From the holdings of the Department of Special Collections , Princeton University Library, Princeton, NJ

These documents are either in the public domain, or can only be used for educational and research purposes (Fair use) as per U.S. Copyright law (text below). By accessing this file, all users agree that their use falls in the public domain or within fair use as defined by the copyrightlaw. We do not charge any permission or usage fees for the publication of images of material inour collections, including those provided by Princeton University Library, via our website, catalog, or directly from staff.For more information regarding copyright, credit, and citation please review the information onthe Copyright, Credit and Citation Guidelines page of our website.
https://library.princeton.edu/special-collections/policies/copyright-credit-and-citation-guidelines. If you have any questions, including how to order higher-quality images from the collection, contact Special Collections Public Services via our Ask Us! Form. https://library.princeton.edu/special-collections/ask-us

---

Doctoral Dissertations, 2011-2026 Economics

2017

Approximation Methods in Continuous Time Economic Models

Ahn, SeHyoun

Princeton, NJ : Princeton University

---

http://arks.princeton.edu/ark:/88435/dsp01qf85nd90d

U.S. Copyright law: The copyright law of the United States (Title 17, §108, United States Code) governs the making of photocopies or other reproductions of copyrighted material. Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be used for any purpose other than private study, scholarship, or research. If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of fair use that user may be liable for copyright infringement. This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve a violation of copyright law. https://www.copyright.gov/title17/title17.pdf

Approximation Methods in Continuous Time Economic Models

SeHyoun Ahn

A Dissertation

Presented to the Faculty

of Princeton University

in Candidacy for the Degree

of Doctor of Philosophy

Recommended for Acceptance

by the Department of

Economics

Adviser: Benjamin Moll

June 2017

© Copyright by SeHyoun Ahn, 2017. All rights reserved.

Abstract

As the number of state variables increases, solving heterogeneous agent problems become infeasible. Hence, this dissertation studies various methods of increasing the feasibility frontier. The following three chapters present *(i)* a method of general function approximations, *(ii)* a model reduction method for Hamilton-Jacobi-Bellman (HJB) equations in continuous time, and *(iii)* a model reduction method for solving heterogeneous agent models with aggregate shocks.

Chapter 1 examines approximations of general functions on a linear subspace. It presents how many different approximations used by economists can be considered as an approximation on a linear subspace, and presents concise and efficient implementations that would allow researchers of heterogeneous agents models to build custom approximations instead of relying on a pre-existing black box codes written by others.

Chapter 2 studies how to solve continuous time dynamical programming problems more efficiently. Even with efficiency gains from the continuous time framework over the discrete time counterpart, the curse of dimensionality makes solving HJB equations impossible for models with a moderate number of state variables. In this chapter, I show how continuous time HJB equations can be approximated on a linear subspace and implement the Smolyak method as an example. The benefits of modeling in continuous time are preserved under the reduction, and hence, the Smolyak method can be applied without nonlinear solvers nor numerical quadratures in comparison to discrete time models.

Chapter 3, co-authored with Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf, studies how to solve heterogeneous agent models with aggregate shocks. With aggregate shocks, the distribution of agents become state variables. As the distribution is infinite-dimensional, the literature has resorted to problem-specific reduction methods like Krusell and Smith (1998) that rely on the approximate aggregation. Using the model reduction literature from engineering, this chapter introduces a model-free method to solve heterogeneous agent models with aggregate shocks. This new method gives around 1000-fold speed gain over the next best algorithm in the literature while achieving the best approximation error.

Acknowledgements

I will be forever indebted to my adviser, Benjamin Moll. He has been the most patient with my random drop-ins even on his busiest days, and without his support, I would not have been able to develop as the researcher I am today. I also thank Faruk Gul for advising me to set aside my mathematical hammer to look at the world as an economist. The world would not have been the same if I still had my head down looking for nails. I also thank Sylvain Chassang and Chris Sims for encouraging me to pick up my hammer and build again when I was lost looking at the world for the first time. Without their encouragement, I would have still been lost looking at the magnificence of the world and the smallness of self. In all, I thank all the heros who saved a bumbling fool lost in a forest whacking at a random tree convinced that it is a nail. I also thank Brian Conrad and Stephen DeBacker for helping me forge my hammer in the first place.

I would also like to thank my co-authors, Greg Kaplan, Thomas Winberry, and Christian Wolf. Their belief in me helped me solve seemingly impossible problems.

I thank Laura Hedden for her fantastic administrative and moral support.

My memory of the basement of Fisher and Julis Romo Rabinowitz Building will shine brightly because of my friends who made my day everyday. In particular, I thank Michael Dobrew, Paul Ho, Sungki Hong, Lasse Mononen, Donghwa Shin, Sharon Traiberman, and CJ Verbeck for the stimulating academic and non-academic conversations. Outside of the economics department, I would like to thank Boris Alexeev, Rachel Kogan, Michael McBreen, Sam Ruth, Kevin Wilson, and Percy Wong for the fond memory of board games and Doctor Who.

Last, for the most important, I thank my parents and brother. Without their endless support, this dissertation would not have been possible.

To my supportive family.

Contents

Abstract iii

Acknowledgements iv

List of Figures ix

1 What Do Polynomials, Splines, and Smolyak Methods Have in Common? 1

1.1 Intuition 1

1.2 Approximation on Linear Subspace 4

1.2.1 Implementation 6

1.3 One Dimension 9

1.3.1 Global Construction 9

1.3.2 Piecewise Construction without Pasting Conditions 11

1.3.3 Piecewise Construction with Pasting 15

1.3.4 Relationship with Interpolation 20

1.4 Higher Dimensions 23

1.4.1 Tensor Construction 24

1.4.2 Piecewise Construction in Higher Dimensions 27

1.4.3 Non-Tensor Product Construction 33

1.5 Conclusion 36

1.6 Codes Appendix 36

2 Projection Methods in Solving Continuous Time Hamilton-Jacobi-Bellman Equations 41

2.1 Introduction . . . 41
2.2 Problem Formulation . . . 43
2.3 Solution on a Linear Subspace . . . 44
2.4 Example Applications . . . 46
2.4.1 Tensor Constructions . . . 47
2.4.2 Generalized Total Degree . . . 50
2.4.3 Adaptivity . . . 55
2.5 Conclusion . . . 58

3 When Inequality Matters for Macro and Macro Matters for Inequality 59
3.1 Introduction . . . 59
3.2 Linearizing Heterogeneous Agent Models . . . 65
3.2.1 Model Description . . . 67
3.2.2 Linearization Procedure . . . 69
3.2.3 What Does Linearization Capture and What Does It Lose? . . . 74
3.2.4 Performance of Linearization in Krusell-Smith Model . . . 77
3.3 Model Reduction . . . 80
3.3.1 Overview of Distribution Reduction . . . 81
3.3.2 Distribution Reduction in Simplified Model . . . 84
3.3.3 Distribution Reduction in Full Model . . . 91
3.3.4 Value Function Reduction . . . 94
3.3.5 Putting It All Together: A Numerical Toolbox . . . 95
3.3.6 Model Reduction in Krusell-Smith Model . . . 97
3.4 Aggregate Consumption Dynamics in Two-Asset Model . . . 100
3.4.1 Model . . . 101
3.4.2 Calibration . . . 106
3.4.3 Computation . . . 113
3.4.4 Aggregate Consumption Dynamics . . . 115

3.4.5 Distributional Implications of TFP Growth Shocks 121
3.5 Richer Macro and Inequality Interactions 123
3.5.1 Augmented Production Structure 124
3.5.2 Capital-Specific Productivity Shocks 125
3.5.3 Unskilled-Specific Productivity Shocks 126
3.6 Conclusion 127
3.7 Appendix 129
3.7.1 Fully Recursive Formulation of 129
3.7.2 Connection to Linearization of Representative-Agent Models 131
3.7.3 Model Reduction 133
3.7.4 Detrending the Two-Asset Model 137

Bibliography **142**

List of Figures

1.1 Approximation with different degree polynomials 11
1.2 Local approximation 13
1.3 Local approximation with Pasting Conditions 19
1.4 2D quadratic spline without the pasting conditions 31
1.5 2D quadratic spline with the pasting conditions 32
1.6 The structure of continuity constraints 32
1.7 Tent (hierarchical surplus/locally linear) functions 34

2.1 Reduction in the skill direction 48
2.2 Reduction in the asset direction 49
2.3 Tensor product basis function 1 50
2.4 Tensor basis function 2 51
2.5 Anisotropic total polynomial degree basis function 53
2.6 Isotropic total polynomial degree basis function 54
2.7 Anisotropic tent function basis function 55
2.8 Inclusion pattern of polynomials 57
2.9 Errors from approximation using different polynomial basis functions 57

3.1 Impulse Responses to TFP Shock in Krusell-Smith Model 97
3.2 Basis Vectors in Distribution Reduction 98
3.3 Internal Consistency Check 100

3.4 Calibrated Adjustment Cost Function 110
3.5 Liquid and Illiquid Wealth Distribution in Steady State 111
3.6 Heterogeneity in MPCs Across Households 112
3.7 Impulse Responses for Different Orders of Distribution Reduction 113
3.8 Basis Vectors for Approximating Distribution in Two-Asset Model 114
3.9 Impulse Responses to TFP Growth Shock in Two-Asset Model 116
3.10 Distributional Responses to TFP Growth Shock in Two-Asset Model 122
3.11 Impulse Responses to Capital-Specific Productivity Shock 127
3.12 Impulse Responses to Unskilled Labor-Specific Productivity Shock 128

Chapter 1

What Do Polynomials, Splines, and Smolyak Methods Have in Common?

1.1 Intuition

Although it might be natural to represent an object in high-dimensional space, the intrinsic dimensionality of the object might be small. As a trivial example, consider a function defined by

$$f(x, y) = x \qquad\qquad x, y \in \mathbb{R}.$$

Although the function is defined over $\mathbb{R}^2$, only x needs to be known to be able to evaluate the function exactly. Hence, if we were to create an equivalence class with the equivalence relation $(x, 0) \sim (x, y)$ for all $y \in \mathbb{R}$; i.e., if we treat all (x, y) with the same value of x as the same, then we can carry only a one-dimensional object. We define the function over the equivalence classes[1] by

$$\widetilde{f}(\pi((x, 0))) = x \qquad\qquad x \in \mathbb{R}.$$

[1] sets of things that we consider to be equal

where $\pi(t)$ denotes the equivalence class containing t. Hence, instead of having to represent a two-dimensional function, we can just represent a one-dimensional function. In addition, given any point (x, y) and an approximate function $\widetilde{f}$, the approximation for the value can be computed with

$$\widetilde{f}(\pi((x, y))) = \widetilde{f}(\pi((x, 0))).$$

This case is so simple that the notation is more confusing than the concept. However, consider the simplest possible generalization of the above function given by

$$g(x, y) = a \cdot x + b \cdot y \qquad\qquad x, y \in \mathbb{R}.$$

Although this is linear and simple, it is not straight forward how the equivalence classes should be defined. Let's take the equivalence classes given by the equivalence relation

$$\begin{bmatrix} a \\ b \end{bmatrix} c + \begin{bmatrix} -b \\ a \end{bmatrix} t \sim \begin{bmatrix} a \\ b \end{bmatrix} c + \begin{bmatrix} -b \\ a \end{bmatrix} s \qquad\qquad \text{for } c, t, s \in \mathbb{R},$$

If we were to define a function over the equivalence classes by

$$\widetilde{g}\left(\pi\left(\begin{bmatrix} a \\ b \end{bmatrix} c + \begin{bmatrix} -b \\ a \end{bmatrix} t\right)\right) = (a^2 + b^2)c \qquad\qquad \text{for } c \in \mathbb{R}$$

then we get the correct value for the function although our current function is being represented as a one-dimensional function.[2] However, this is not the end of the line of the approximation problems arising in economics. For certain problems, the given function to be approximated is

[2]It is just straightforward algebra to check that the $\widetilde{g}$ gives the exact approximation of g.

time dependent. For example, suppose that our given function is

$$h_t(x, y, z) = a \cdot x + b \cdot y + c_t \cdot z \qquad\qquad x, y, z, t \in \mathbb{R}.$$

In this case, the equivalence class defined by

$$\begin{bmatrix} a \\ b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v + \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix} w \sim \begin{bmatrix} a \\ b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v + \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix} s \qquad \text{for } u, v, w, s \in \mathbb{R}$$

gives the exact reduction for all possible functions for any given time t with the function defined by

$$\tilde{h}_t \left(\pi \left(\begin{bmatrix} a \\ b \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} v + \begin{bmatrix} -b \\ a \\ 0 \end{bmatrix} w \right) \right) = (a^2 + b^2)u + c_t \cdot v$$

$$\text{for } u, v, t \in \mathbb{R}.$$

Hence, the given family of three-dimensional functions $\{h_t\}_{t \in \mathbb{R}^+}$ can be represented as a two-dimensional object.

As the approximation theory is a large field with problem specific optimizations, this chapter will not review all techniques in approximation theory. For example, the model reduction literature for the linear systems tackles this problem to find a systematic way generates the equivalence classes and functions on the equivalent classes. This is applied to economic problems in chapter 3. However, the model reduction literature imposes a very restrictive structure on the problem that limits applicability to a small subset of problems. This chapter will give a practical guide for creating basis functions when the problem does not fit into the form required by the model

reduction literature while requiring that it is fast and simple on computers.[3] This chapter might not be new for most people on an intuitive level. However, in this chapter, I intend to use the subspace framework to unify different approaches that people take and provide implementations of each so that people do not have to rely on pre-implemented programs that tend to behave like a black box for most researchers. For the presentation, MATLAB codes will be used, but they can be simply generalized to Python and Julia.[4] For example, implementing the cubic spline (in one-dimension) takes only around 15 lines of code. It will be shown how these approximations can be generalized to higher dimensions effectively. For example, a simple generalization of a one-dimensional approximation to higher dimensions takes only one line of code per dimension. The rest of the chapter is structured as follows. Section 1.2 will define notations for the subspace approximation, section 1.3 will apply the idea to the one dimension, and section 1.4 will generalize higher dimensions. Section 1.5 will summarize.

1.2 Approximation on Linear Subspace

Consider functions

$$f : \mathfrak{D} \to \mathbb{R}$$

in a Banach space[5] $\mathfrak{F}$ with norm $\|\cdot\|$. Our problem is for a given subset $\mathfrak{G} \subset \mathfrak{F}$ of functions, creating a way of approximating the functions in a systematic way. It is possible to represent this function by storing the value of the function for every point in the domain. However, even if this were feasible, it is not very efficient depending on the properties of $\mathfrak{G}$. For example, if $\mathfrak{G}$ is a set of constant functions, then only one value needs to be stored to represent the function exactly. In this chapter, we consider approximations by considering (linear) subspace spanned by basis functions.

[3] If the linear systems set up is applicable, the techniques found in that literature should be used since they will give more optimized reductions.

[4] The codes are available at `http://www.sehyoun.com`

[5] (complete) normed vector space over functions

Hence, given a set of basis functions $\{\phi_i\}_{i=1,\ldots,n}$, we will approximate a given function by

$$\tilde{f}(x) = \sum_{i=1}^{n} a_i \phi_i(x)$$

such that

$$\left\| \tilde{f} - f \right\| = \min_{b_i} \left\| \sum_{i=1}^{n} b_i \phi_i - f \right\|.$$

Hence, given the basis functions, approximations can be stored with a vector $a \in \mathbb{R}^n$. Although any norm can be used in theory, using the (weighted) L^2 norm simplifies the problem computationally. This is analogous to why statisticians uses a quadratic loss function in linear regressions. For the remainder of the chapter, the L^2 norm for the relevant domain is used. Hence, given a (weighted) L^2 norm,[6] the optimal approximation problem becomes

$$\min_{b_i} \int_{\mathfrak{D}} \left(f(x) - \sum_{i=1}^{n} b_i \phi_i(x) \right)^2 \omega(x)\, dx$$

where $\omega(x)$ is a weight function. The first-order condition of the minimization problem gives

$$\int_{\mathfrak{D}} 2 \left(f(x) - \sum_{i=1}^{n} b_i \phi_i(x) \right) \phi_i(x) \omega(x)\, dx = 0 \qquad \text{for } i = 1, \ldots, n.$$

This just means that we are computing b_i by projections.[7] Given that b_i's can be computed by projections, the only task remaining is coming up with good basis functions. The properties of the subset of functions, $\mathfrak{G}$, to be approximated will dictate what a good set of basis functions would be. It is impossible to create one family of basis functions that necessarily dominates other families for all $\mathfrak{G}$,[8] and the basis functions would need to be selected for specific problems.

[6] $\int$ notation stands for the summation for finite domain $\mathfrak{D}$.

[7] A reminder of how this is done in a computer will be given in a later section.

[8] The same exact analogue exists in econometrics.

1.2.1 Implementation

Although functions can be defined over an uncountable domain, functions can be evaluated on only a finite number of points on a computer. Therefore, language at the generality of the continuum is not necessary when considering the implementation on a computer. Therefore, as a canonical notation, let x be an $(n \times 1)$ vector of points to evaluate the function. Then, each function can be represented as the $(n \times 1)$ vector of the function values at x; i.e., f can be represented by a $(n \times 1)$ vector

$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix}$$

As a basis function is also just a function, basis functions can also be represented by an $(n \times 1)$ vector. In addition, given this vector representation, representing a set of m basis functions is also straightforward. They can be represented by a $(n \times m)$ matrix, Φ, with each column as a basis functions, i.e.,

$$\Phi = \begin{pmatrix} \phi_1(x) & \phi_2(x) & \dots & \phi_m(x) \end{pmatrix} = \begin{pmatrix} \phi_1(x_1) & \phi_2(x_1) & \dots & \phi_m(x_1) \\ \phi_1(x_2) & \phi_2(x_2) & \dots & \phi_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_1(x_n) & \phi_2(x_n) & \dots & \phi_m(x_n) \end{pmatrix}$$

Computing the L^2 norm numerically on the proper domain requires a numerical integration. However, if the problem contains a set of points of interest given as x, there is a natural approximation for the weighted L^2-norm. The sum of the squared errors at x can be used as the

approximation for the error; i.e.,

$$\|f\| = \sum_{i=1}^{n} [f(x_i)]^2$$

Hence, given this norm, the calculation of the projection becomes completely analogous to the linear regression setting that economists use every day. Given the function $f(= y)$, and basis functions $\Phi(= X)$, approximating $b(= \beta)$ can be computed by

$$b = \left(\Phi^T \Phi\right)^{-1} \Phi^T f.$$

The approximated function values can be evaluated by[9]

$$\widetilde{f} = \Phi b.$$

Many times, basis functions are built using intuition for how the functions behave, but the resulting basis functions are not always linearly independent. This can happen due to the machine accuracy limit in representing the real number. Hence, although the function might have been linearly independent, they can become linearly dependent on the evaluation points. This is completely analogous to how multicollinearity can occur in econometrics though there can be intuition for why different factors might be linearly independent. However, unlike the econometrics case, this is not a problem when creating basis functions. When basis functions are built for approximations, only the subspace spanned by the basis functions matters, not the actual basis functions. Hence, linearly dependent basis functions can be dropped to get proper basis functions. Numerically, this can be done by computing the (economical) singular value decomposition (SVD)[10] and keeping the vectors with high enough singular values. In MATLAB, this can be implemented as

```
crit = 1e-15;                % Example Cutoff value
```

[9]This is the same as computing the predicted value for linear regression.

[10]Most programs contain implementations of the SVD. Otherwise, any orthonormalization can be used, e.g., the modified Gram-Schmidt algorithm.

```
% Compute SVD
[u,d,~] = svd(Phi,'econ');

% Compute number of singular values bigger than cutoff
n = sum(diag(d)>crit);

% Keep basis vectors with large enough singular values
Phi = u(:,1:n);
```

Since the SVD already does orthonormalization ($\Phi^T\Phi = I_m$), the computation simplifies to

$$b = (\Phi^T\Phi)^{-1}\Phi^T f = (I_m)^{-1}\Phi^T f = \Phi^T f$$

Hence, the resulting computation on subspace requires only matrix multiplications for projections and blowups:

$$b = \Phi^T f \qquad\qquad \widetilde{f} = \Phi b.$$

Last, under the evaluation point setup, there is a canonical set of basis functions. Φ can be taken to be a $(n \times n)$ identity matrix, i.e., each basis function contains function values at n evaluation points of x. Many times, the number of basis functions should be small, $m \ll n$, to improve the computational speed and/or the memory efficiency. Note, however, that under identity basis functions, there is no reduction in representing the function.

