dimensional dynamic models. *Working Paper – Available at SSRN 2349281*.
- Bungartz, H.-J. and S. Dirnstorfer (2003). Multivariate quadrature on adaptive sparse grids. *Computing* 71, 89–114.
- Bungartz, H.-J. and M. Griebel (2004). Sparse grids. *Acta Numerica* 13, 1–123.
- Cai, Y., K. L. Judd, and T. S. Lontzek (2015). The social cost of carbon with economic and climate risks. *ArXiv e-prints*.
- Cai, Y., K. L. Judd, G. Thain, and S. J. Wright (2013). Solving dynamic programming problems on a computational grid. NBER Working Papers 18714, National Bureau of Economic Research.
- Chien, Y., H. Cole, and H. Lustig (2011). A multiplier approach to understanding the macro implications of household finance. *The Review of Economic Studies* 78(1), 199–234.
- Chien, Y. and H. Lustig (2010). The market price of aggregate risk and the wealth distribution. *The Review of Financial Studies* 23(4), 1596–1650.
- Coleman, W. J. (1990). Solving the stochastic growth model by policy-function iteration. *Journal of Business & Economic Statistics* 8(1), 27–29.

- Culler, D. E., A. Gupta, and J. P. Singh (1997). *Parallel Computer Architecture: A Hardware/Software Approach* (1st ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Den Haan, W. J. (2010). Comparison of solutions to the incomplete markets model with aggregate uncertainty. *Journal of Economic Dynamics and Control* 34(1), 4–27.
- Den Haan, W. J. and P. Rendahl (2010). Solving the incomplete markets model with aggregate uncertainty using explicit aggregation. *Journal of Economic Dynamics and Control* 34(1), 69–78.
- Dongarra, J. J. and A. J. van der Steen (2012). High-performance computing systems: Status and outlook. *Acta Numerica* 21, 379–474.
- Dumas, B. and A. Lyasoff (2012). Incomplete-market equilibria solved recursively on an event tree. *The Journal of Finance* 67(5), 1897–1941.
- Epstein, L. G. and S. E. Zin (1989). Substitution, risk aversion, and the temporal behavior of consumption and asset returns: A theoretical framework. *Econometrica* 57(4), 937–969.
- Evans, D. G. (2015). *Perturbation theory with heterogeneous agents: Theory and applications*. Ph. D. thesis, New York University.
- Favilukis, J., S. C. Ludvigson, and S. V. Nieuwerburgh (2010). The macroeconomic effects of housing wealth, housing finance, and limited risk-sharing in general equilibrium. Working Paper 15988, National Bureau of Economic Research.
- Feldman, M. and C. Gilles (1985). An expository note on individual risk without aggregate uncertainty. *Journal of Economic Theory* 35(1), 26–32.
- Flynn, M. J. (1972). Some computer organizations and their effectiveness. *IEEE Trans. Comput.* 21(9), 948–960.
- Garcke, J. and M. Griebel (2012). *Sparse Grids and Applications*. Lecture Notes in Computational Science and Engineering Series. Springer.
- Harenberg, D. and A. Ludwig (2014). Social security and the interactions between aggregate and idiosyncratic risk. SAFE Working Paper Series 59, Research Center SAFE - Sustainable Architecture for Finance in Europe, Goethe University Frankfurt.
- Heer, B. and A. Maussner (2009). *Dynamic general equilibrium modeling: computational methods and applications*. Springer.
- Heinecke, A. and D. Pflueger (2013). Emerging architectures enable to boost massively parallel data mining using adaptive sparse grids. *International Journal of Parallel Programming* 41(3), 357–399.

- Judd, K. L. (1992). Projection methods for solving aggregate growth models. *Journal of Economic Theory* 58(2), 410–452.
- Judd, K. L. (1998). *Numerical methods in economics*. The MIT press.
- Judd, K. L., F. Kubler, and K. Schmedders (2003). Computational methods for dynamic equilibria with heterogeneous agents. In *Advances in economics and econometrics: Theory and applications, Eighth World Congress*, Volume 3, pp. 243. Cambridge University Press.
- Judd, K. L., L. Maliar, S. Maliar, and R. 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.
- Juillard, M. (2003). Dynare: A program for solving rational expectations models.
- Khan, A. and J. K. Thomas (2013). Credit shocks and aggregate fluctuations in an economy with production heterogeneity. *Journal of Political Economy* 121(6), 1055–1107.
- Klimke, A. and B. Wohlmuth (2005). Algorithm 847: Spinterp: piecewise multilinear hierarchical sparse grid interpolation in matlab. *ACM Trans. Math. Softw.* 31(4), 561–579.
- Kollmann, R., S. Maliar, B. A. Malin, and P. Pichler (2011). Comparison of solutions to the multi-country real business cycle model. *Journal of Economic Dynamics and Control* 35(2), 186–202.
- Kreps, D. M. and E. L. Porteus (1978). Temporal resolution of uncertainty and dynamic choice theory. *Econometrica*, 185–200.
- Krueger, D. and F. Kubler (2004). Computing equilibrium in OLG models with stochastic production. *Journal of Economic Dynamics and Control* 28(7), 1411–1436.
- Krueger, D. and F. Kubler (2006). Pareto-Improving Social Security Reform when Financial Markets are Incomplete!?. *American Economic Review* 96(3), 737–755.
- Krueger, D., K. Mitman, and F. Perri (2015). Macroeconomics and heterogeneity, including inequality. *forthcoming in: J.B. Taylor and H. Uhlig (eds.), Handbook of Macroeconomics Vol. 2*.
- Krusell, P. and A. A. Smith (1997). Income and wealth heterogeneity, portfolio choice, and equilibrium asset returns. *Macroeconomic Dynamics* 1(2), 387–422.
- Krusell, P. and A. A. Smith, Jr (1998). Income and wealth heterogeneity in the macroeconomy. *Journal of Political Economy* 106(5), 867–896.
- Kubler, F. and K. Schmedders (2005). Approximate versus exact equilibria in dynamic economies. *Econometrica* 73(4), 1205–1235.

- Ma, X. and N. Zabararas (2009). An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Comput. Phys.* 228(8), 3084–3113.
- Ma, X. and N. Zabararas (2010). An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *J. Comput. Phys.* 229(10), 3884–3915.
- Maliar, L. and S. Maliar (2014). Chapter 7 - numerical methods for large-scale dynamic economic models. In K. Schmedders and K. L. Judd (Eds.), *Handbook of Computational Economics Vol. 3*, pp. 325–477. Elsevier.
- Maliar, S., L. Maliar, and K. Judd (2011). Solving the multi-country real business cycle model using ergodic set methods. *Journal of Economic Dynamics and Control* 35(2), 207–228.
- McKay, A. and R. Reis (2013). The role of automatic stabilizers in the u.s. business cycle. Working Paper 19000, National Bureau of Economic Research.
- Mertens, T. M. and K. L. Judd (2013). Equilibrium existence and approximation for incomplete market models with substantial heterogeneity. *Working Paper – Available at SSRN 1859650*.
- Murarasu, A., J. Weidendorfer, G. Buse, D. Butnaru, and D. Pflüeger (2011). Compact data structure and parallel algorithms for the sparse grid technique. In *16th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*.
- Pflüger, D. (2010). *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Ph. D. thesis, München.
- Pflüger, D. (2012). Spatially adaptive refinement. In J. Garcke and M. Griebel (Eds.), *Sparse Grids and Applications*, Lecture Notes in Computational Science and Engineering, Berlin Heidelberg, pp. 243–262. Springer.
- Pichler, P. (2011). Solving the multi-country real business cycle model using a monomial rule Galerkin method. *Journal of Economic Dynamics and Control* 35(2), 240–251.
- Proehl, E. (2015). Computing the cross-sectional distribution to approximate stationary markov equilibria with heterogeneous agents and incomplete markets. *Working Paper – Available at SSRN 2620937*.
- Rabitz, H. and O. F. Alis (1999). General foundations of high-dimensional model representations. *Journal of Mathematical Chemistry* 25(2–3), 197–233.
- Reinders, J. (2007). *Intel Threading Building Blocks* (First ed.). Sebastopol, CA, USA: O’Reilly & Associates, Inc.

- Reiter, M. (2009). Solving heterogeneous-agent models by projection and perturbation. *Journal of Economic Dynamics and Control* 33(3), 649–665.
- Reiter, M. (2010). Solving the incomplete markets model with aggregate uncertainty by backward induction. *Journal of Economic Dynamics and Control* 34(1), 28–35.
- Sager, E. (2014). Solving the incomplete markets model with aggregate uncertainty: The method of mixtures. *Working Paper, Bureau of labor statistics*.
- Skjellum, A., W. Gropp, and E. Lusk (1999). *Using MPI*. MIT Press.
- Spear, S. E. (1988). Existence and local uniqueness of functional rational expectations equilibria in dynamic economic models. *Journal of Economic Theory* 44(1), 124–155.
- Storesletten, K., C. I. Telmer, and A. Yaron (2007). Asset pricing with idiosyncratic risk and overlapping generations. *Review of Economic Dynamics* 10(4), 519–548.
- Terry, S. (2014). Alternative methods for solving heterogeneous firm models. *Working Paper, Stanford University*.
- Waechter, A. and L. T. Biegler (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* 106(1), 25–57.
- Winschel, V. and M. Kraetzig (2010). Solving, estimating, and selecting nonlinear dynamic models without the curse of dimensionality. *Econometrica* 78(2), 803–821.
- Young, E. R. (2010). Solving the incomplete markets model with aggregate uncertainty using the Krusell–Smith algorithm and non-stochastic simulations. *Journal of Economic Dynamics and Control* 34(1), 36–41.
- Zenger, C. (1991). Sparse grids. In W. Hackbusch (Ed.), *Parallel Algorithms for Partial Differential Equations*, Volume 31 of *Notes on Numerical Fluid Mechanics*, pp. 241–251. Vieweg.