1.3 One Dimension

Now that the formal setup is done, we are ready to construct basis functions that behave well. As always, it is simple to start in one dimension. Hence, the function of interest to be approximated is

$$f : [0, 1] \to \mathbb{R}$$

1.3.1 Global Construction

One of the first approximations that people learn to use is the polynomial representation. Although rarely presented as such, polynomial approximations are also approximations on a linear subspace. A polynomial approximation is given by

$$\widetilde{f}(x) = \sum_{i=0}^{n} a_i x^i$$

This is just a linear combination of basis functions given by

$$\{x^i\}_{i=0,1,\ldots,n}$$

Hence, the approximation by the n^{th}-degree polynomial can be considered a linear approximation on a (n+1)-dimensional subspace. Polynomial basis functions approximate smooth functions very well. In fact, the Stone-Weierstrass theorem tells us the function can be approximated within $\varepsilon > 0$ in sup-norm.[11]

Last, note that using the k^{th}-degree polynomial is equivalent to approximation that uses $(k+1)$ Chebyshev polynomials becuase both sets of the basis functions span the same subspace. The only difference is the Chebyshev polynomials are orthogonalized by construction, and have a simple

[11] Stronger than the L^2 norm considered in this chapter.

solution for the minimization problem.[12] A Chebyshev polynomial implementation of the polynomial approximation is given in function <cheb_1_dim.m>.

Now, there is nothing special about the polynomial family, and in fact, Fourier analysis uses the trigonometric family of basis functions. Of course, the problem is about coming up with a set of basis functions that approximate functions in some $\mathfrak{G}$ with computer. Therefore, Fourier series probably is not the proper family of orthonormal basis functions for a low-degree approximation of functions without continuity at the boundaries ($f(0) = f(1)$). There are tent function approximations, but their benefits are mostly from the adaptivity in higher-dimensions. Hence, they will be presented in a later section.

Implementation

Continuing with the previous notation to represent basis functions, we build a $(n \times m)$ matrix, Φ, where each column corresponds to a basis function evaluated at the $(n \times 1)$ vector, x. With polynomial basis functions, each column will correspond to different powers of x, i.e.,

$$\begin{aligned}\Phi &= \begin{bmatrix}\phi_1(x) & \phi_2(x) & \dots & \phi_m(x)\end{bmatrix} \\ &= \begin{bmatrix}1 & x & \dots & x^{m-1}\end{bmatrix}\end{aligned}$$

This can be coded in vectorized fashion as

```
x = linspace(0,1,100)';
Phi = bsxfun(@power,x,0:m-1);
```

[12] Therefore, "we use Chebyshev polynomials to approximate ..." is a fancy way of saying "we approximate using polynomials..." Numerically, there can be some differences, but the subspace they represent is the same. There recently has been active research in how to minimize numerical problems while using polynomials, but that is beyond the scope of this chapter and will be updated in the future.

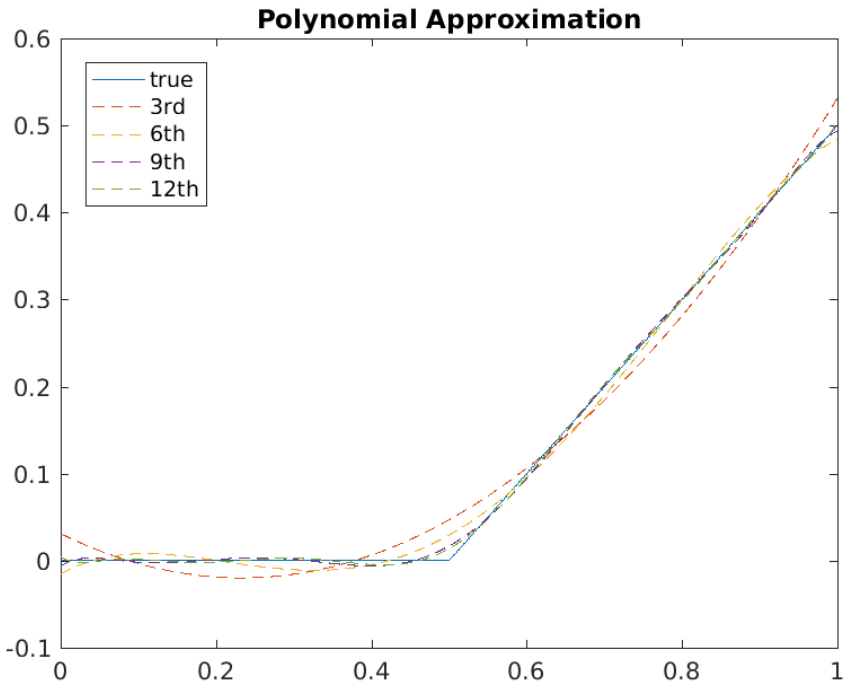


Figure 1.1: Approximation with different degree polynomials

1.3.2 Piecewise Construction without Pasting Conditions

Although any set of basis functions can be used, in practice, it is hard to create (global) basis functions beyond polynomial expansions. Also, depending on $\mathfrak{G}$, it might be better to work with a more local construction instead of using global basis functions like polynomials.[13] As a simple concrete example, consider $\mathfrak{G} = \{f_a\}_{a\in\mathbb{R}}$ with

$$f_a(x) = \begin{cases} 0 & \text{for } x \in \left[0, \frac{1}{2}\right) \\ a\left(x - \frac{1}{2}\right) & \text{for } x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

When polynomials are used, the functions in $\mathfrak{G}$ cannot be approximated too well as shown in figure 1.1. This is partly because functions in $\mathfrak{G}$ contain a kink that any global approximation by smooth functions will not handle well. However, if functions with localized supports are used, the

[13] Technically, the division between global and local is not clear as you can mix both ideas.

functions in $\mathfrak{G}$ can be approximated exactly with just one basis function

$$\phi_{1,1}(x) = \begin{cases} 0 & \text{for } x \in \left[0, \frac{1}{2}\right) \\ x - \frac{1}{2} & \text{for } x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

The idea of piecewise construction is to create partitions of the domain and to use basis functions with support within each partition.[14]

For notation, let $(x_1 = 0, x_1, \ldots, x_{n+1} = 1)$ be the knot points that define the partitions, where x_i's do not need to be uniformly spaced. For each partition, $[x_i, x_{i+1}]$, we can define a set of basis functions

$$\{\phi_{i,j}\}_{j=0,\ldots,m}$$

with support of $[x_i, x_{i+1})$. Therefore, for the given partition, the complete basis functions to approximate a given function are

$$\{\phi_{i,j}\}_{i=1,\ldots,n;j=0,\ldots,m}$$

For example, with polynomials, the basis functions are given by

$$\phi_{i,j}(x) = \begin{cases} \left(\frac{x - x_i}{x_{i+1} - x_i}\right)^j & x \in [x_i, x_{i+1}) \\ 0 & x \notin [x_i, x_{i+1}) \end{cases}$$

As an example, consider approximating

$$\log(x + \varepsilon) \qquad\qquad x \in [0, 1], 0 < \varepsilon \ll 1$$

[14]There is a different sense of "local," where you can use bump functions, and this is usually what is done in non-parametric methods in econometrics. For this chapter, "local" stands for partitioning.

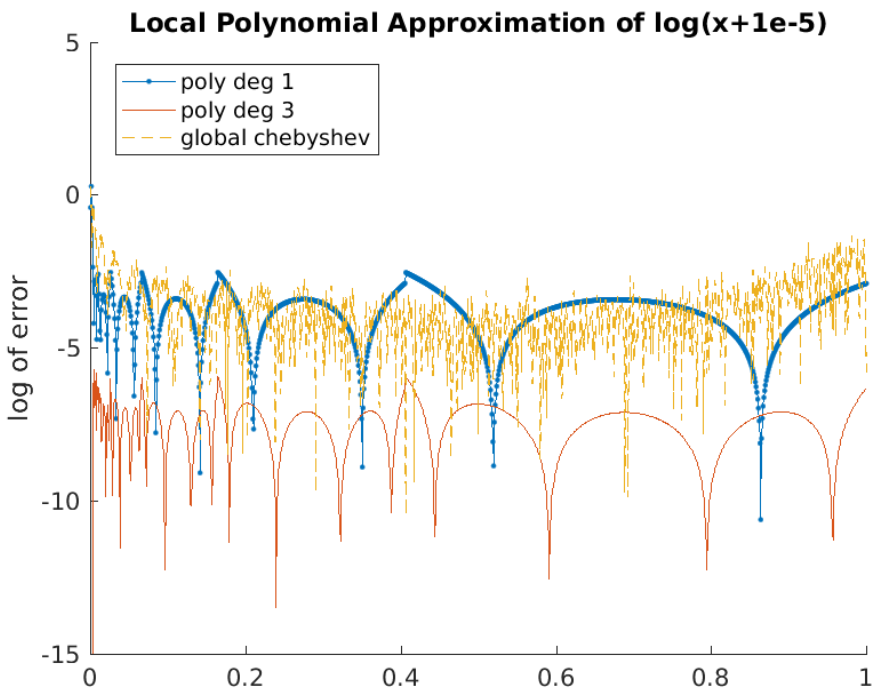


Figure 1.2: Local approximation

Local approximation by degree 3 and approximation by Chebyshev polynomials and quadratic spline with smooth pasting condition used 40 basis functions (20 for approximation by degree 1).

Because the function behaves poorly around 0, using the polynomial approximation behaves poorly. As shown in figure 1.2, using partitions to approximate functions separately can lead to vast improvements in approximation. This is a general trade-off when constructing basis functions. It is possible to increase the order of approximations by increasing the number of *global* basis functions. However, it is also possible to introduce more knots where the local function in the partition can be approximated with lower-order basis functions. There is no clean theoretical result in determining whether a higher order should be used instead of increasing the number of knot points. Heuristically, higher-order approximation would work better if the functions being approximated are smooth, and piecewise construction is better otherwise.

Implementation

One of the main benefits of using piecewise construction compared with another generalization method is the simplicity in programming. Continuing with the above definitions, let x be a $(n \times 1)$ vector of evaluation points, and let x_{knots} be the knot points that determine the partitions of the domain. First, given x, the partition that contains x needs to be found. This can be done in a simple way by just adding up the number of knots that are smaller than or equal to the given x; i.e.,

```
loc = sum(bsxfun(@minus,x',x_knots)>=0);
```

Since basis functions are not defined outside the knot points, the end points need to be adjusted to be the partitions defined by the knot points with the following codes

```
loc = max(1,loc);
loc = min(loc,length(x_knots)-1);
```

Given the partition, the value of x needs to be normalized to match the notation in this section.[15] This also can be done in a vectorized fashion by

```
t = (x - x(loc))./(x(loc+1) - x(loc));
```

Given this, the construction of polynomial basis functions requires only multiplication by t:

```
phi_0 = sparse(1:length(x),loc(x),ones(length(x),1),...
                          length(x),length(x_knots)-1);
phi_1 = bsxfun(@times, t, phi_0);
phi_2 = bsxfun(@times, t, phi_1);
phi_3 = bsxfun(@times, t, phi_2);
```

where each ϕ_i corresponds to i^{th}-order polynomial basis functions. This is it! It is literally only four lines of code to normalize x and n lines to get polynomial basis functions on partitions.

[15] Normalization is not strictly necessary, but it matches generalization of the global approximation of function over $[0, 1]$ to global constructions.

1.3.3 Piecewise Construction with Pasting

In the previous section, the basis functions between partitions were assumed to be completely independent. However, independence to this degree might not be necessary. For example, if we knew that the functions in $\mathfrak{G}$ are continuous, then it might be possible to impose constraints to further reduce the number of basis functions required. As a concrete example, consider the following basis functions

$$\phi_{1,0}(x) = \begin{cases} 1 & x \in \left[0, \frac{1}{2}\right) \\ 0 & x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

$$\phi_{1,1}(x) = \begin{cases} 2x & x \in \left[0, \frac{1}{2}\right) \\ 0 & x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

$$\phi_{2,0}(x) = \begin{cases} 0 & x \in \left[0, \frac{1}{2}\right) \\ 1 & x \in \left[\frac{1}{2}, 1\right] \end{cases}$$

Without constraints, the linear combinations between the functions can result in discontinuity at $\frac{1}{2}$. Worse, it might have been possible to infer the values for $x \in \left[\frac{1}{2}, 1\right]$ using only values from the left half, so we might carry excess information without the need. Instead, suppose that we were to impose continuity at $\frac{1}{2}$; i.e.,

$$\begin{aligned} a_{1,0} + a_{1,1} &= \lim_{x \to \frac{1}{2}-} a_{1,0}\phi_{1,0}(x) + a_{1,1}\phi_{1,1}(x) + a_{2,0}\phi_{2,0}(x) \\ &= \lim_{x \to \frac{1}{2}+} a_{1,0}\phi_{1,0}(x) + a_{1,1}\phi_{1,1}(x) + a_{2,0}\phi_{2,0}(x) \\ &= a_{2,0} \end{aligned}$$

Thus,

$$a_{1,0} + a_{1,1} - a_{2,0} = 0.$$

That means that given $\begin{bmatrix} \tilde{a}_{1,0} \\ \tilde{a}_{1,1} \end{bmatrix}$, the approximation for the function is defined by

$$a = \begin{bmatrix} \tilde{a}_{1,0} \\ \tilde{a}_{1,1} \\ \tilde{a}_{1,0} + \tilde{a}_{1,1} \end{bmatrix}.$$

Therefore, when continuity is assumed, $\tilde{a} \in \mathbb{R}^2$ is sufficient to represent the approximation with the given basis functions. Now, this can be applied generally given a restriction in linear equations; i.e.,

$$Ra = 0.$$

Hence, if the approximating functions are assumed to satisfy the pasting conditions, then a has to lie in the null space of R. Therefore, finding basis functions with pasting conditions is a two-step process. The first step is to find basis functions that span the relevant (functional) subspace of interest, and the second step is to reduce the coefficients of the basis functions to satisfy the pasting conditions.

Although any general constraint R can be used, most of the time, the restriction R used is of continuity constraints. To formalize the notation, let $a_{i,j}$ be the basis vector of the coefficients and the approximating function

$$\sum_{i,j} a_{i,j}\phi_{i,j}(x).$$

Hence, the continuity constraint at knot k is

$$\begin{aligned}\sum_{i,j} a_{i,j} \left(\lim_{x \to x_{k+1}-} \phi_{i,j}(x) \right) &= \lim_{x \to x_{k+1}-} \sum_{i,j} a_{i,j} \phi_{i,j}(x) \\ &= \lim_{x \to x_{k+1}+} \sum_{i,j} a_{i,j} \phi_{i,j}(x) \\ &= \sum_{i,j} a_{i,j} \left(\lim_{x \to x_{k+1}+} \phi_{i,j}(x) \right)\end{aligned}$$

which leads to a constraint of the form

$$
\begin{aligned}
&\sum_{i,j} a_{i,j}\phi_{i,j}(x_{k+1}+) - \phi_{i,j}(x_{k+1}-) \\
&\qquad = \begin{bmatrix} \phi_{1,0}(x_{k+1}+) - \phi_{1,0}(x_{k+1}-) \\ \phi_{1,1}(x_{k+1}+) - \phi_{1,1}(x_{k+1}-) \\ \vdots \\ \phi_{1,m}(x_{k+1}+) - \phi_{1,m}(x_{k+1}-) \\ \phi_{2,1}(x_{k+1}+) - \phi_{2,1}(x_{k+1}-) \\ \vdots \\ \phi_{n,m}(x_{k+1}+) - \phi_{n,m}(x_{k+1}-) \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a_{1,0} \\ a_{1,1} \\ \vdots \\ a_{1,m} \\ a_{2,1} \\ \vdots \\ a_{n,m} \end{bmatrix} \\
&\qquad = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \phi_{k,1}(x_{k+1}+) - \phi_{k,1}(x_{k+1}-) \\ \vdots \\ \phi_{k,m}(x_{k+1}+) - \phi_{k,m}(x_{k+1}-) \\ \phi_{k+1,1}(x_{k+1}+) - \phi_{k+1,1}(x_{k+1}-) \\ \vdots \\ \phi_{k+1,m}(x_{k+1}+) - \phi_{k+1,m}(x_{k+1}-) \\ 0 \\ \vdots \\ 0 \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} a_{1,0} \\ a_{1,1} \\ \vdots \\ a_{1,m} \\ a_{2,1} \\ \vdots \\ a_{n,m} \end{bmatrix} \\
&\qquad \equiv R_k a = 0.
\end{aligned}
$$

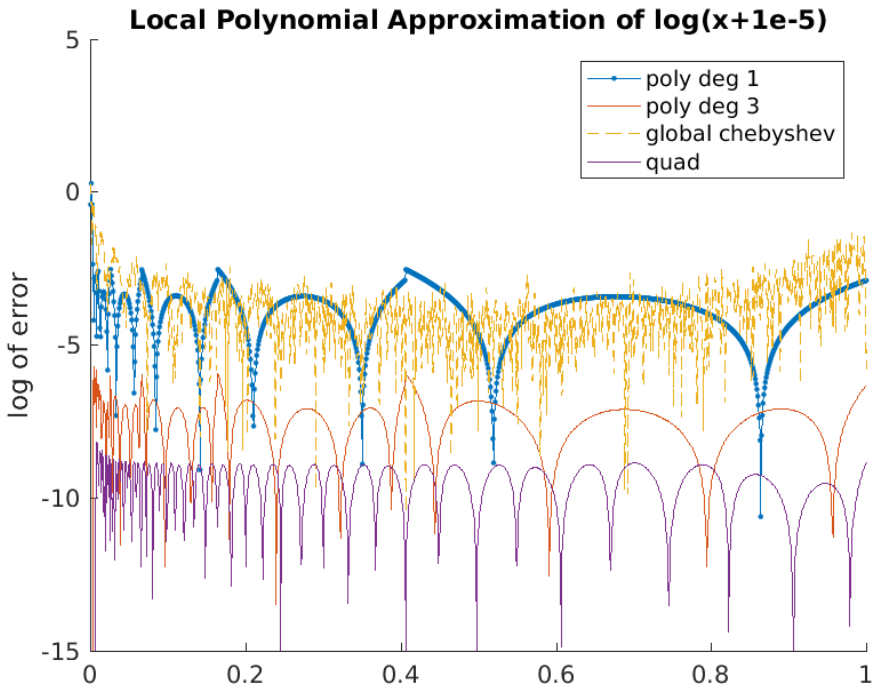


Figure 1.3: Local approximation with Pasting Conditions

Local approximation by degree 3 and the approximation by Chebyshev polynomials used 40 basis functions (20 for approximation by degree 1).

Therefore, imposing continuity constraints at all points becomes

$$\begin{bmatrix} R_2 \\ R_3 \\ \vdots \\ R_n \end{bmatrix} a \equiv Ra = 0.$$

The higher-order continuously differentiability requirements at the boundaries take the same form except for $\phi_{i,j}^{(k)}(x_{i+1}+)$'s and $\phi_{i,j}^{(k)}(x_{i+1}-)$'s taking the places of $\phi_{i,j}(x_{i+1}+)$'s and $\phi_{i,j}(x_{i+1}-)$'s.

As an example of the usage, consider the same problem of approximating $\log(x+\varepsilon)$ again. The approximation given by the quadratic spline with continuously differentiability is given in

figure 1.3. Although the quadratic spline does not use the highest order, it gives the best approximation for the same dimension of the subspace. As mentioned before, the trade-off between increasing the order within the partition versus increasing the number of partitions does not have a clean answer and will depend on the specific problem.[16]

Example and Implementation

For the normalized polynomials built up above, the constraint can be applied in a vectorized fashion in one line of MATLAB code per continuity constraint. As a concrete example, suppose we are building a cubic spline with continuity and continuously differentiability constraints. For a continuity constraint at knot k, we have

$$a_{k,0} + a_{k,1} + a_{k,2} + a_{k,3} = a_{k+1,0},$$

which can be implemented simply as:

```
n_part = length(x_knots) - 1;
R_0 = spdiags(ones(n_part,4),[0,n_part,2*n_part,3*part],n_part-1,4*n_part)
          - spdiags(ones(n_part,1),1,n_part-1,4*n_part);
```

1.3.4 Relationship with Interpolation

Although the notation has been very different, interpolation can be considered to be an approximation by linear subspaces of basis functions. First, a standard interpolation problem is

Interpolation Problem. *Given $f(x_1), f(x_2), \ldots, f(x_n)$ of function values at some set of points, approximate the function value $\widetilde{f}(y)$ for any given y.*

To follow the standard notation from before, let x be the $(n \times 1)$ vector of points where the function values are given, and let y be the $(l \times 1)$ vector of points to interpolate the values. Given

[16]There is one argument against going to higher order if the shape of the approximating function matters. It is well-known that higher-order polynomial approximations lead to loss of monotonicity/concavity.

m basis functions $\{\phi_i\}_{i=1,\ldots,m}$, using our standard representation of basis functions, we get that

$$\widetilde{f}(z) = \sum_{i=1}^{m} a_i \phi_i(z) \tag{1.1}$$

where a_i is found using the function values from x minimizing the loss function. Given the function, computing the value at any given point y involves plugging in y in place of z in equation (1.1). In addition, this is simple to implement on a computer. Let Φ_x and Φ_y be the (orthonormalized) matrix representation of the basis functions with the evaluation points of x and y, respectively. Interpolation requires the same calculation

```
tilde_f_y = Phi_y*((Phi_x'*Phi_x)\(Phi_x'*f_x));
```

The most commonly used interpolations can be understood geometrically by considering the subspaces spanned by the basis functions instead of taking them to be a black box. Treating interpolations as black boxes can lead to unintended consequences. For example, consider the cubic spline interpolation that most numerical programming languages have available as a function. There is a subtlety to which people do not give any thought. In the language we built up, the "cubic interpolation" is a piecewise cubic approximation with a twice continuously differentiability pasting condition where the knot points of the pasting conditions are exactly the points where the function is given; i.e., $x = x_{knots}$. However, consider the number of basis functions under this construction. There are $4(\{1, x, x^2, x^3\})$ basis functions per partition, and three continuity conditions per internal knot points. Thus, the number of basis functions is $4(n-1)-3(n-2) = n+2$. Hence, $m > n$, and the given function points do not pin down the coefficients of the subspace completely.[17] This is because there are no pasting conditions at the end knot points. In other words, the standard intuition for splines does not give enough restrictions to define a unique interpolation. In some implementations of cubic interpolation, the natural boundary conditions

$$\widetilde{f}''(x_1) = 0 \qquad\qquad \widetilde{f}''(x_n) = 0$$

[17] In econometrics terms, the full rank condition is violated.

are used. Alternately, other forms of boundary conditions can be used that require a guess about what a reasonable boundary condition would be. For example,

$$\widetilde{f}'(x_1) = a_1 \qquad\qquad \widetilde{f}''(x_n) = a_n$$

where a_1 and a_n are guesses of the derivative values. For example, the divided difference can be used to guess the derivative values. In economics, however, any method that requires setting external boundary conditions is likely to give bad approximations. This is because borrowing constraints will be imposed at the boundaries, and that is precisely where the value function will have the highest curvature. Hence, these are precisely the locations where a guess of the boundary conditions would be really poor.

When basis functions are generated not for interpolation, this problem is side-stepped by not imposing the boundary conditions, keeping the dimensionality of the basis functions at $n + 2$, and requiring that the number of evaluation points is bigger than $n + 2$.[18] Even if the basis functions were used for interpolation, there are better ways to handle the problem than the standard boundary conditions in the literature. The first point is that the definition of partition does not have to line up with the evaluation points even if the piecewise cubic function is used. Second, if more conditions were to be imposed on coefficients, then the end point with a large curvature is not the point to which to impose the constraint. Hence, it would be more natural to impose constraints on points with a smaller curvature. The locations would be problem specific, which is why a black box interpolation can fail. When a prebuilt interpolation scheme does not work, one should consider what the subspace of reduction is before declaring defeat. This is also because the implementation of the custom subspace is very fast in evaluation and in coding time.[19]

[18] In our implementation, we are actually using orthonormalization to reduce the number of basis functions.

[19] This is program dependent, but a cubic interpoltion in MATLAB by default imposes not-a-knot constraint, which means that you have to carry two extra function values.

1.4 Higher Dimensions

Thus far, we have created a systematic way of approximating functions in one dimension. In this section, the construction is generalized to higher dimensions. Higher dimensions require a new set of notations. Although a general set of evaluation points can be taken, in this section a structured set of evaluation points is used. For a function defined over $\mathbb{R}^d$, x_i's are $(n_i \times 1)$ vectors representing the values at which the function is defined in the i^{th} coordinate. Therefore, the function is to be evaluated on $\prod_i n_i$ points defined by $(x_{1;i_1}, x_{2;i_2}, \ldots, x_{d;i_d})$, where $x_{i;j_i}$ denotes the j_i points of evaluation in the i^{th} dimension. For example, in $\mathbb{R}^2$, if

$$x_1 = \begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix} \qquad\qquad x_2 = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix}$$

then the functions are evaluated on

$$\begin{matrix} (0,3) & (0,4) & (0,5) \\ (1,3) & (1,4) & (1,5) \\ (2,3) & (2,4) & (2,5) \end{matrix}$$

In addition, the indexing notation gives $x_{1;2} = 1$ and $x_{2;1} = 4$. Hence, this product form of the grid results in rectangular grids. These are often the grids that are used in solving partial differential equations (PDEs) with finite difference/volume methods.

Implementation

Although x can be stored in any order, one representation is the most efficient. In the following, x denotes points of the function evaluation given in the following order

$$x = \begin{bmatrix} (x_{1;1}, x_{2;1}, \ldots, x_{d;1}) \\ (x_{1;2}, x_{2;1}, \ldots, x_{d;1}) \\ \vdots \\ (x_{1;m_1}, x_{2,1}, \ldots, x_{d;1}) \\ (x_{1;1}, x_{2,2}, \ldots, x_{d;1}) \\ \vdots \\ (x_{1;m_1-1}, x_{2,m_2}, \ldots, x_{d;m_d}) \\ (x_{1;m_1}, x_{2,m_2}, \ldots, x_{d;m_d}) \end{bmatrix}$$

This is a natural order as this is how a d-dimensional array is stored in MATLAB in memory. Reshaping back and forth between the vector form and the tensor form does not require any reshuffling of data and, thus, the above memory representation is more efficient than other possible orderings.[20] For the example given in the previous section, x would be

$$x = \begin{bmatrix} (0,3) & (1,3) & (2,3) & (0,4) & (1,4) & (2,4) & (0,5) & (1,5) & (2,5) \end{bmatrix}^T$$

1.4.1 Tensor Construction

Given one-dimensional functions, it is possible to create a function defined over d dimensions using tensor products.[21] For example, given $\{\phi_i(x)\}_{i=1,\ldots,d}$, it is possible to define a d-dimensional function as

$$\phi(x_1, x_2, \ldots, x_d) = \phi_1(x_1) \cdot \phi_2(x_2) \cdot \ldots \cdot \phi_d(x_d) \qquad x_1, x_2, \ldots, x_d \in [0,1]$$

[20] In other systems, with different memory representation of matrices like Python, x should be stacked differently.
[21] No knowledge of tensor products is needed to read this section.

Therefore, given any family of basis functions in each dimension, it is possible to define higher-dimensional basis functions using the product structure. Since notations can become messy with more than two dimensions, the following is denoted with the notation for $\mathbb{R}^2$, but the extension to an arbitrary d is straightforward. Let $\{\phi_i(x)\}_{i=1,2,\ldots,m_1}$ and $\{\varphi_i(y)\}_{i=1,2,\ldots,m_2}$ be the basis functions for the first and second coordinates, respectively. Then, the tensor product form of the basis function is

$$\{\phi_{(i,j)}(x,y) := \phi_i(x) \cdot \varphi_j(y)\}_{i=1,2,\ldots,m_1;j=1,2,\ldots,m_2}$$

For example, with polynomial basis functions,

$$\begin{aligned} \phi_1(x) &= 1 & \varphi_1(y) &= 1 \\ \phi_2(x) &= x & \varphi_2(y) &= y \\ \phi_3(x) &= x^2 & \varphi_3(y) &= y^2 \end{aligned}$$

The basis functions for two dimensions is simply given by

$$\begin{aligned} \phi_1(x) \cdot \varphi_1(y) &= 1 & \phi_1(x) \cdot \varphi_2(y) &= y & \phi_1(x) \cdot \varphi_3(y) &= y^2 \\ \phi_2(x) \cdot \varphi_1(y) &= x & \phi_2(x) \cdot \varphi_2(y) &= xy & \phi_2(x) \cdot \varphi_3(y) &= xy^2 \\ \phi_3(x) \cdot \varphi_1(y) &= x^2 & \phi_3(x) \cdot \varphi_2(y) &= x^2y & \phi_3(x) \cdot \varphi_3(y) &= x^2y^2 \end{aligned}$$

Implementation

This form of basis function generation is straightforward using the notations built up from before. Let $A(= \Phi_1)$ and $B(= \Phi_2)$ be the basis functions defined over their respective evaluation points in

coordinates 1 and 2. Then, following the notation, the basis function for the product will be

$$\begin{pmatrix}
A_{11}\cdot B_{11} & A_{12}\cdot B_{11} & \dots & A_{1m_1}B_{11} & A_{11}B_{12} & \dots & A_{1m_1}\cdot B_{1m_2} \\
A_{21}\cdot B_{11} & A_{22}\cdot B_{11} & \dots & A_{2m_1}B_{11} & A_{21}B_{12} & \dots & A_{2m_1}\cdot B_{1m_2} \\
 & & & \vdots & & & \\
A_{n_11}\cdot B_{11} & A_{n_12}\cdot B_{11} & \dots & A_{n_1m_1}B_{11} & A_{n_11}B_{12} & \dots & A_{n_1m_1}\cdot B_{1m_2} \\
A_{11}\cdot B_{21} & A_{12}\cdot B_{21} & \dots & A_{1m_1}B_{21} & A_{11}B_{22} & \dots & A_{1m_1}\cdot B_{2m_2} \\
 & & & \vdots & & & \\
A_{n_11}\cdot B_{n_21} & A_{n_12}\cdot B_{n_21} & \dots & A_{n_1m_1}B_{n_11} & A_{n_11}B_{n_22} & \dots & A_{n_1m_1}\cdot B_{n_2m_2}
\end{pmatrix}$$

In inspection, this is just

$$\begin{pmatrix}
A\cdot B_{11} & A\cdot B_{12} & \dots & A\cdot B_{1m_2} \\
A\cdot B_{21} & A\cdot B_{22} & \dots & A\cdot B_{2m_2} \\
\vdots & & & \\
A\cdot B_{n_21} & A\cdot B_{n_22} & \dots & A\cdot B_{n_2m_2}
\end{pmatrix}$$

However, this is just a tensor product of B and A, $B\otimes A$. Most high-level programming languages have tensor products implemented. Therefore, this can be written as

```
Phi = kron(Phi_2,Phi_1);  %Phi_1 = A, Phi_2 = B
```

Note that this is the case due to the specific ordering of x that aligns well with tensor products. In addition, this simplicity generalizes to higher dimensions by only a recursive application. For example, for three dimensions, it can be simply written as

```
Phi = kron(Phi_3,kron(Phi_2,Phi_1));
```

and it can be checked that this computes things in the correct order by checking the indices of the tensor product. Hence, the extension of the basis function to higher dimensions is only one line of

code per dimension.[22] Last, it is possible that no "basis functions" are used in some dimensions. This is equivalent to using canonical basis functions as explained in section 1.2.1. Thus, in this case, $\Phi_i = I_{n_i}$, and the same tensor product works as given.

1.4.2 Piecewise Construction in Higher Dimensions

Although tensor product construction is simple, as seen in the one-dimensional setting, it might be better to build basis functions that are more localized. The piecewise construction can be generalized to higher dimensions directly with a simple generalization of the implementation. Unfortunately, the pasting condition does not generalize to higher dimensions unless a specific family of functions is used. This is because the boundary between partitions has dimension $(d-1)$. Therefore, this is an uncountable object that cannot be matched by putting finite pasting conditions. Fortunately, matching the boundary conditions is still feasible for the most frequently used cases of polynomial splines. The construction is basis function dependent. The construction for a two-dimensional quadratic spline with the continuity condition will be given here, but similar constructions can be generalized as long as a reasonable set of polynomial functions is used. Note that the same points about interpolation given in section 1.3.4 apply here as well. No boundary conditions were used, and hence, the dimensionality of the space is higher by not imposing pasting conditions at the boundaries. Bicubic splines with implicit boundary conditions are more standard,[23] but again, care needs to be taken to make sure what form of boundary conditions is being imposed before using the computer implementation of interpolation/splines.

Implementation of Two-Dimensional Quadratic Spline

Let x_{knot} and y_{knot} be the knot points that determine the partitions in two dimensions and n_x and n_y be the number of knot points in each dimension. Given a partition, the basis functions to be

[22] Just note to order basis functions in reverse order in tensor products.

[23] from the usage in image processing

used are

$$1 \qquad x \qquad y \qquad x^2 \qquad xy \qquad y^2 \qquad x^2y \qquad xy^2$$

Therefore, there are eight basis functions for each partition. These basis functions are represented by stacking the partition location. For example, if (x_1, y_1) is in partition $(1, 1)$ and (x_2, y_1) is in $(2, 1)$, then the basis function at those evaluation points is

$$\begin{array}{c} \\ (x_1,y_1) \\ (x_2,y_1) \\ \\ \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & x & \dots & xy^2 & 1 & x & \dots & xy^2 & \dots \end{array} \\ \begin{pmatrix} 1 & x_1 & \dots & x_1y_1^2 & 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & \dots & 0 & 1 & x_2 & \dots & x_2y_1^2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots \end{pmatrix} \end{array}$$

They can be coded using the same technique as above. First, we compute at which partition the evaluation points are located:

```
n_x = length(x_knots);
n_y = length(y_knots);


loc_x = sum(bsxfun(@minus,x',x_knots)>=0);
loc_y = sum(bsxfun(@minus,y',y_knots)>=0);


loc_x = max(loc_x,1);
loc_x = min(loc_x,n_x-1);
loc_y = max(loc_y,1);
loc_y = min(loc_y,n_y-1);
```

Given the partition, we compute the normalized value of x and y in the partition

```
t_x = (x - x(loc_x))./(x(loc_x+1) - x(loc_x));
```

```
ones_x = ones(length(x),1);
t_y = (y - y(loc_y))./(y(loc_y+1) - y(loc_y));
ones_y = ones(length(y),1);
```

Given the values, we build up the eight basis functions for the partition

```
phi_1 = bsxfun(@times, ones_x,ones_y');
phi_x = bsxfun(@times, t_x,ones_y');
phi_y = bsxfun(@times, ones_x,t_y');
phi_x2 = bsxfun(@times, t_x.^2,ones_y');
phi_xy = bsxfun(@times, t_x,t_y');
phi_y2 = bsxfun(@times, ones_x,t_y');
phi_x2y = bsxfun(@times, t_x.^2,t_y');
phi_xy2 = bsxfun(@times, t_x,t_y'.^2);

phi = [phi_1(:);phi_x(:);phi_y(:);phi_x2(:);...
       phi_xy(:);phi_y2(:);phi_x2y(:);phi_xy2(:)];
```

Although it is more complicated than in one dimension, we can compute which partition the values are in by:[24]

```
loc_1 = bsxfun(@plus,loc_x(:)*8-7,(loc_y(:)'-1)*8*n_x);
loc_x = bsxfun(@plus,loc_x(:)*8-6,(loc_y(:)'-1)*8*n_x);
loc_y = bsxfun(@plus,loc_x(:)*8-5,(loc_y(:)'-1)*8*n_x);
loc_x2 = bsxfun(@plus,loc_x(:)*8-4,(loc_y(:)'-1)*8*n_x);
loc_xy = bsxfun(@plus,loc_x(:)*8-3,(loc_y(:)'-1)*8*n_x);
loc_y2 = bsxfun(@plus,loc_x(:)*8-2,(loc_y(:)'-1)*8*n_x);
loc_x2y = bsxfun(@plus,loc_x(:)*8-1,(loc_y(:)'-1)*8*n_x);
```

[24] Finding the location can be done in one line, but it is hard to decipher. Refer to the attached codes to see how the following can be done in one line.

```
loc_xy2 = bsxfun(@plus,loc_x(:)*8,(loc_y(:)'-1)*8*n_x);


loc = [loc_1(:);loc_x(:);loc_y(:);loc_x2(:);...
       loc_xy(:);loc_y2(:);loc_x2y(:);loc_xy2(:)];
```

Finally, we build the basis functions corresponding to the proper partitions:

```
Phi = sparse(repmat((1:length(x)*length(y))',8,1),loc,phi);
```

This results in basis functions over the partitions that do not impose a continuity requirement. For example, when these basis functions are used to approximate

$$f(x,y) = x^2 + y^2 \qquad\qquad x, y \in [0,1]$$

with equi-spaced knot points, the functions do not retain continuity over the boundary as can be seen in figure (1.4).

Continuity can be imposed at the boundary conditions. This is possible because of the way the basis functions were chosen. Given the eight basis functions, the continuity at the boundary can be guaranteed by matching the level values and one of the directional derivatives as shown in figure 1.6.

The form of the boundary condition is similar to that of one dimension. For example, matching the value at (x_{i+1}, y_i) between partition (i,j) and partition (i+1,j) is

$$\sum_k a_{(i,j);k}\phi_{(i,j);k}(x_{i+1}, y_j) = \sum_k a_{(i+1,j);k}\phi_{(i+1,j);k}(x_{i+1}, y_j)$$

where $k \in \{1, x, y, x^2, xy, y^2, x^2y, xy^2\}$ and index $(i,j);k$ denotes basis function/coefficient k in partition (i,j). Using the definition of the basis functions, we get the constraint to be

$$a_{(i,j);1} + a_{(i,j);x} + a_{(i,j);x^2} = a_{(i+1,j);1}$$

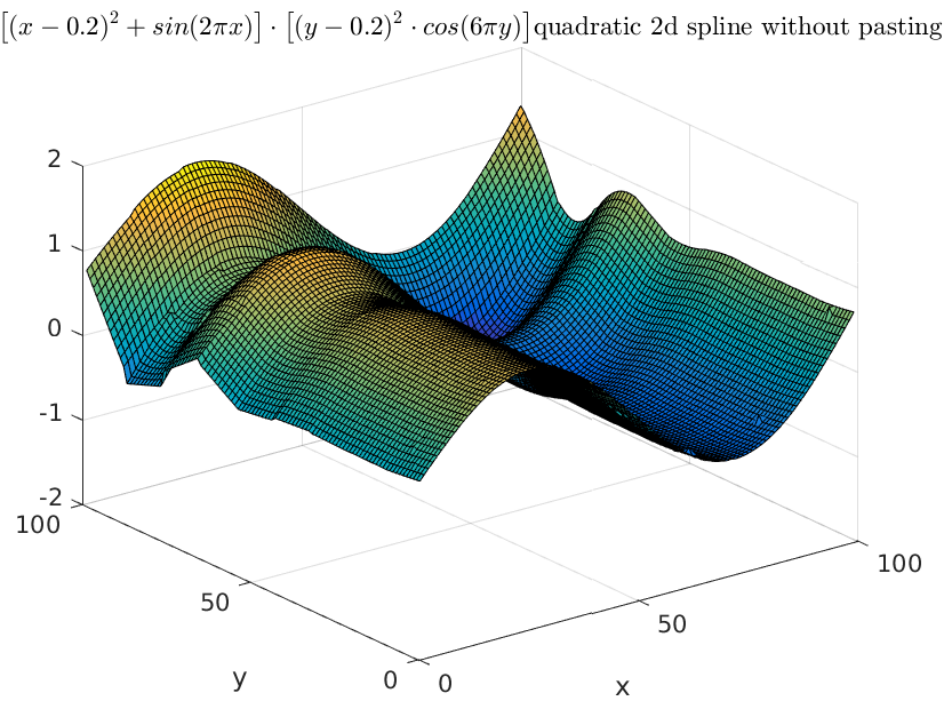


Figure 1.4: 2D quadratic spline without the pasting conditions

Only five knot points in each direction are used.

Similarly, the continuity at (x_{i+1}, y_{i+1}) gives

$$a_{(i,j);1} + a_{(i,j);x} + a_{(i,j);y} + a_{(i,j);x^2} + a_{(i,j);xy} + a_{(i,j);y^2} + a_{(i,j);x^2y} + a_{(i,j);xy^2} = a_{(i+1,j);1} + a_{(i+1,j);y} + a_{(i+1,j);y^2}$$

Collecting all the pasting conditions results in a constraint of the form $Ra = 0$. R can be built as a sparse matrix with specific diagonals filled in completely analogous to the pasting condition in one dimension, and thus are left as an exercise. For this problem, the null space can be computed directly, saving an SVD. This has been implemented as `<twoDquad_spline.m>`. As can be seen in figure 1.5, imposing pasting conditions recovers the continuity.

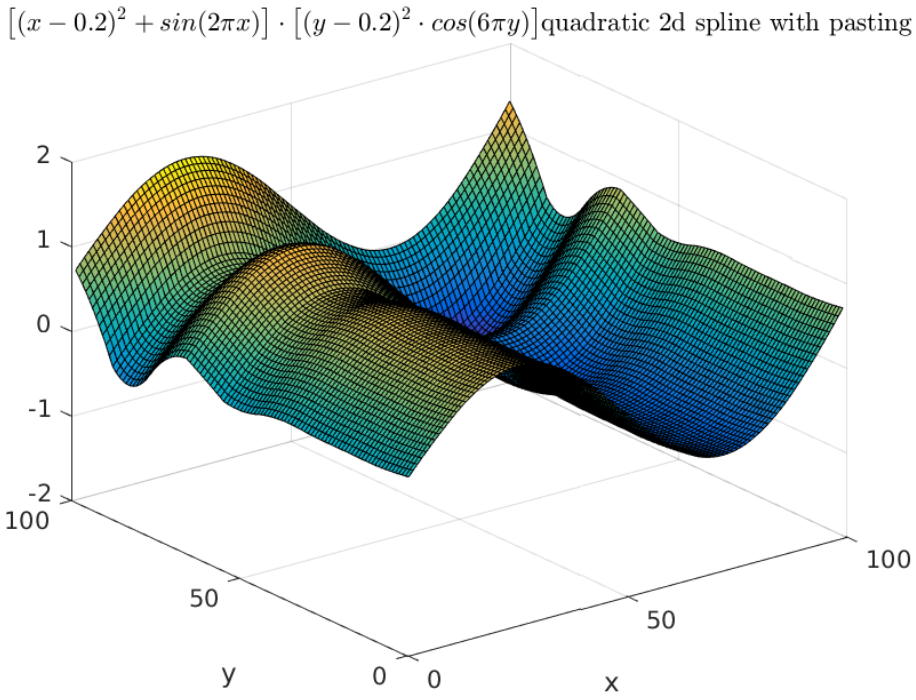


Figure 1.5: 2D quadratic spline with the pasting conditions

Only five knot points in each direction are used.

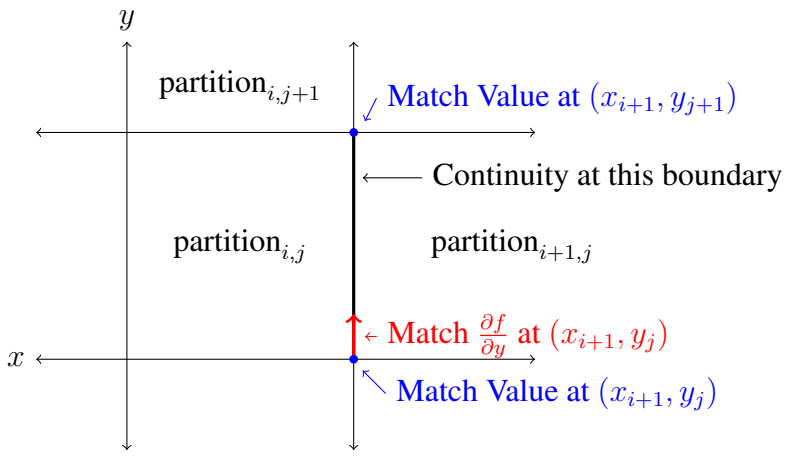


Figure 1.6: The structure of continuity constraints

Continuity across partitions can be guaranteed if a good set of basis functions is chosen as shown in the text.

1.4.3 Non-Tensor Product Construction

Although the tensor product construction for generalizing basis functions from one dimension to higher dimensions is straightforward, that might not be the most efficient thing to do. For example, if tensor product basis functions were used, then with quadratic polynomials in each direction, the basis function for two dimensions results in

$$\begin{array}{c} \\ 1 \\ y \\ y^2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & x & x^2 \end{array} \\ \begin{pmatrix} 1 & x & x^2 \\ y & xy & x^2y \\ y^2 & xy^2 & x^2y^2 \end{pmatrix} \end{array}$$

Of the fourth-order terms, only the x^2y^2 term is included. In terms of taking polynomial approximations, this is an inefficient ordering of basis functions. This can be generally if the basis functions in one dimension have some sort of ordering defined on them. For polynomials, the degree of the (monic) polynomial would be the ordering of the approximating function. Polynomials are not the only family of functions that have intrinsic ordering. For example, one family of functions that works very well for higher-dimensional construction is the family of tent functions.[25] They are defined by

$$\phi_{l,p}(x) = max(1 - \left|2^l \cdot x - p\right|, 0) \qquad\qquad x \in [0,1]$$

where l is a non-negative integer, p is an odd integer between 0 and 2^l when $l > 0$, and $p \in \{0,1\}$ when $l = 0$. These functions can be more intuitively understood when they are plotted. As seen in figure 1.7, these are just "tent" functions with progressively smaller support. Hence, this family of functions also has an implicit ordering defined by the level term l. In the following, the notation $o(\phi_i) \in \mathbb{R}_+$ is used to compute the order/degree of the basis function. For example, $o(x^k) := k$ for

[25] These were not introduced as a part of global functions. This family of basis functions can be used in one dimension as well. They would work even better with adaptivity, but these functions have more benefits in higher dimensions.

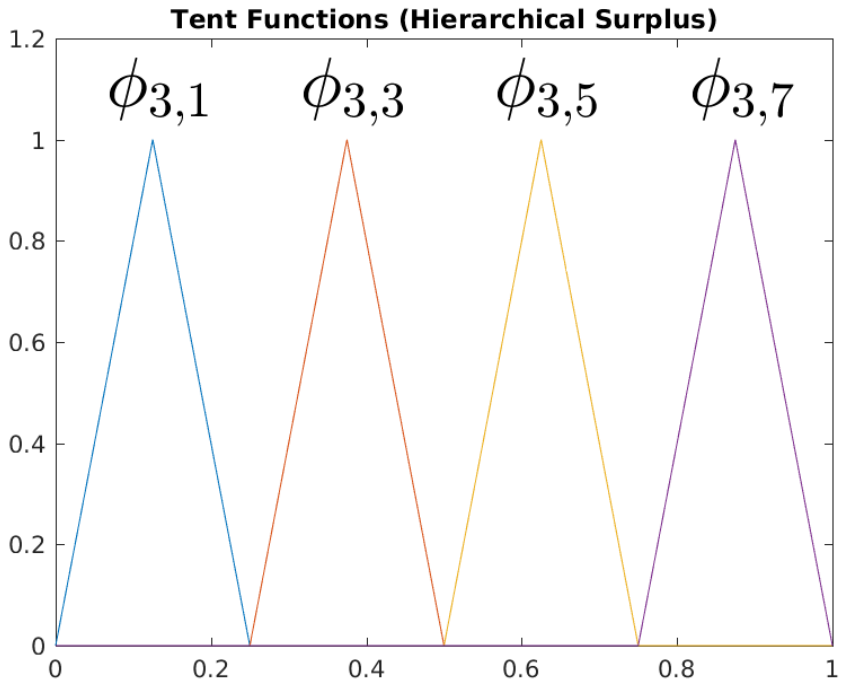


Figure 1.7: Tent (hierarchical surplus/locally linear) functions

polynomials, and $o(\phi_{l,p}) = l$ for tent functions. For the last notation, let superscripts denote the coordinate in which the basis function is defined. For example, $\phi^3_{l,p}$ is a (l, p) tent function defined for the third coordinate. As before, the basis function can be generated using the product structure by

$$\phi_{(i_1,\ldots,i_d)}(x_1, \ldots, x_d) = \prod_j \phi^j_{i_j}(x_j)$$

With this notation, a different method of basis function generation can be defined simply.

Tensor Product. *A tensor product set of functions of order* $(k_1, \ldots, k_d)$ *is defined by*

$$\left\{\phi_{(i_1,\ldots,i_d)}\right\} \qquad\qquad \forall j, \quad o(\phi_{i_j}) \leq k_j$$

Total Degree. *The set of basis functions with the total degree less than k is defined by*

$$\left\{\phi_{(i_1,\ldots,i_d)}\right\} \qquad\qquad \sum_j o(\phi_{i_j}) \leq k$$

There are infinitely many ways of defining the set of basis functions with a further generalization:

Generalized Total Degree. *The set of basis functions with the generalized total degree less than k with a given transformation* $g : \mathbb{R}_+^d \to \mathbb{R}_+$ *is defined by*

$$\left\{\phi_{(i_1,\ldots,i_d)}\right\} \qquad\qquad g(o(\phi_{i_1}),\ldots,o(\phi_{i_d})) \leq k$$

Note that g does not have to be symmetric. It is called an "anisotropic" construction when g is asymmetric.

For example, for polynomial basis functions, Bäck et al. (2011) noted that what they call hyperbolic cross space $g(x_1,\ldots,x_d) = \prod_{i=1}^d (x_i+1)$ and Smolyak polynomial space $g(x_1,\ldots,x_d) = \sum f(x_i)$ where

$$f(x) = \begin{cases} 0, & p = 0 \\ 1, & p = 1 \\ \lceil log_2(x) \rceil, & p \geq 2 \end{cases}$$

work well. For tent functions, total degree construction is known as sparse grids, and they are known to behave very well as noted in the literature survey of sparse grid by Bungartz and Griebel (2004).[26]

Last, the decision to include basis functions did not depend on the coefficients of the functions, $a_{(i_1,\ldots,i_d)}$, that are being approximated in the presentation given above. However, it is possible to

[26]The Smolyak method (sparse grids) is presented most of the time as some black box, but the intuition for the construction is simple as it is carried over from the polynomial example.

create a selection criterion depending on how important a certain basis function is in approximating the given function. This can be encoded with another cutoff function h that can depend on $a_{(i_1,\ldots,i_d)}$. Since the notation will be problem specific with not much intuition gained, it is not formalized here. For the dimension adaptivity, refer to Gerstner and Griebel (2003), and for local adaptivity, refer to Griebel (1998) and Ma and Zabaras (2009). Chapter 2 contains applications of dimension adaptive methods for solving Hamilton-Jacobi-Bellman equations in continuous time.

1.5 Conclusion

This chapter is written with the wish that people consider geometrically about the approximations they make instead of using prebuilt approximation routines and declaring premature defeat. Most of the approximations that people make are approximations on linear subspaces. By thinking about the approximation process this way, the thought process for choosing which set of basis functions is clearer. Under this unified framework, there is no problem in working with a hybrid case where one can mix Chebyshev polynomials and local splines when a specific problem requires them as the creation of those basis functions is very simple. At the very least, the codes provided here can shorten programming time. As the codes themselves are so short, it might take less time to implement than to learn the syntax of a prebuilt function.

1.6 Codes Appendix

The codes that were separated into multiple lines are presented here for completeness. The following are the piecewise cubic basis function, polynomial basis function, and the piecewise quadratic basis function with the continuously differentiability condition. A fair amount of codes are for plotting. The following codes are also available at `http://www.sehyoun.com` :

```
n_deg = 3;
n_red = 8;
n_cheb = 4*n_red;
```

```
%% Create x values to evaluate the function
x_fine = linspace(0,1,1000)';
y = log(x_fine+1e-5);

%% Create Partitions
lambda = 0.1;     % Determine non-uniformity
                  % Higher lambda leads to more non-uniform distribution
x_knots = exp(lambda*(1:n_red)');
x_knots = (x_knots-x_knots(1))/(x_knots(end)-x_knots(1));


%% Create local polynomial basis functions
% Find the partition x lies in
loc = sum(bsxfun(@minus,x_knots',x_fine)<=0,2);
loc = min(loc,length(x_knots)-1);
loc = max(loc,1);

% Compute normalized value
t = (x_fine - x_knots(loc))./(x_knots(loc+1)-x_knots(loc));

% Generate monic polynomials with local support
phi_i{1} = sparse(1:length(x_fine),loc,ones(length(x_fine),1));
for i = 2:n_deg+1
    phi_i{i} = bsxfun(@times,t,phi_i{i-1});
end

%% Piecewise Constant
X0 = cell2mat(phi_i(1:1));
[u0,~,~] = svd(full(X0),'econ');

%% Piecewise Linear
X1 = cell2mat(phi_i(1:2));
[u1,~,~] = svd(full(X1),'econ');

%% Piecewise Quadratic
X2 = cell2mat(phi_i(1:3));
[u2,~,~] = svd(full(X2),'econ');

%% Piecewise Cubic
X3 = cell2mat(phi_i(1:4));
[u3,~,~] = svd(full(X3),'econ');

%% Chebyshev Polynomial
t_cheb = (x_fine-min(x_fine))/(max(x_fine)-min(x_fine));
X_cheb = ones(length(x_fine),1);
```

```
for i = 1:n_cheb
    X_cheb = [X_cheb,X_cheb(:,end).*t_cheb];
end
[u_cheb,d_cheb,~] = svd(X_cheb,'econ');
u_cheb = u_cheb(:,1:sum(diag(d_cheb)>10*eps));

%% Comparison Plot
f = figure;
hold on

aux = abs((y-u1*(u1'*y)));
plot(x_fine,log(aux),'.-');

aux = abs((y-u3*(u3'*y)));
plot(x_fine,log(aux),'-');

aux = abs((y-u_cheb*(u_cheb'*y)));
plot(x_fine,log(aux),'--');
legend('poly deg 1','poly deg 3','global chebyshev');
legend('Location','NorthWest');
title('Local Polynomial Approximation of log(x+1e-5)');
ylabel('log of error');
ylim([-15,5]);

%% Quadratic with Boundary Constraints
%%%% Same codes as above
%% Create Partitions
n_red = 31;
lambda = 0.0001;     % Determine non-uniformity
                  % Higher lambda leads to more non-uniform distribution
x_knots = exp(lambda*(1:n_red)');
x_knots = (x_knots-x_knots(1))/(x_knots(end)-x_knots(1));

%% Create local polynomial basis functions
% Find the partition x lies in
loc = sum(bsxfun(@minus,x_knots',x_fine)<=0,2);
loc = min(loc,length(x_knots)-1);
loc = max(loc,1);

% Compute normalized value
t = (x_fine - x_knots(loc))./(x_knots(loc+1)-x_knots(loc));

% Generate monic polynomials with local support
phi_i{1} = sparse(1:length(x_fine),loc,ones(length(x_fine),1));
for i = 2:3
```

```
    phi_i{i} = bsxfun(@times,t,phi_i{i-1});
end

X2 = cell2mat(phi_i(1:3));
n_part = length(x_knots)-1;

%% Build R matrix
% Most of the time only a few lines need to be written
%    down to see the diagonal patterns
aux = spdiags(ones(n_part,3),[0,n_part,2*n_part],...
      n_part-1,3*n_part) - spdiags(ones(n_part,1),1,n_part-1,3*n_part);
spacing = diff(x_knots);
aux1 = spdiags([1./spacing(1:end-1),...
       -1./spacing(2:end),2./spacing(1:end-1)],...
       [n_part,n_part+1,2*n_part],n_part-1,3*n_part);

%% Compute Null Space of R
n_cond = size(res,1);
[~,~,res] = svd(full([aux;aux1]));
res = res(:,n_cond+1:end);
X2_res = X2*res;
[u2_res,~,~] = svd(X2_res,'econ');


%% Making Comparison Plots
plot(x_fine,y);
hold on;
plot(x_fine,u2_res*(u2_res'*y));

f = figure;
ylim([-15,5]);
hold on

aux = abs((y-u1*(u1'*y)));
plot(x_fine,log(aux),'.-');

aux = abs((y-u3*(u3'*y)));
plot(x_fine,log(aux),'-');

aux = abs((y-u_cheb*(u_cheb'*y)));
plot(x_fine,log(aux),'--');

aux = abs(u2_res*(u2_res'*y)-y);
plot(x_fine,log(aux));
legend('poly deg 1','poly deg 3','global chebyshev','quad');
```

```
legend('Location','NorthEast');
title('Local Polynomial Approximation of log(x+1e-5)');
ylabel('log of error');
ylim([-15,5]);
```

Chapter 2

Projection Methods in Solving Continuous Time Hamilton-Jacobi-Bellman Equations

2.1 Introduction

Macroeconomics today would not have exist without the advances in numerical dynamic programming. Unfortunately, most of the problem formulation has been restricted to discrete time. Part of the reason is because there is a misconception that solving a continuous time dynamic programming problem is hard. For example, Rust (1996) stated, "While there is an elegant theory of continuous time dynamic programming in which the value function V can be shown to be the solution to a system of partial differential equations ... [there is an] inherent difficulty," and the best approach might be to approximate the problem with a discrete time problem. This, however, does not represent continuous time problems in a fair light. As noted in Achdou et al. (2015), a continuous time framework actually has a few advantages over discrete time. One of the main advantages of continuous time implementation is the improvement in speed. The speed gain arises because the interaction pattern in continuous time is local. In continuous time, a person who has \$100 cares only about what his situation will be when he holds \$101 or \$99. However, in discrete time (depending on the length of time), the person has to consider the possibilities of holding

$101, $99, $102, $98, and so forth. This can be considered a local interaction even in discrete time, but this will require special approximations that would be problem specific. As macroeconomic models become more complicated, this local/sparse interaction will matter more. Therefore, as macroeconomists model heterogeneity across agents, economists will have to use the continuous time modeling technique the same way people had to pick up numerical dynamic programming.

Although the continuous time framework can be more efficiently implemented than the discrete time counterpart, the continuous time dynamic programming problem is still subject to the "curse of dimensionality." As the number of state variables increases, the number of grid points require to approximate the value function scales exponentially in the degree of the state variables. This is not unique to continuous time. Conquering the curse of dimensionality has been an active research area in discrete time. The sparse grid method proposed by Smolyak (1963) has recently been found by economists with the pioneering work of Krueger and Kubler (2004). Gordon (2011) even applied Smolyak method to heterogeneous agent models in discrete time. In the present paper, a general model reduction methodology is considered that encompasses the Smolyak method as a special case. The basic idea is to project the value function to a lower-dimensional space and solve the dynamic problem in the reduced space. This method inherits all the nice properties of modeling in continuous time. For example, in discrete time, even with the Smolyak method, some form of quadrature needs to be computed to approximate expectations. In continuous time, the resulting partial differential equation (PDE) automatically handles the expectation without the need for an integration. Judd et al. (2014) argued that fixed point iteration should be used when solving problems with the Smolyak methods because the optimization problem given the value function is a nonlinear equation that requires multiple approximations (and quadratures) on sparse grids per iteration of the nonlinear solver. However, in continuous time as Achdou et al. (2015) noted, the optimization problem is simple and can be solved analytically for the family of utility functions commonly used in practice.

The chapter is structured as follows. In section 2.2, the Hamilton-Jacobi-Bellman equation to be solved is set up. In section 2.3, the reduction method is defined, and in section 2.4, the application of the method is shown. In section 2.5, the conclusion is provided.

2.2 Problem Formulation

In economics, agents usually solve optimization problems of the form

$$\max_{a_t \in A} \int_{t=0}^{\infty} e^{-\rho t} u(a_t, x_t)\, dt \qquad \text{s.t}$$
$$dx_t = f(a_t, x_t)dt + \sigma(x_t)dW_t$$
$$x_0 = \tilde{x}$$

where $x_t \in \mathbb{R}^d$ are state variables, $a_t \in A$ are choice variables, and $f : A \times \mathbb{R}^d \to \mathbb{R}^d$ and W_t one-dimensional Brownian motions with $\sigma(x_t) \in \mathbb{R}^d$.[1] This optimization problem can be solved systematically using Hamilton-Jacobi-Bellman (HJB) equations:

$$\rho V = \max_{a \in A} u(a, x) + f(a, x)DV + \sigma^T(x)D^2V\sigma(x)$$

In most cases, analytic solutions cannot be found, and one has to rely on numerical methods. One well-known thread of literature uses finite difference methods to discretize the HJB equation. Achdou and Capuzzo-Dolcetta (2010) showed the convergence of the finite difference scheme to the viscosity solution of the HJB equation, and this has been applied to solve heterogeneous agent models in Achdou et al. (2015). This results in the discretized problem on a rectangular grid on V_{ij} of the form

$$A(V_{ij})V_{ij} = u(V_{ij})$$

[1] For notation simplicity, W_t is taken to be one-dimensional Brownian motion.

where V_{ij} is the value function stacked as a vector, $A(V_{ij})$ is a matrix representing the discretization scheme, and $u(V_{ij})$ is the utility value evaluated at the corresponding discretized coordinate taking V_{ij} as the value function.[2] Let N be the number of grid points in each dimension, so the number of total grid points of V_{ij} is N^d.[3] Because the interaction in continuous time is local, $A(V_{ij})$ is very sparse. In fact, it can have, at most, $(2d+1)$ non-zero diagonals, resulting in representation that scales at $O(dN^d)$. This enables economists to build models bigger than in discrete time where the non-local transitions would lead to problem scaling at a faster rate than $O(dN^d)$.[4]

Now, the discretized HJB equation, in theory, can be solved with any nonlinear solution methods to find V_{ij}. However, this is a very high-dimensional nonlinear equation that is not trivial to solve. Achdou and Capuzzo-Dolcetta (2010) showed that the iteration of the form

$$(A(V_{ij}^n) + \mu I)V_{ij}^{n+1} = u(V_{ij}^n) + \mu \cdot V_{ij}^n \tag{2.1}$$

where I is the identity matrix and μ is some large number has a unique fixed point. Since $(A(V_{ij}^n) + \mu I)$ is a sparse matrix, a sparse linear solver like (UMFPACK) can be used to solve for V_{ij}^{n+1} fast. With the sparse linear solvers, this iteration is fast and robust for most economics problems to which the method has been applied.

2.3 Solution on a Linear Subspace

From this point on, V denotes the discretized V instead of using V_{ij}. Recall that as the state variable grows V grows exponentially. This is the "curse of dimensionality" noted in the introduction. However, even with the curse, different parts of the solution methods hit the bottleneck faster than others. In practice, the first bottleneck in this solution method is solving the linear equation given in equation (2.1).

[2] Notation V_{ij} is used to denote discretization, and does not mean that V_{ij} is two dimensional.

[3] N can be different for different dimensions, and in fact, different N's should be used. However, N is taken to be the same in all dimensions so that N^d can be used instead of $\prod N_i$ for notation.

[4] Reiter (2010) noted that you cannot solve problems with non-sparse transitions in discrete time for this reason.

Continuous time models allow the modeler to be able to compute $(A(V^n) + \mu I)$ as it is sparse and scales at $O(dN^d)$ unlike in discrete time. Although $(A(V^n) + \mu I)$ is sparse, and hence represented efficiently in memory, solving the linear equation leads to a fill-in with a direct sparse solver, such as UMFPACK. This is because the matrix needs to be updated with intermediate values while solving for the solution, and in the calculation, it might need to store a non-zero value at a location that had initial zero. With direct methods, the fill-in will dictate the true scale of the linear equation, and the scale of this problem will determine the scale of the entire solution method. However, with the sparsity structure of A resulting from the discretization, the fill-in seems to behave worse than $O(dN^d)$.[5] Given that the direct method leads to fill-in with a worse scale than $O(dN^d)$, the iterative sparse linear solver can be tried. However, the convergence of the iterative method depends on the problem, and the convergence results were poor when they were tested.

If the scale of the linear equation can be reduced, problems with large states can be solved. Since the next bottleneck is $O(dN^d)$, if the fill-in were to be at scale $O(N^{2(d-1)})$, then we are losing about double the number of state variables due to the linear solution method.[6]

In this paper, this problem is tackled by using reducing the dimensionality of the problem on a linear subspace. Let Φ be an $N^d \times m$ matrix with orthonormalized columns representing the basis functions. With the basis functions, V can be approximated as

$$V \approx \widetilde{V} := \Phi v \qquad\qquad v = \Phi^T V$$

[5]Heuristically it seems to be $O((N^{d-1})^2)$. This makes some intuitive sense because the outermost diagonal of the matrix is at N^{d-1}. The fill-in can behave better than the largest diagonal, but in practice, this seems not to be the case.

[6]This is only about memory representation. In terms of speed, the matrix inverse scales at a higher rate than matrix $O(N^{2(d-1)})$, and this depends on the fill-in as well.

Equation (2.1) on approximated space becomes

$$\begin{aligned}(A(\Phi v^n) + \mu I)\Phi v^{n+1} &= (A(\widetilde{V}^n) + \mu I)\widetilde{V}^{n+1} \\ &= u(\widetilde{V}^n) + \mu \cdot \widetilde{V}^n \\ &= u(\Phi v^n) + \mu \cdot \Phi v^n\end{aligned}$$

Premultiplying the equation with Φ^T, i.e., projecting the problem down, we get

$$\Phi^T(A(\Phi v v^n) + \mu I)\Phi v^{n+1} = \Phi^T u(\Phi v^n) + \mu \cdot v^n$$

Hence, the linear problem that needs to be solved is of dimensionality $O(m)$.[7] Therefore, as long as m scales better than the fill-in resulting from the direct sparse linear solver, there can be a potential benefit from solving the problem in a lower-dimensional space. Of course, there is a trade-off in accuracy, but this is a trade-off that economists have faced many times. However, the problem with the scale becomes much worse in higher dimensions as each of the one-dimensional problem gets aggregated multiplicatively in high dimensions.

Although it might seem restrictive initially, the linear subspace representation underlies most of the approximation methods that exist in the literature. This is linear subspaces can be constructed intuitively in high dimensions. Many can be used as linear subspaces, such as the Smolyak method, Chebyshev polynomial approximations, sparse grids, and spline interpolations, to result in reasonable m that can behave better than N^{d-1}.

2.4 Example Applications

To show the generality of the method, an example problem will be solved with different sets of basis functions. These include intuitive approximations, such as spline interpolation, to the leading

[7] $O(m^2)$ for memory representation and $O(m^3)$ for solving the linear equation.

edge of the approximation theory, such as the Smolyak method used by economists (for example, in Judd et al. (2014)). In addition to the simplicity of the framework, the method generalizes to other potential interpolation schemes without much adjustments. For example, to the author's knowledge, this is the first time the dimension-adaptive Smolyak method has been used in economics. Refer to chapter 1 to see how spline interpolation, polynomial approximation, and the Smolyak method can be considered to be a subspace approximation.

In the following, the HJB equation resulting from the standard Aiyagari-Beweley-Huggett model is used as an example. Therefore, households are solving the optimization problem:

$$\begin{aligned}
&\max_{c_t} \int e^{-\rho t} u(c_t)\, dt \qquad s.t. \\
&da_t = r a_t + w z_t - c_t \\
&dz_t = \mu(z) dt + \sigma(z) dW_t \\
&a_t \geq \underline{a}
\end{aligned}$$

This results in the HJB equation of the form

$$\rho v(a,z) = \max_c u(c) + D_a v(a,z)(wz + ra - c) + \mu(z) D_z v(a,z) + \frac{1}{2}\sigma^2(z) D_{zz} v(a,z)$$

The discretization is done with 100 grid points in asset levels and 40 grid points in skill levels. The reduction was done to 200 grid points (from 4000) for a clear comparison across different subspaces being used.

2.4.1 Tensor Constructions

Following the notation from chapter 1, tensor grids can be built using basis functions in one direction at a time. As the first example, polynomial (Chebyshev polynomial) basis functions can be used for one dimension, while the canonical/identity basis functions are used in the other.

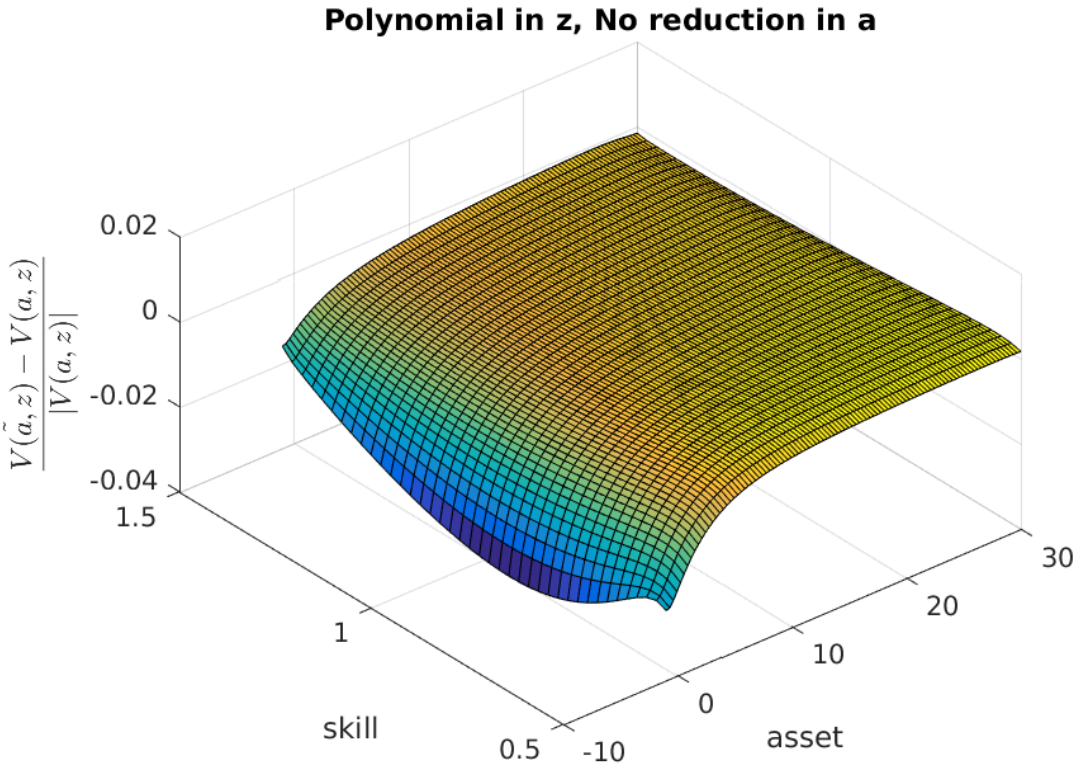


Figure 2.1: Reduction in the skill direction

(using polynomial basis functions)

When a polynomial basis functions of degree 2 is used in the skill dimension, the reduction does very poorly resulting in $\|\widetilde{V} - V\|_\infty = 0.037909$ and $\|\widetilde{V} - V\|_2 = 0.21573$.[8] As can be seen from figure 2.1, the error occurs very systematically, as it fails to capture the complex behavior at the borrowing constraint.

The result is marginally better when polynomial basis functions are used in a with $\|\widetilde{V} - V\|_\infty = 0.030658$ and $\|\widetilde{V} - V\|_2 = 0.0162547$. As also can be seen from figure 2.2, this tensor structure fails to capture the complexity near the borrowing constraint.

Things behave very well when tensor products of different bases are used in both dimensions. Part of the reason the approximations failed when only one dimension is reduced is because the

[8] $\|\widetilde{V} - V\|_\infty$ and $\|\widetilde{V} - V\|_2$ denote weighted norms weighted by the inverse of $|V(a,z)|$, i.e., the norms are computed on relative errors.

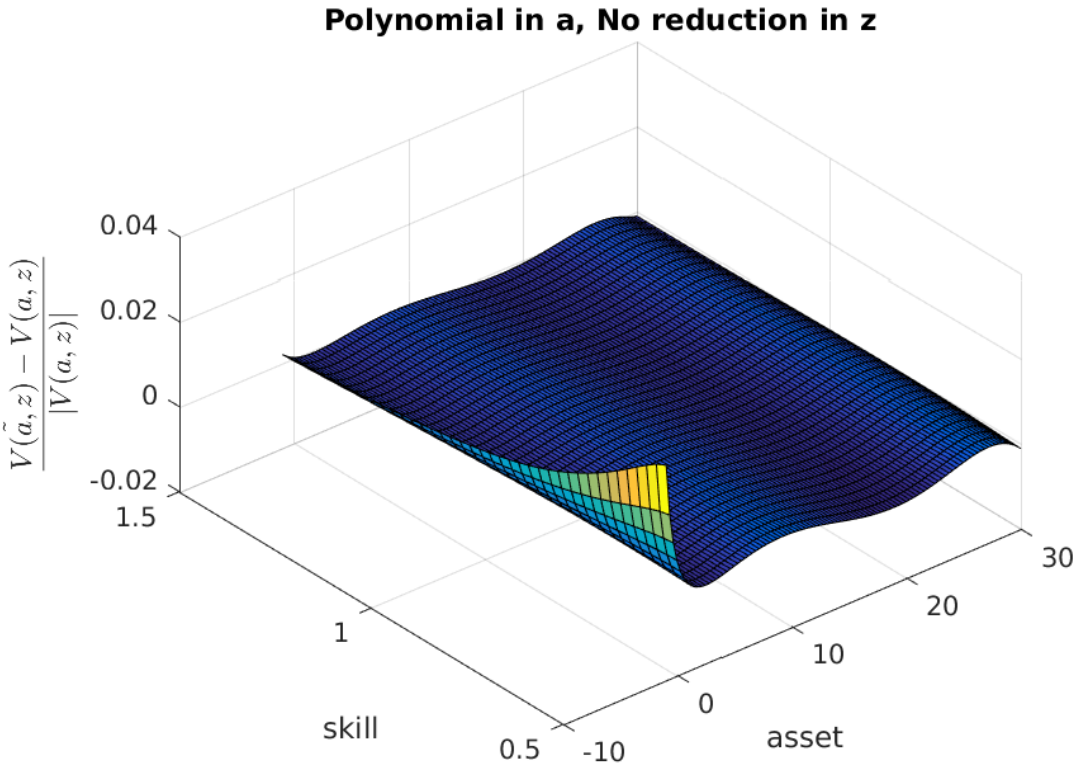


Figure 2.2: Reduction in the asset direction

only using polynomial basis functions

20-fold reduction arising from going from 4000 dimension to 200 dimenion imposes an unrealistic constraint on the dimension of the one dimension being reduced. The reduction should be taken in both directions. When non-uniformly spaced spline approximation is used in each direction, the approximation error is $\|\widetilde{V} - V\|_\infty = 4.0465 \times 10^{-4}$ and $\|\widetilde{V} - V\|_2 = 8.1403 \times 10^{-6}$.

With a polynomial basis of dimension 20 in the asset direction and polynomial basis of dimension 10 in the skill direction, the approximation errors are $\|\widetilde{V} - V\|_\infty = 9.6616 \times 10^{-5}$ and $\|\widetilde{V} - V\|_2 = 3.0246 \times 10^{-6}$ with the error shown in 2.4.

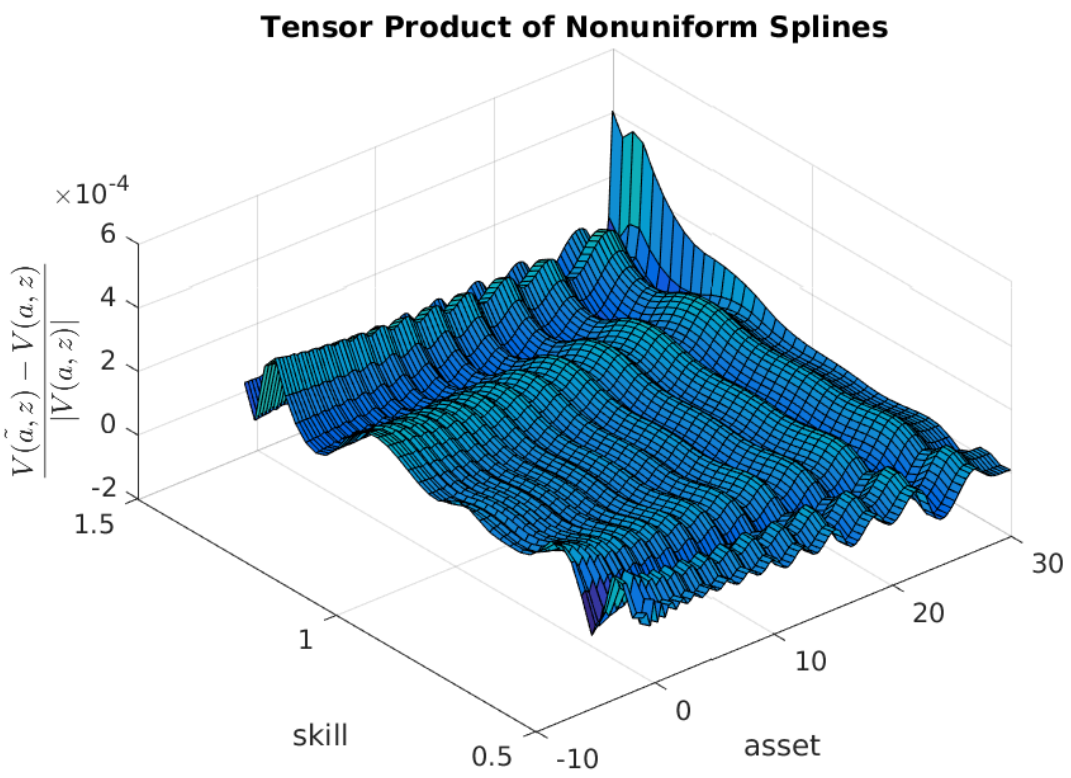


Figure 2.3: Tensor product basis function 1

(with a quadratic spline with the continuously differentiability condition)

2.4.2 Generalized Total Degree

Note the error patterns given in figure 2.4 and figure 2.3. When people do polynomial approximations, they take all polynomials of a certain degree. However, this is not the case when tensor product basis functions are used. For example,

$$\{1, x, x^2\} \otimes \{1, y, y^2\} = \{1, x, y, x^2, xy, y^2, x^2y, xy^2, x^2y^2\}$$

Therefore, of the third-order polynomial terms, x^3 and y^3 are missing, and of fourth-order polynomial terms, only two of six are included. Therefore, when approximations are generalized to higher orders, the basis functions taken from the product structure need not match those found

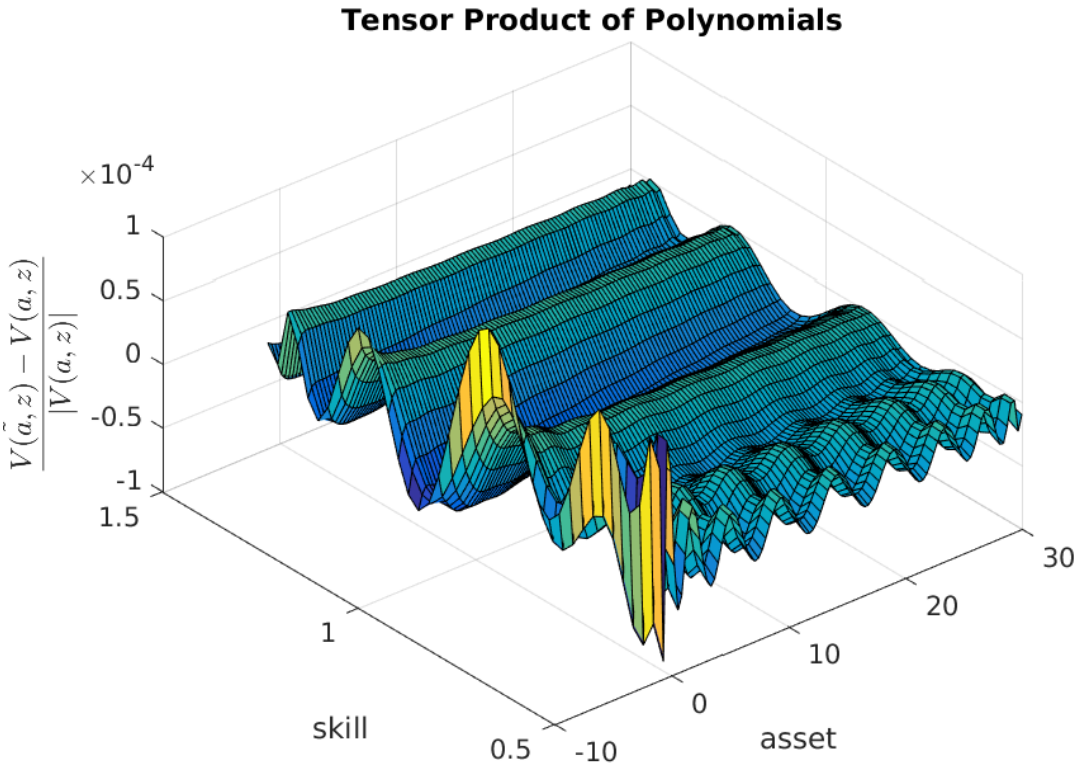


Figure 2.4: Tensor basis function 2

(with polynomials with dimension 20 in a and dimension 10 in z)

from tensor products. This idea can be consolidated using generalized degree function g. Let $o(\phi_i)$ be the order/degree of a given basis function, and let $g : \mathbb{R}^d \to \mathbb{R}$ be the generalized degree. For example, with $o(x^n) = n$,

$$g(o(\phi_1), o(\phi_2), \ldots, o(\phi_d)) = \sum o(\phi_i)$$

defines the degree of the polynomial term as the generalized degree. Now, the generalized degree function can be used to define a set of basis functions by

$$\{\phi_{i_1}(x_1) \cdot \phi_{i_2}(x_2) \cdot \ldots \cdot \phi_{i_d}(x_d)\}_{i_1,\ldots,i_d} \qquad s.t.$$
$$g(o(\phi_{i_1}), o(\phi_{i_2}), \ldots, o(\phi_{i_d})) \leq K$$

With the polynomial example above, this will result in basis functions of polynomial terms with a maximum degree of K.

Note that the degree function does not impose symmetry, and in the literature, using an asymmetric g is called using anisotropic grids.[9] In the Aiyagari-Beweley-Huggett example, the asset needs to be approximated with higher dimensions than the skill level direction, and they were taken into the approximation with the tensor grid construction in the previous section. Hence, in this case, the generalized degree function given as

$$g(o(\phi_a), o(\phi_z)) = \mu_a o(\phi_a) + \mu_z o(\phi_z)$$

where $\mu_z > \mu_a$ results in anisotropic basis functions where higher-order terms are used in the asset direction instead of in the skill direction. For polynomial basis functions, taking $\frac{\mu_z}{\mu_a} = 3$ works well with resulting $\|\widetilde{V} - V\|_\infty = 6.9369 \times 10^{-5}$ and $\|\widetilde{V} - V\|_2 = 7.0275 \times 10^{-7}$.

Following the intuition for working with different degrees, the anisotropic total degree polynomial basis function works better than the (anisotropic) tensor polynomial construction[10] that ended up with more basis functions. This can be seen more clearly when the error terms are compared at different locations shown in figure 2.4 and figure 2.5. The shape of the error terms shows more prominent polynomial patterns in the tensor product case than the anisotropic with degree function g given above. Last, for completeness, with an isotropic grid with polynomials, the errors are $\|\widetilde{V} - V\|_\infty = 3.9228 \times 10^{-4}$ and $\|\widetilde{V} - V\|_2 = 1.0830 \times 10^{-5}$, which is worse than just anisotropic tensor product construction, and the error shown in figure 2.6 shows an even more polynomial pattern that is missed by the subspace. This also confirms the intuition that people should carry when they build basis functions over higher dimensions. The generalized degree function construction helps in picking the basis functions in the correct order, but sometimes, the asymmetry between dimensions matters more than just getting the degree number correctly. This intuition will carry over to higher dimensions, but the expected gain from collecting the correct degree is higher in

[9] It was first introduced to economics for the Smolyak methods by Judd et al. (2014).

[10] Creating an anisotropic polynomial in the tensor structure is straightforward, so it rarely gets emphasized, but even tensor product construction was built anisotropically.

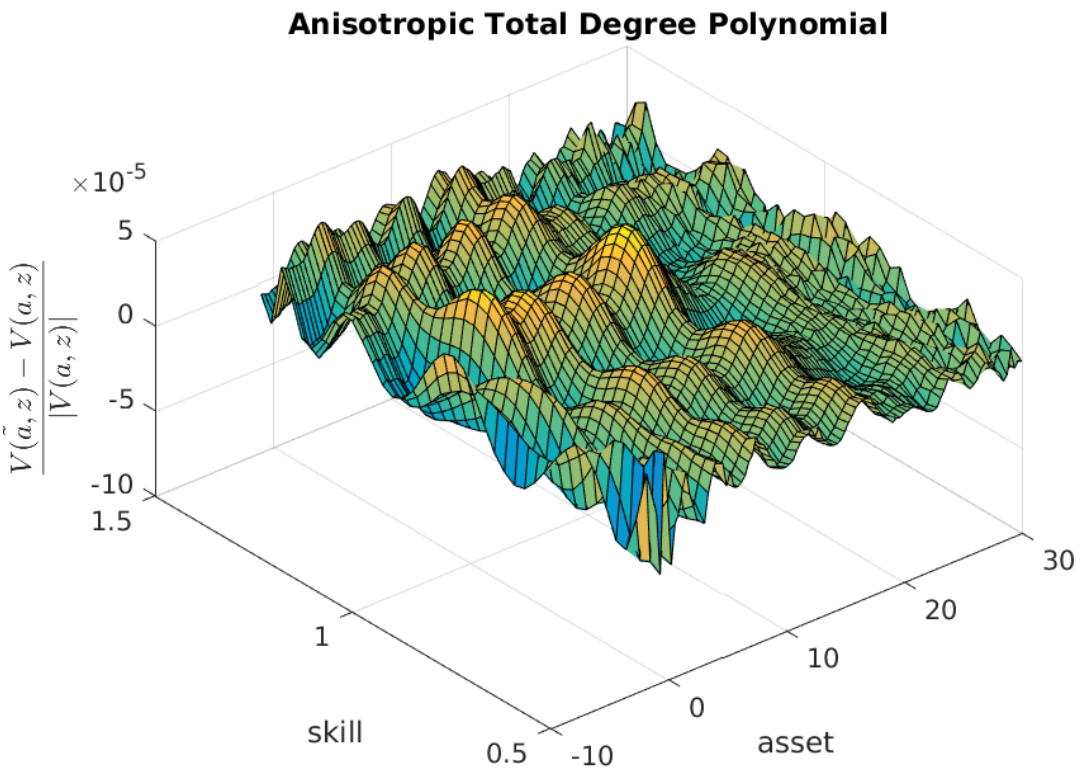


Figure 2.5: Anisotropic total polynomial degree basis function

(generated with $\frac{\mu_z}{\mu_a} = 3$ (μ_a normalized to 1) with K = 32, which results in 198 basis functions.)

higher dimensions. For example, in three dimensions, the tensor product of the cubic polynomials in each coordinate introduces a ninth-degree polynomial term ($x^3y^3z^3$) when more than half of the terms have a polynomial degree of less than three.

Now, polynomials are not the only one-dimensional basis function that you can use. For example, you can use tent functions (hierarchical surplus functions). With tent functions the degree (usually called the level in the literature) of the function is given by $o(\phi_{l,p}) = l$. Using the same weighted sum degree function g with $\frac{\mu_x}{\mu_a}$ results in the approximation given by $\|\widetilde{V} - V\|_\infty = 0.0053961$ and $\|\widetilde{V} - V\|_2 = 0.0024601$. This result is disappointing, but the reason why this happens can be seen in figure 2.7. With the anisotropic positioning of the tent functions, there are too many basis functions where the asset levels and the skill levels are high, whereas

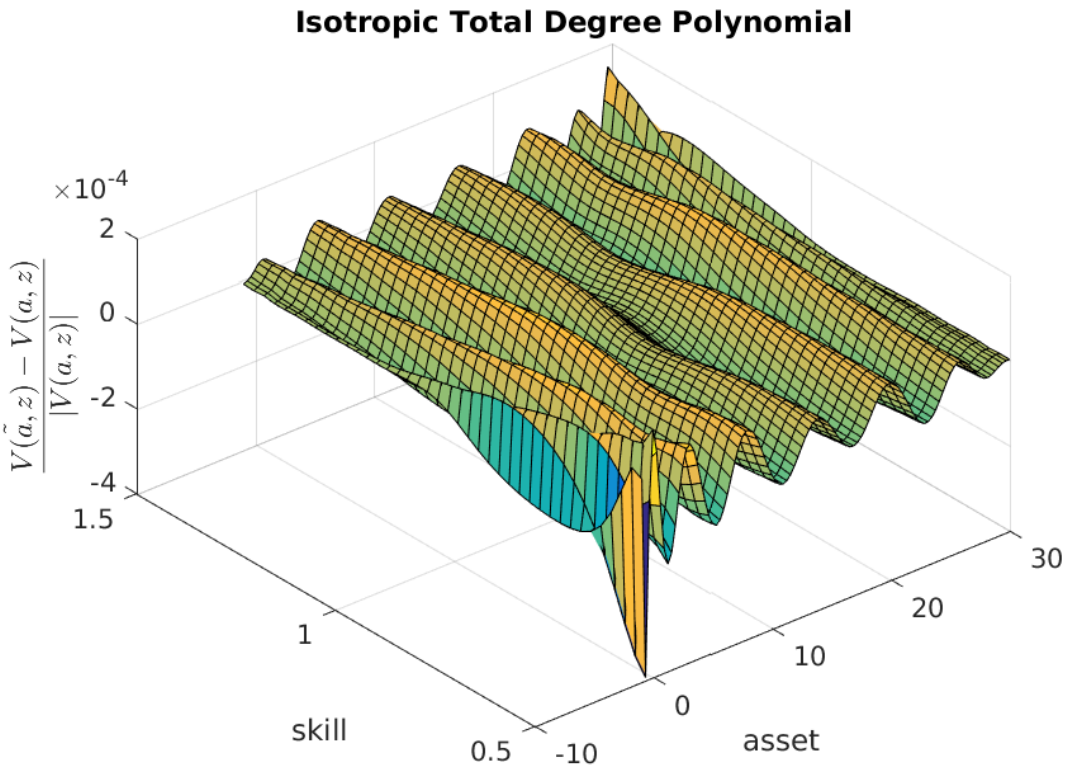


Figure 2.6: Isotropic total polynomial degree basis function

(generated with $\frac{\mu_z}{\mu_a} = 1$ (μ_a normalized to 1) with K = 14, which results in 196 basis functions.)

there are not enough basis functions to approximate the behavior well enough at the boundary with borrowing constraints. Tent functions are very localized; therefore, adding points away from the borrowing constraints does not improve approximation at all. Thus, the approximation fails to improve around the corner. Part of the reason is because the anisotropic grid was only "level(l)" specific and did not use a function that included "position(p)." This can be improved by using order function $o(\phi_{i,p})$ that uses l and p, but it would be better to implement local adaptivity instead of creating a complex degree function.

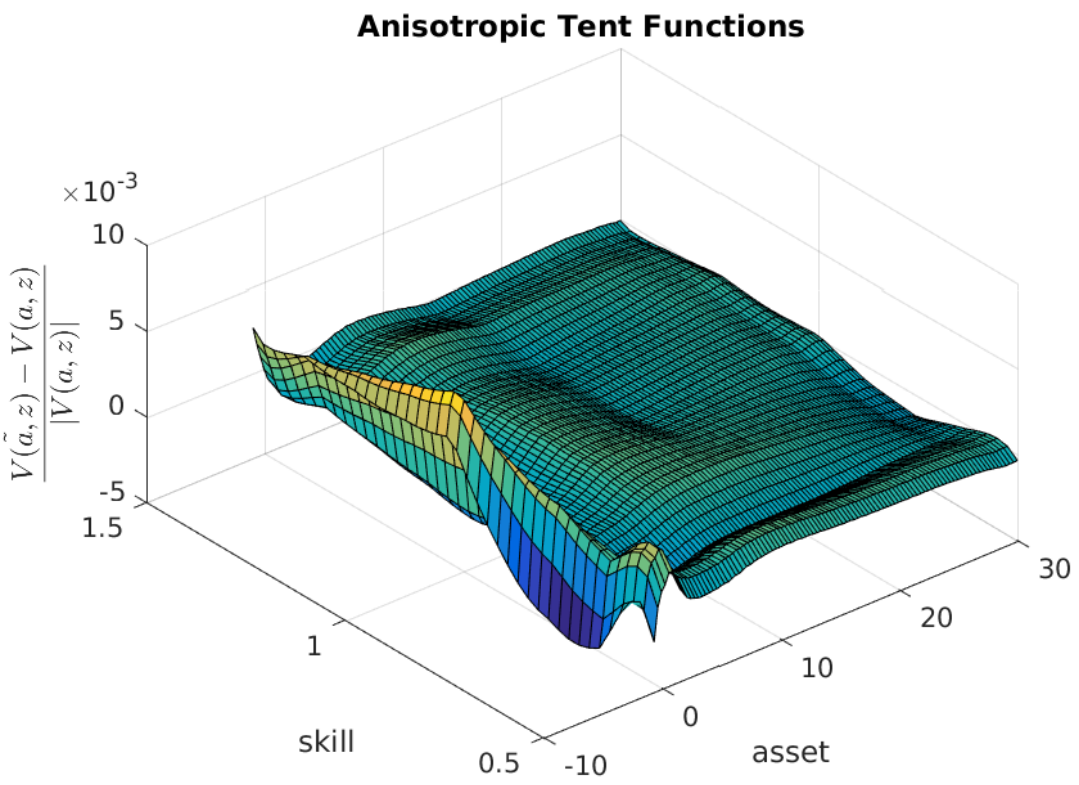


Figure 2.7: Anisotropic tent function basis function

(generated with $\frac{\mu_z}{\mu_a} = 2$ (μ_a normalized to 1) with K = 7, which results in 165 basis functions.)

2.4.3 Adaptivity

In theory, a very specialized degree function g can be constructed for a particular problem of interest. For example, even in the anisotropic case given above, $\frac{\mu_z}{\mu_a}$ was calibrated to find a good approximation. This can be done by researchers, but this process can be automated. As with the definition of g, the definition for the adaptive process can be very general. In practice, Gerstner and Griebel (2003) noted that dimension adaptivity works very well.

As the intuition is clear, no clarity will be gained from introducing additional notation. Hence, only an example case for polynomials will be given, but the general procedure should be clear. Let $\mathfrak{B} \subset \mathbb{Z}^d_{\geq 0}$ denote the set of degrees to include for the basis functions. Now, given $\mathfrak{B}$, the set of basis functions is $\{x_1^{i_1} \cdot \ldots \cdot x_d^{i_d} : (i_1, \ldots, i_d) \in \mathfrak{B}\}$. Given a set of basis functions $\mathfrak{B}_n$,

the approximated solution can be solved on the subspace generated by $\mathfrak{B}_n$. Let $a_{(i_1,\ldots,i_d)}$ be the coefficients that approximate value function; i.e.,

$$\widetilde{V} = \sum_{(i_1,\ldots,i_d)\in\mathfrak{B}_n} a_{(i_1,\ldots,i_d)} x_1^{i_1} \cdot \ldots \cdot x_d^{i_d}$$

Then, a new subspace can be created by the rule

$$\begin{aligned}
\mathfrak{B}_{n+1} = \mathfrak{B}_n \cup \{(i_1+1,\ldots,i_d) : \quad & (i_1,i_2,\ldots,i_d) \in \mathfrak{B}_n \\
& a_{(i_1,\ldots,i_d)} > \alpha, \\
& \text{if } i_2 > 0, (i_1+1, i_2-1, \ldots, i_d) \in \mathfrak{B}_n, \\
& \ldots \\
& \text{if } i_d > 0, (i_1+1,\ldots,i_d-1) \in \mathfrak{B}_n\} \\
\cup \{(i_1,i_2+1,\ldots,i_d) : \quad & (i_1,i_2,\ldots,i_d) \in \mathfrak{B}_n \\
& a_{(i_1,\ldots,i_d)} > \alpha, \\
& \text{if } i_1 > 0, (i_1-1, i_2+1, \ldots, i_d) \in \mathfrak{B}_n, \\
& \ldots \\
& \text{if } i_d > 0, (i_1,i_2+1\ldots,i_d-1) \in \mathfrak{B}_n\} \\
& \vdots \\
\cup \{(i_1,i_2,\ldots,i_d+1) : \quad & (i_1,i_2,\ldots,i_d) \in \mathfrak{B}_n \\
& a_{(i_1,\ldots,i_d)} > \alpha, \\
& \text{if } i_1 > 0, (i_1-1, i_2, \ldots, i_d+1) \in \mathfrak{B}_n, \\
& \ldots \\
& \text{if } i_{d-1} > 0, (i_1,\ldots,i_{d-1}+1,i_d+1) \in \mathfrak{B}_n\}
\end{aligned}$$

for some given cutoff $\alpha > 0$.

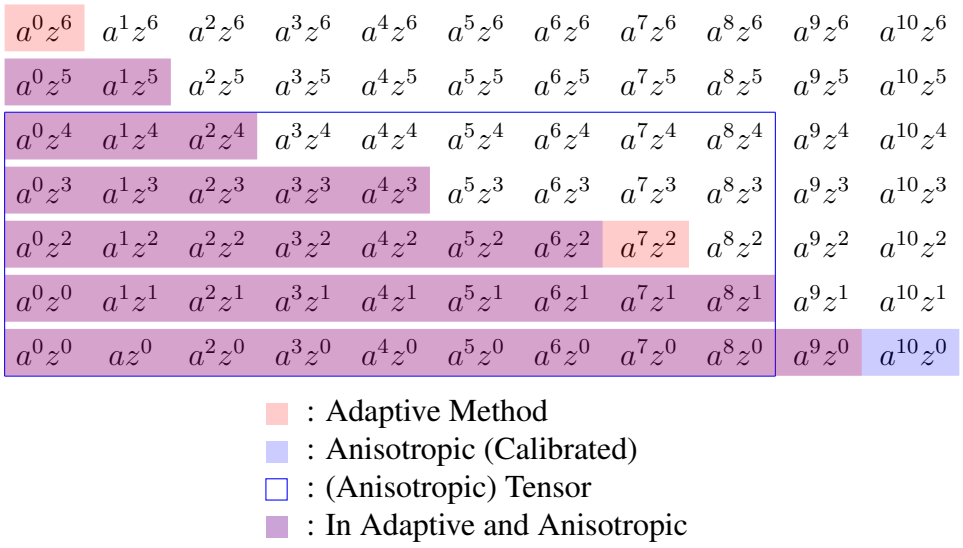


Figure 2.8: Inclusion pattern of polynomials

Example of how polynomials are used in different methods of generalizing to higher dimensions. Things were matched so that the result would have 36 basis functions (a small difference since adaptive procedure introduces multiple basis vectors at a time). The tensor case has 36 basis functions, anisotropic 37, and the adaptive 38.

	Tensor	Anisotropic	Adaptive
$\left\|\widetilde{V}(a,z) - V(a,z)\right\|_\infty$	0.01159	0.0094944	0.0040667
$\left\|\widetilde{V}(a,z) - V(a,z)\right\|_2$	0.075054	0.015330	0.0085258

Figure 2.9: Errors from approximation using different polynomial basis functions

This is seems complex, but it is a simple rule. For example, it just says if $\alpha_{(1,2)} > \alpha$, then of the candidate basis functions of x^2y^2 and xy^3, introduce them only if lower-order terms; i.e., x^2y and y^3, respectively, are already in the set of basis functions. This is the dimension adaptivity advocated by Gerstner and Griebel (2003). For example, when this adaptivity is applied on an isotropic polynomial basis function (with $\frac{\mu_z}{\mu_a} = 1$ and $K = 3$ in the above notation), the basis function generated follows the anisotropic pattern that is found by calibrating $\frac{\mu_z}{\mu_a}$ with a known solution. This can be seen in figure 2.8. Comparing the basis structure given in figure 2.8, we get that the errors are much better for the adaptively built basis functions than the tensor product basis functions.

When this adaptive method is applied to the Aiyagari-Beweley-Huggett case above, the method achieves errors of $\|\widetilde{V} - V\|_\infty = 7.9732 \times 10^{-5}$ and $\|\widetilde{V} - V\|_2 = 1.2233 \times 10^{-6}$. This is approximately as good as the anisotropic total degree polynomial. This actually means that the adaptive method works pretty well as the anisotropic case was calibrated with a known solution to get a good approximation. However, the adaptive methods has its own tuning parameter of α; therefore, both the anisotropic and the adaptive methods should be systematic methods to keep the basis functions jointly instead of independently. In addition, the adaptivity used for the polynomial basis is somewhat unnatural since the problem coefficients do not represent their local impact. However, this can be adapted with the greedy search method of looking for potential basis functions; thus, the generality of the method remains.

2.5 Conclusion

The ability to solve a given model has always been the limit of modeling. As economists push the frontier of solvable models, the sparsity of model representation and the solution methods for finding the solutions will matter more. Continuous time models provide a structured method for representing a model very sparsely allowing the researcher to set up more complicated models than in discrete time. However, there is still a solution bound in the continuous time modeling method. In this chapter, a general reduction method has been shown that reduces the dimensionality of the problem immensely.

Chapter 3

When Inequality Matters for Macro and Macro Matters for Inequality [1]

Abstract. We develop an efficient and easy-to-use computational method for solving a wide class of general equilibrium heterogeneous agent models with aggregate shocks. Our method extends standard linearization techniques and is designed to work in cases when inequality matters for the dynamics of macroeconomic aggregates. We present two applications that analyze a two-asset incomplete markets model parameterized to match the distribution of income, wealth, and marginal propensities to consume. First, we show that our model is consistent with two key features of aggregate consumption dynamics that are difficult to match with representative agent models: (i) the sensitivity of aggregate consumption to predictable changes in aggregate income and (ii) the relative smoothness of aggregate consumption. Second, we extend the model to feature capital-skill complementarity and show how factor-specific productivity shocks shape dynamics of income and consumption inequality.

3.1 Introduction

Over the last thirty years, tremendous progress has been made in developing models that reproduce salient features of the rich heterogeneity in income, wealth, and consumption behavior across households that is routinely observed in micro data. These heterogeneous agent models often de-

[1] This chapter is a coauthored work with Greg Kaplan, Benjamin Moll, Thomas Winberry, and Christian Wolf forthcoming in NBER Macroannual as Ahn et al. (2017).

liver strikingly different implications of monetary and fiscal policies than do representative agent models, and allow us to study the distributional implications of these policies across households.[2] In principle, this class of models can therefore incorporate the rich interaction between inequality and the macroeconomy that characterizes our world: on the one hand, inequality shapes macroeconomic aggregates; on the other hand, macroeconomic shocks and policies affect inequality.

Despite providing a framework for thinking about these important issues, heterogeneous agent models are not yet part of policy makers' toolbox for evaluating the macroeconomic and distributional consequences of their proposed policies. Instead, most quantitative analyses of the macroeconomy, particularly in central banks and other policy institutions, still employ representative agent models. Applied macroeconomists tend to make two excuses for this abstraction. First, they argue that the computational difficulties involved in solving and analyzing heterogeneous agent models render their use intractable, especially compared to the ease with which they can analyze representative agent models using software packages like `Dynare`. Second, there is a perception among macroeconomists that models which incorporate realistic heterogeneity are unnecessarily complicated because they generate only limited additional explanatory power for aggregate phenomena. Part of this perception stems from the seminal work of Krusell and Smith (1998), who found that the business cycle properties of aggregates in a baseline heterogeneous agent model are virtually indistinguishable from those in the representative agent counterpart.[3,4]

Our paper's main message is that both of these excuses are less valid than commonly thought. To this end, we make two contributions. First, we develop an efficient and easy-to-use computational method for solving a wide class of general equilibrium heterogeneous agent macro models

[2]For examples studying fiscal policy, see McKay and Reis (2013) and Kaplan and Violante (2014); for monetary policy, see McKay, Nakamura and Steinsson (2015), Auclert (2014), and Kaplan, Moll and Violante (2016).

[3]More precisely, in Krusell and Smith (1998)'s baseline model, which is a heterogenous agent version of a standard Real Business Cycle (RBC) model with inelastic labor supply, the effects of technology shocks on aggregate output, consumption and investment are indistinguishable from those in the RBC model.

[4]Lucas (2003) succinctly captures many macroeconomists' view when he summarizes Krusell and Smith's findings as follows: "For determining the behavior of aggregates, they discovered, realistically modeled household heterogeneity just does not matter very much. For individual behavior and welfare, of course, heterogeneity is everything." Interestingly, there is a discrepancy between this perception and the results in Krusell and Smith (1998): they show that an extension of their baseline model with preference heterogeneity, thereby implying more realistic wealth distribution, "features aggregate time series that depart significantly from permanent income behavior."

with aggregate shocks, thereby invalidating the first excuse. Importantly, our method also applies in environments that violate what Krusell and Smith (1998) have termed "approximate aggregation", i.e. that macroeconomic aggregates can be well described using only the mean of the wealth distribution.

Second, we use the method to analyze the time series behavior of a rich two-asset heterogeneous agent model parameterized to match the distribution of income, wealth, and marginal propensities to consume (MPCs) in the micro data. We show that the model is consistent with two features of the time-series of aggregate consumption that have proven to be a challenge for representative agent models: consumption responds to predictable changes in income but at the same time is substantially less volatile than realized income. We then demonstrate how a quantitatively plausible heterogeneous agent economy such as ours can be useful in understanding the distributional consequences of aggregate shocks, thus paving the way for a complete analysis of the transmission of shocks to inequality. These results invalidate the second excuse: not only does macro matter for inequality, but inequality also matters for macro. We therefore view an important part of the future of macroeconomics as the study of distributions – the representative-agent shortcut may both miss a large part of the story (the distributional implications) and get the small remaining part wrong (the implications for aggregates).

In Section 3.2, we introduce our computational methodology, which extends standard linearization techniques, routinely used to solve representative agent models, to the heterogeneous agent context.[5] For pedagogical reasons, we describe our methods in the context of the Krusell and Smith (1998) model, but the methods are applicable much more broadly. We first solve for the stationary equilibrium of the model *without* aggregate shocks (but with idiosyncratic shocks) using a global non-linear approximation. We use the finite difference method of Achdou et al. (2015) but, in principle, other methods can be used as well. This approximation gives a discretized rep-

[5] As we discuss in more detail below, the use of linearization to solve heterogeneous agent economies is not new. Our method builds on the ideas of Dotsey, King and Wolman (1999), Campbell (1998), and Reiter (2009), and is related to Preston and Roca (2007). In contrast to these contributions, we cast our linearization method in continuous time. While discrete time poses no conceptual difficulty, working in continuous time has a number of numerical advantages that we heavily exploit.

resentation of the model's stationary equilibrium, which includes a non-degenerate distribution of agents over their individual state variables. We then compute a first-order Taylor expansion of the discretized model *with* aggregate shocks around the stationary equilibrium. This results in a large, but linear, system of stochastic differential equations, which we solve using standard solution techniques. Although our solution method relies on linearization with respect to the economy's aggregate state variables, it preserves important non-linearities at the micro level. In particular, the response of macroeconomic aggregates to aggregate shocks may depend on the distribution of households across idiosyncratic states because of heterogeneity in the response to the shock across the distribution.

Our solution method is both faster and more accurate than existing methods. Of the five solution methods for the Krusell and Smith (1998) model included in the *Journal of Economic Dynamics and Control* comparison project (Den Haan (2010)), the fastest takes around 7 minutes to solve. With the same calibration our model takes around a quarter of a second to solve. The most accurate method in the comparison project has a maximum aggregate policy rule error of 0.16% (Den Haan (2010)'s preferred accuracy metric). With a standard deviation of productivity shocks that is comparable to the Den Haan, Judd and Julliard (2010) calibration, the maximum aggregate policy rule error using our method is 0.05%. Since our methodology uses a linear approximation with respect to aggregate shocks, the accuracy worsens as the standard deviation of shocks increases.[6]

However, the most important advantage of our method is not its speed or accuracy for solving the Krusell and Smith (1998) model. Rather, it is the potential for solving much larger models in which approximate aggregation does not hold and existing methods are infeasible. An example is the two-asset model of Kaplan, Moll and Violante (2016), where the presence of three individual state variables renders the resulting linear system so large that it is numerically impossible to solve. In order to be able to handle larger models such as this, in Section 3.3 we develop a model-free reduction method to reduce the dimensionality of the system of linear stochastic differential

[6]See Table 16 of Den Haan (2010). See Section 3.2 for a description of this error metric and how we compare our continuous-state, continuous-time productivity process with the two-state, discrete-time productivity process in Den Haan (2010)

equations that characterizes the equilibrium. Our method generalizes Krusell and Smith (1998)'s insight that only a small subset of the information contained in the cross-sectional distribution of agents is required to accurately forecast the variables that agents need to know in order to solve their decision problems. Krusell and Smith (1998)'s procedure posits a set of moments that capture this information based on economic intuition, and verifies its accuracy ex-post using a forecast-error metric; our method instead leverages advances in engineering to allow the computer to identify the necessary information in a completely model-free way.[7]

To make these methods as accessible as possible, and to encourage the use of heterogeneous agent models among researchers and policy-makers, we are publishing an open source suite of codes that implement our algorithms in an easy-to-use toolbox.[8] Users of the codes provide just two inputs: (i) a function that evaluates the discretized equilibrium conditions; and (ii) the solution to the stationary equilibrium *without* aggregate shocks. Our toolbox then solves for the equilibrium of the corresponding economy *with* aggregate shocks – linearizes the model, reduces the dimensionality, solves the system of stochastic differential equations and produces impulse responses.[9]

In Sections 3.4 and 3.5 we use our toolbox to solve a two-asset heterogeneous agent economy inspired by Kaplan and Violante (2014) and Kaplan, Moll and Violante (2016), in which households can save in liquid and illiquid assets. In equilibrium, illiquid assets earn a higher return than liquid assets because they are subject to a transaction cost. This economy naturally generates "wealthy hand-to-mouth" households – households who endogenously choose to hold all their wealth as illiquid assets, and to set their consumption equal to their disposable income. Such households have high MPCs, in line with empirical evidence presented in Johnson, Parker and Souleles (2006), Parker et al. (2013) and Fagereng, Holm and Natvik (2016). Because of the

[7]More precisely, we apply tools from the so-called *model reduction* literature, in particular Amsallem and Farhat (2011) and Antoulas (2005). We build on Reiter (2010) who first applied these ideas to reduce the dimensionality of linearized heterogeneous agent models in economics.

[8]The codes will be initially released as a `Matlab` toolbox, but we hope to make them available in other languages in future releases. Also see the Heterogeneous Agent Resource and toolKit (HARK) by Carroll et al. (2016, available at `https://github.com/econ-ark/HARK`) for another project that shares our aim of encouraging the use of heterogeneous agent models among researchers and policy-makers by making computations easier and faster.

[9]We describe our methodology in the context of incomplete markets models with heterogeneous households, but the toolbox is applicable for a much broader class of models. Essentially any high dimensional model in which equilibrium objects are a smooth function of aggregate states can be handled with the linearization methods.

two-asset structure and the presence of the wealthy hand-to-mouth, the parameterized model can match key features of the joint distribution of household portfolios and MPCs - properties that one-asset models have difficulty in replicating.[10] Matching these features of the data leads to a failure of approximate aggregation, which together with the model's size, render it an ideal setting to illustrate the power of our methods. To the best of our knowledge, this model cannot be solved using any existing methods.

In our first application (Section 3.4) we show that inequality can matter for macro aggregates. We demonstrate that the response of aggregate consumption growth to a shock to productivity growth is substantially smaller and more persistent in the two-asset economy than in either the corresponding representative agent or one-asset heterogeneous agent economies. Matching the wealth distribution, in particular the consumption-share of hand-to-mouth households, drives this finding – since hand-to-mouth households are limited in their ability to immediately increase consumption in response to higher future income growth, their impact consumption response is weaker, and their lagged consumption response is stronger, than the response of non hand-to-mouth households. An implication of these individual-level consumption dynamics is that the two-asset model outperforms the representative agent models in terms of its ability to match the smoothness and sensitivity of aggregate consumption.[11] Jointly matching these two features of aggregate consumption dynamics has posed a challenge for many benchmark models in the literature (Campbell and Mankiw (1989),Christiano (1989), Ludvigson and Michaelides (2001)). Our two-asset heterogeneous agent model is instead consistent with both of these features of the data.[12]

[10]One-asset heterogeneous agent models models, in the spirit of Aiyagari (1994) and Krusell and Smith (1998), endogenize the fraction of hand-to-mouth households with a simple borrowing constraint. Standard calibrations of these models which match the aggregate capital-income ratio feature far too few high-MPC households relative to the data. In contrast when these models are calibrated to only *liquid* wealth, they are better able to match the distribution of MPCs in the data. Such economies, however, grossly understate the level of aggregate capital, and so are ill-suited to general equilibrium settings. They also miss almost the entire wealth distribution, and so are of limited use in studying the effects of macro shocks on inequality.

[11]"Sensitivity" is a term used to describe how aggregate consumption responds more to predictable changes in aggregate income than implied by benchmark representative agent economies. "Smoothness" is a term used to describe how aggregate consumption growth is less volatile, relative to aggregate income growth, than implied by benchmark representative agent economies.

[12]If we confine ourselves to matching aggregate data only, there are of course a number of other alternative models that have the potential to jointly match sensitivity and smoothness. These include the spender-saver model (Campbell

In our second application (Section 3.5) we show that macro shocks can additionally matter for inequality, resulting in rich interactions between inequality and the macroeconomy. To clearly highlight how quantitatively realistic heterogeneous agent economies such as ours can be useful in understanding the distributional consequences of aggregate shocks, in Section 3.5 we relax the typical assumption in incomplete market models that the cross-sectional distribution of labor income is exogenous. We adopt a nested CES production function with capital-skill complementarity as in Krusell et al. (2000), in which high skilled workers are more complementary with capital in production than are low skilled workers. First, we show how a positive shock to the productivity of capital generates a boom that disproportionately benefits high-skilled workers, thus leading to an increase in income and consumption inequality. Second, we show how a negative shock to the productivity of unskilled labor generates a recession that disproportionately hurts low-skilled workers, thus also leading to an increase in income and consumption inequality. The response of aggregate consumption to both of these aggregate shocks differs dramatically from that in the representative agent counterpart, thereby providing a striking counterexample to the main result of Krusell and Smith (1998). These findings illustrate how different aggregate shocks shape the dynamics of inequality and may generate rich interactions between inequality and macroeconomic aggregates.

3.2 Linearizing Heterogeneous Agent Models

We present our computational method in two steps. First, in this section we describe our approach to linearizing heterogeneous agent models. Second, in Section 3.3 we describe our model-free reduction method for reducing the size of the linearized system. We separate the two steps because the reduction step is only necessary for large models.

and Mankiw (1989)), which directly assumes that an exogenous fraction of households are permanently hand-to-mouth, and models with habit formation or "sticky consumption" (Carroll, Slacalek and Sommer, 2011).

We describe our method in the context of the Krusell and Smith (1998) model. This model is a natural expository tool because it is well-known and substantially simpler than the two-asset model in Section 3.4. As we show in Section 3.4, our method is applicable to a broad class of models.

Continuous Time We present our method in continuous time. While discrete time poses no conceptual difficulty (in fact, Campbell (1998), Dotsey, King and Wolman (1999), and Reiter (2009) originally proposed this general approach in discrete time), working in continuous time has three key numerical advantages that we heavily exploit.

First, it is easier to capture occasionally binding constraints and inaction in continuous time than in discrete time. For example, the borrowing constraint in the Krusell and Smith (1998) model below is absorbed into a simple boundary condition on the value function and therefore the first-order condition for consumption holds with equality everywhere in the interior of the state space. Occasionally binding constraints and inaction are often included in heterogeneous agent models in order to match features of micro data.

Second and related, first-order conditions characterizing optimal policy functions typically have a simpler structure than in discrete time and can often be solved by hand.

Third, and most importantly in practice, continuous time naturally generates sparsity in the matrices characterizing the model's equilibrium conditions; intuitively, continuously moving state variables like wealth only drift an infinitesimal amount in an infinitesimal unit of time, and therefore a typical approximation that discretizes the state space has the feature that households reach only states that directly neighbor the current state. Our two-asset model in Section 3.4 is so large that sparsity is necessary to store and manipulate these matrices.[13]

[13] As Reiter (2010) notes in his discussion of a related method "For reasons of computational efficiency, the transition matrix [...] should be sparse. With more than 10000 state variables, a dense [transition matrix] might not even fit into computer memory. Economically this means that, from any given individual state today (a given level of capital, for example), there is only a small set of states tomorrow that the agent can reach with positive probability. The level of sparsity is usually a function of the time period. A model at monthly frequency will probably be sparser, and therefore easier to handle, then a model at annual frequency." We take this logic a step further by working with a continuous-time model. As Reiter's discussion makes clear, discrete-time models can also generate sparsity in particular cases. However, this will happen either in models with very short time periods (as suggested by Reiter) which are known to be difficult to solve because the discount factor of households is close to one; or the resulting matrices will be sparse

3.2.1 Model Description

Environment There is a continuum of households with fixed mass indexed by $j \in [0, 1]$ who have preferences represented by the expected utility function

$$\mathbb{E}_0 \int_0^\infty e^{-\rho t} \frac{c_{jt}^{1-\theta}}{1-\theta} dt,$$

where ρ is the rate of time preference and θ is the coefficient of relative risk aversion. At each instant t, a household's idiosyncratic labor productivity is $z_{jt} \in \{z_L, z_H\}$ with $z_L < z_H$. Households switch between the two values for labor productivity according to a Poisson process with arrival rates λ_L and λ_H.[14] The aggregate supply of efficiency units of labor is exogenous and constant and denoted by $\bar{N} = \int_0^1 z_{jt} dj$. A household with labor productivity z_{jt} earns labor income $w_t z_{jt}$. Markets are incomplete; households can only trade in productive capital a_{jt} subject to the borrowing constraint $a_{jt} \geq \underline{a}$.

There is a representative firm which has access to the Cobb-Douglas production function

$$Y_t = e^{Z_t} K_t^\alpha N_t^{1-\alpha},$$

where Z_t is (the logarithm of) aggregate productivity, K_t is aggregate capital and N_t is aggregate labor. The logarithm of aggregate productivity follows the Ornstein-Uhlenbeck process

$$\mathrm{d}Z_t = -\eta Z_t \mathrm{d}t + \sigma \mathrm{d}W_t, \tag{3.1}$$

where $\mathrm{d}W_t$ is the innovation to a standard Brownian motion, η is the rate of mean reversion, and σ captures the size of innovations.[15]

but with a considerably higher *bandwidth* or *density* than in the matrices generated by a continuous time model. A low bandwidth is important for efficiently solving sparse linear systems.

[14] The assumption that idiosyncratic shocks follow a Poisson process is for simplicity of exposition; the method can also handle diffusion or jump-diffusion shock processes.

[15] This process is the analog of an AR(1) process in discrete time.

Equilibrium In equilibrium, household decisions depend on individual state variables, specific to a particular household, and aggregate state variables, which are common to all households. The individual state variables are capital holdings a and the idiosyncratic labor productivity shock z. The aggregate state variables are aggregate productivity Z_t and the cross-sectional distribution of households over their individual state variables, $g_t(a,z)$.

For notational convenience, we denote the dependence of a given equilibrium object on a particular realization of the aggregate state $(g_t(a,z), Z_t)$ with a subscript t. That is, we use time-dependent notation with respect to those aggregate states. In contrast, we use recursive notation with respect to the idiosyncratic states (a,z). This notation anticipates our solution method which linearizes with respect to the aggregate states but not the idiosyncratic states.[16] An equilibrium of the model is characterized by the following equations:

$$\begin{aligned}\rho v_t(a,z) = &\max_c \; u(c) + \partial_a v_t(a,z)(w_t z + r_t a - c) \\ &+ \lambda_z(v_t(a,z') - v_t(a,z)) + \frac{1}{\mathrm{d}t}\mathbb{E}_t[\mathrm{d}v_t(a,z)],\end{aligned} \tag{3.2}$$

$$\frac{\mathrm{d}g_t(a,z)}{\mathrm{d}t} = -\partial_a[s_t(a,z)g_t(a,z)] - \lambda_z g_t(a,z) + \lambda_{z'} g_t(a,z'), \tag{3.3}$$

$$\mathrm{d}Z_t = -\eta Z_t \mathrm{d}t + \sigma \mathrm{d}W_t, \tag{3.4}$$

$$w_t = (1-\alpha)e^{Z_t}K_t^{\alpha}\bar{N}^{-\alpha}, \tag{3.5}$$

$$r_t = \alpha e^{Z_t}K_t^{\alpha-1}\bar{N}^{1-\alpha} - \delta, \tag{3.6}$$

$$K_t = \int a g_t(a,z)\mathrm{d}a\mathrm{d}z. \tag{3.7}$$

and where $s_t(a,z)$ is the optimal saving policy function corresponding to the household optimization problem (3.2).

For detailed derivations of these equations, see Achdou et al. (2015). The household's Hamilton-Jacobi-Bellman equation (3.2) is the continuous-time analog of the discrete time Bellman equation. The flow value of a household's lifetime utility is given by the sum of four terms:

[16] Appendix 3.7.1 writes the equilibrium conditions using fully recursive condition and shows how to obtain the system here by evaluating these conditions "along the characteristic" $(g_t(a,z), Z_t)$.

the flow utility of consumption, the marginal value of savings, the expected change due to idiosyncratic productivity shocks, and the expected change due to aggregate productivity shocks. Due to our use of time-dependent notation with respect to aggregate states, $\mathbb{E}_t$ denotes the conditional expectation with respect to aggregate states only.[17] The Kolmogorov Forward Equation (3.3) describes the evolution of the distribution over time. The flow change in the mass of households at a given point in the state space is determined by their savings behavior and idiosyncratic productivity shocks. Equation (3.4) describes the evolution of aggregate productivity. Finally, equations (3.5) to (3.7) define prices given the aggregate state.

We define a *steady state* as an equilibrium with constant aggregate productivity $Z_t = 0$ and a time-invariant distribution $g(a, z)$. The steady state system is given by

$$\rho v\left(a,z\right)=\max_{c}\ u\left(c\right)+\partial_{a}v\left(a,z\right)\left(wz+ra-c\right)+\lambda_{z}\left(v\left(a,z'\right)-v\left(a,z\right)\right), \tag{3.8}$$

$$0=-\partial_{a}\left[s\left(a,z\right)g\left(a,z\right)\right]-\lambda_{z}g\left(a,z\right)+\lambda_{z'}g\left(a,z'\right), \tag{3.9}$$

$$w=\left(1-\alpha\right)K^{\alpha}\bar{N}^{-\alpha}, \tag{3.10}$$

$$r=\alpha K^{\alpha-1}\bar{N}^{1-\alpha}-\delta, \tag{3.11}$$

$$K=\int ag(a,z)\mathrm{d}a\mathrm{d}z. \tag{3.12}$$

3.2.2 Linearization Procedure

Our linearization procedure consists of three broad steps. First, we solve for the steady state of the model without aggregate shocks but with idiosyncratic shocks. Second, we take a first-order Taylor expansion of the equilibrium conditions around the steady state, yielding a linear system of stochastic differential equations. Third, we solve the linear system using standard techniques. Conceptually, each of these steps is a straightforward extension of standard linearization techniques

[17] The borrowing constraint only affects (3.2) through the boundary condition $u'\left(w_t z_i + r_t \underline{a}\right) \geq \partial_a v_t\left(a,z\right)$ for $i = L, H$. We impose this condition in our numerical computations, but for the ease of exposition suppress the notation here.

to the heterogeneous agent context. However, the size of heterogeneous agent models leads to a number of computational challenges which we address.

Step 1: Approximate Steady State Because households face idiosyncratic uncertainty, the steady state value function varies over individual state variables $v(a, z)$, and there is a non-degenerate stationary distribution of households $g(a, z)$. To numerically approximate these functions we must represent them in a finite-dimensional way. We use a non-linear approximation in order to retain the rich non-linearities and heterogeneity at the individual level. In principle, any approximation method can be used in this step; we use the finite difference methods outlined in Achdou et al. (2015) because they are fast, accurate, and robust.

We approximate the value function and distribution over a discretized grid of asset holdings $\mathbf{a} = (a_1 = \underline{a}, a_2, ..., a_I)^{\mathrm{T}}$. Denote the value function and distribution along this discrete grid using the vectors $\mathbf{v} = (v(a_1, z_L), ..., v(a_I, z_H))^{\mathrm{T}}$ and $\mathbf{g} = (g(a_1, z_L), ..., g(a_I, z_H))^{\mathrm{T}}$; both $\mathbf{v}$ and $\mathbf{g}$ are of dimension $N \times 1$ where $N = 2I$ is the total number of grid points in the individual state space. We solve the steady state versions of (3.2) and (3.3) at each point on this grid, approximating the partial derivatives using finite differences. Achdou et al. (2015) show that if the finite difference approximation is chosen correctly, the discretized steady state is the solution to the following system of matrix equations:

$$\begin{aligned} \rho \mathbf{v} &= \mathbf{u}(\mathbf{v}) + \mathbf{A}(\mathbf{v}; \mathbf{p})\mathbf{v} \\ \mathbf{0} &= \mathbf{A}(\mathbf{v}; \mathbf{p})^{\mathrm{T}} \mathbf{g} \\ \mathbf{p} &= \mathbf{F}(\mathbf{g}). \end{aligned} \tag{3.13}$$

The first equation is the approximated steady state HJB equation (3.8) for each point on the discretized grid, expressed in our vector notation. The vector $\mathbf{u}(\mathbf{v})$ is the maximized utility function over the grid and the matrix multiplication $\mathbf{A}(\mathbf{v}; \mathbf{p})\mathbf{v}$ captures the remaining terms in (3.8). The second equation is the discretized version of the steady state Kolmogorov Forward equation (3.9). The transition matrix $\mathbf{A}(\mathbf{v}; \mathbf{p})$ is simply the transpose of the matrix from the discretized HJB equa-

tion because it encodes how households move around the individual state space. Finally, the third equation defines the prices $\mathbf{p} = (r, w)^{\mathrm{T}}$ as a function of aggregate capital through the distribution $\mathbf{g}$.[18]

Since $\mathbf{v}$ and $\mathbf{g}$ each have N entries, the total system has $2N+2$ equations in $2N+2$ unknowns. In simple models like this one, highly accurate solutions can be obtained with as little as $N = 200$ grid points (i.e., $I = 100$ asset grid points together with the two income states); however, in more complicated models, such as the two-asset model in Section 3.4, N can easily grow into the tens of thousands. Exploiting the sparsity of the transition matrix $\mathbf{A}(\mathbf{v}; \mathbf{p})$ is necessary to even represent the steady state of such large models.

Step 2: Linearize Equilibrium Conditions The second step of our method is to compute a first-order Taylor expansion of the model's discretized equilibrium conditions around steady state. With aggregate shocks, the discretized equilibrium is characterized by

$$\begin{aligned}
\rho \mathbf{v}_t &= \mathbf{u}(\mathbf{v}_t) + \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)\mathbf{v}_t + \frac{1}{\mathrm{d}t}\mathbb{E}_t \mathrm{d}\mathbf{v}_t \\
\frac{\mathrm{d}\mathbf{g}_t}{\mathrm{d}t} &= \mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)^{\mathrm{T}} \mathbf{g}_t \\
\mathrm{d}Z_t &= -\eta Z_t \mathrm{d}t + \sigma \mathrm{d}W_t \\
\mathbf{p}_t &= \mathbf{F}(\mathbf{g_t}; Z_t).
\end{aligned} \tag{3.14}$$

The system (3.14) is a non-linear system of $2N + 3$ stochastic differential equations in $2N + 3$ variables (the $2N + 2$ variables from the steady state, plus aggregate productivity Z_t). Shocks to TFP Z_t induce fluctuations in marginal products and therefore prices $\mathbf{p}_t = \mathbf{F}(\mathbf{g_t}; Z_t)$. Fluctuations in prices in turn induce fluctuations in households' decisions and therefore in $\mathbf{v}_t$ and the transition

[18] The fact that prices are an explicit function of the distribution is a special feature of the Krusell and Smith (1998) model. In general, market clearing conditions take the form $\mathbf{F}(\mathbf{v}, \mathbf{g}, \mathbf{p}) = \mathbf{0}$. Our solution method also handles this more general case.

matrix $\mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)$.[19] Fluctuations in the transition matrix then induce fluctuations in the distribution of households $\mathbf{g}_t$.

The key insight is that this large-dimensional system of stochastic differential equations has exactly the same structure as more standard representative agent models which are normally solved by means of linearization methods. To make this point, Appendix 3.7.2 relates the system (3.14) to the real business cycle (RBC) model. The discretized value function points $\mathbf{v}_t$ are jump variables, like aggregate consumption C_t in the RBC model. The discretized distribution $\mathbf{g}_t$ points are endogenous state variables, like aggregate capital K_t in the RBC model. TFP Z_t is an exogenous state variable. Finally, the wage and real interest rate are statically defined variables, just as in the Krusell and Smith (1998) model.

As already anticipated, we exploit this analogy and solve the non-linear system (3.14) by linearizing it around the steady state. Since the dimension of the system is large it is impossible to compute derivatives by hand. We use a recently developed technique called automatic (or algorithmic) differentiation that is fast and accurate up to machine precision. It dominates finite differences in terms of accuracy and symbolic differentiation in terms of speed. Automatic differentiation exploits the fact that the computer represents any function as the composition of various elementary functions, such as addition, multiplication, or exponentiation, which have known derivatives. It builds the derivative of the original function by iteratively applying the chain rule. This allows automatic differentiation to exploit the sparsity of the transition matrix $\mathbf{A}(\mathbf{v}_t;\mathbf{p}_t)$ when taking derivatives, which is essential for numerical feasibility in large models.[20]

[19] We have written the price vector $\mathbf{p}_t$ as a function of the state vector to easily exposit our methodology in a way that directly extends to models with more general market clearing conditions (see footnote 18). However, this approach is not necessary in the Krusell and Smith (1998) model because we can simply substitute the expression for prices directly into the households' budget constraint and hence the matrix $\mathbf{A}(\mathbf{v}_t; \mathbf{p}_t)$.

[20] To the best of our knowledge, there is no existing open-source automatic differentiation package for `Matlab` which exploits sparsity. We therefore wrote our own package for the computational toolbox.

The first-order Taylor expansion of (3.14) can be written as:[21]

$$\mathbb{E}_t \begin{bmatrix} \mathrm{d}\widehat{\mathbf{v}}_t \\ \mathrm{d}\widehat{\mathbf{g}}_t \\ \mathrm{d}Z_t \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{vv} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{vp} \\ \mathbf{B}_{gv} & \mathbf{B}_{gg} & \mathbf{0} & \mathbf{B}_{gp} \\ \mathbf{0} & \mathbf{0} & -\eta & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{pg} & \mathbf{B}_{pZ} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{v}}_t \\ \widehat{\mathbf{g}}_t \\ Z_t \\ \widehat{\mathbf{p}}_t \end{bmatrix} \mathrm{d}t \tag{3.15}$$

The variables in the system, $\widehat{\mathbf{v}}_t$, $\widehat{\mathbf{g}}_t$, Z_t and $\widehat{\mathbf{p}}_t$, are expressed as deviations from their steady state values, and the matrix is composed of the derivatives of the equilibrium conditions evaluated at steady state. Since the pricing equations are static, the fourth row of this matrix equation only has non-zero entries on the right hand side.[22] It is convenient to plug the pricing equations $\widehat{\mathbf{p}}_t = \mathbf{B}_{pg}\widehat{\mathbf{g}}_t + \mathbf{B}_{pZ}Z_t$ into the remaining equations of the system, yielding

$$\mathbb{E}_t \begin{bmatrix} \mathrm{d}\widehat{\mathbf{v}}_t \\ \mathrm{d}\widehat{\mathbf{g}}_t \\ \mathrm{d}Z_t \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{B}_{vv} & \mathbf{B}_{vp}\mathbf{B}_{pg} & \mathbf{B}_{vp}\mathbf{B}_{pZ} \\ \mathbf{B}_{gv} & \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg} & \mathbf{B}_{gp}\mathbf{B}_{pZ} \\ \mathbf{0} & \mathbf{0} & -\eta \end{bmatrix}}_{\mathbf{B}} \begin{bmatrix} \widehat{\mathbf{v}}_t \\ \widehat{\mathbf{g}}_t \\ Z_t \end{bmatrix} \mathrm{d}t. \tag{3.16}$$

Step 3: Solve Linear System The final step of our method is to solve the linear system of stochastic differential equations (3.16). Following standard practice, we perform a Schur decom-

[21] To arrive at (3.15), we first rearrange (3.14) so that all time derivatives are on the left-hand side. We then take the expectation of the entire system and use the fact that the expectation of a Brownian increment is zero $\mathbb{E}_t[\mathrm{d}W_t] = 0$ to write (3.14) compactly without the stochastic term as

$$\mathbb{E}_t \begin{bmatrix} \mathrm{d}\mathbf{v}_t \\ \mathrm{d}\mathbf{g}_t \\ \mathrm{d}Z_t \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{u}\left(\mathbf{v}_t;\mathbf{p}_t\right) + \mathbf{A}\left(\mathbf{v}_t;\mathbf{p}_t\right)\mathbf{v}_t - \rho\mathbf{v}_t \\ \mathbf{A}\left(\mathbf{v}_t;\mathbf{p}_t\right)^{\mathrm{T}}\mathbf{g}_t \\ -\eta Z_t \\ \mathbf{F}\left(\mathbf{g_t};Z_t\right) - \mathbf{p}_t \end{bmatrix} \mathrm{d}t.$$

Finally, we linearize this system to arrive at (3.15). Note that this compact notation loses the information contained in the stochastic term $\mathrm{d}W_t$. However, since we linearize the system, this without loss of generality: as we discuss later linearized systems feature certainty equivalence.

[22] The special structure of the matrix $\mathbf{B}$ involving zeros is particular to the Krusell and Smith (1998) model and can be relaxed. In addition, the fact that we can express prices as a static function of $\widehat{\mathbf{g}}_t$ and Z_t is a special feature of the model; more generally, the equilibrium prices are only defined implicitly by a set of market clearing conditions.

position of the matrix $\mathbf{B}$ to identify the stable and unstable roots of the system. If the Blanchard and Kahn (1980) condition holds, i.e., the number of stable roots equals the number of state variables $\widehat{\mathbf{g}}_t$ and Z_t, then we can compute the solution:

$$\begin{aligned}
\widehat{\mathbf{v}}_t &= \mathbf{D}_{vg}\widehat{\mathbf{g}}_t + \mathbf{D}_{vZ}Z_t, \\
\frac{\mathrm{d}\widehat{\mathbf{g}}_t}{\mathrm{d}t} &= (\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg} + \mathbf{B}_{gv}\mathbf{D}_{vg})\widehat{\mathbf{g}}_t + (\mathbf{B}_{gp}\mathbf{B}_{pZ} + \mathbf{B}_{gv}\mathbf{D}_{vZ})Z_t, \\
\mathrm{d}Z_t &= -\eta Z_t \mathrm{d}t + \sigma \mathrm{d}W_t, \\
\widehat{\mathbf{p}}_t &= \mathbf{B}_{pg}\widehat{\mathbf{g}}_t + \mathbf{B}_{pZ}Z_t.
\end{aligned} \tag{3.17}$$

The first line of (3.17) sets the control variables $\widehat{\mathbf{v}}_t$ as functions of the state variables $\widehat{\mathbf{g}}_t$ and Z_t, i.e. the matrices $\mathbf{D}_{vg}$ and $\mathbf{D}_{vZ}$ characterize the optimal decision rules as a function of aggregate states. The second line plugs that solution into the system (3.16) to compute the evolution of the distribution. The third line is the stochastic process for the aggregate productivity shock and the fourth line is the definition of prices $\widehat{\mathbf{p}}_t$.

3.2.3 What Does Linearization Capture and What Does It Lose?

Our method uses a mix of nonlinear approximation with respect to individual state variables and linear approximation with respect to aggregate state variables. Concretely, from the first line of (3.17), the approximated solution for the value function is of the form

$$v_t(a_i, z_j) = v(a_i, z_j) + \sum_{k=1}^{I}\sum_{\ell=1}^{2} \mathbf{D}_{vg}[i, j; k, l](g_t(a_k, z_\ell) - g(a_k, z_\ell)) + \mathbf{D}_{vZ}[i, j]Z_t, \tag{3.18}$$

where $\mathbf{D}_{vg}[i, j; k, l]$ and $\mathbf{D}_{vZ}[i, j]$ denote the relevant elements of $\mathbf{D}_{vg}$ and $\mathbf{D}_{vZ}$, and $v(a, z)$ and $g(a, z)$ are the steady state value function and distribution. Given the value function $v_t(a_i, z_j)$, optimal consumption at different points of the income and wealth distribution is then given by

$$c_t(a_i, z_j) = (\partial_a v_t(a_i, z_j))^{-1/\theta}. \tag{3.19}$$

Certainty Equivalence Expressions (3.18) and (3.19) show that our solution features *certainty equivalence* with respect to aggregate shocks; the standard deviation σ of aggregate TFP Z_t does not enter households' decision rules.[23] This is a generic feature of all linearization techniques.

However, our solution does *not* feature certainty equivalence with respect to idiosyncratic shocks, because the distribution of idiosyncratic shocks enters the HJB equation (3.2) as well as its linearized counterpart in (3.16) directly. A corollary of this is that our method *does* capture the effect of aggregate uncertainty to the extent that aggregate shocks affect the distribution of idiosyncratic shocks. For example, Bloom et al. (2014) study the effect of "uncertainty shocks" that result in an increase in the dispersion of idiosyncratic shocks and can be captured by our method.[24]

Our solution method may instead be less suitable for various asset-pricing applications in which the direct effect of aggregate uncertainty on individual decision rules is key. In future work we hope to encompass such applications by extending our first-order perturbation method to higher orders, or by allowing the decision rules to depend non-linearly on relevant low-dimensional aggregate state variables (but not the high-dimensional distribution).

Distributional Dependence of Aggregates A common motivation for studying heterogeneous agent models is that the response of macroeconomic aggregates to aggregate shocks may depend on the distribution of idiosyncratic states. For example, different joint distributions of income and wealth $g(a, z)$ can result in different impulse responses of aggregates to the same aggregate shock. Our solution method preserves such *distributional dependence*.

To fix ideas, consider the impulse response of aggregate consumption C_t to a productivity shock Z_t, starting from the steady-state distribution $g(a, z)$. First consider the response of initial aggregate consumption C_0 only. We compute the impact effect of the shock on the initial value function $v_0(a, z)$ and initial consumption $c_0(a, z)$ from (3.18) and (3.19). Integrate this over households to

[23] Note that σ does not enter the matrix $\mathbf{B}$ characterizing the linearized system (3.16) and therefore also does not enter the matrices characterizing the optimal decision rules $\mathbf{D}_{vg}$ and $\mathbf{D}_{vZ}$.

[24] McKay (2017) studies how time-varying idiosyncratic uncertainty on aggregate consumption dynamics. Terry (2017) studies how well discrete-time relatives of our method capture time-variation in the dispersion of productivity shocks in a heterogeneous firm model.

get aggregate consumption

$$C_0 = \int c_0(a,z)g(a,z)dadz \approx \sum_{i=1}^{I}\sum_{j=1}^{2} c_0(a_i,z_j)g(a_i,z_j)\Delta a\Delta z.$$

The impulse response of C_0 depends on the initial distribution $g_0(a,z)$ because the elasticities of individual consumption $c_0(a,z)$ with respect to the aggregate shock Z_0 are different for individuals with different levels of income and/or wealth. These individual elasticities are then aggregated according to the initial distribution. Therefore, the effect of the shock is *distributional-dependent* through the initial distribution $g_0(a,z)$.

To see this even more clearly, it is useful to briefly work with the continuous rather than discretized value and consumption policy functions. Analogous to (3.18), we can write the initial value function response as $\widehat{v}_0(a,z) = \mathbf{D}_{vZ}(a,z)Z_0$ where $\mathbf{D}_{vZ}(a,z)$ are the elements of $\mathbf{D}_{vZ}$ in (3.17) and where we have used that the initial distribution does not move $\widehat{g}_0(a,z) = 0$ by virtue of being a state variable. We can use this to show that the deviation of initial consumption from steady state satisfies $\widehat{c}_0(a,z) = \mathbf{D}_{cZ}(a,z)Z_0$ where $\mathbf{D}_{cZ}(a,z)$ captures the responsiveness of consumption to the aggregate shock.[25] The impulse response of initial aggregate consumption is then

$$\widehat{C}_0 = \int \mathbf{D}_{cZ}(a,z)g(a,z)dadz \times Z_0. \tag{3.20}$$

It depends on the steady-state distribution $g(a,z)$ since the responsiveness of individual consumption to the aggregate shock $\mathbf{D}_{cZ}(a,z)$ differs across (a,z).

Size- and Sign-Dependence Another question of interest is whether our economy features size- or sign-dependence, that is, whether it responds non-linearly to aggregate shocks of different sizes or asymmetrically to positive and negative shocks.[26] In contrast to state dependence, our lineariza-

[25] In particular $\mathbf{D}_{cZ}(a,z) = (\partial_a v(a,z))^{-\frac{1}{\theta}-1}\partial_a \mathbf{D}_{vZ}(a,z)$. To see this note that $\widehat{c}_0(a,z) = (\partial_a v(a,z))^{-\frac{1}{\theta}-1}\partial_a \widehat{v}_0(a,z) = (\partial_a v(a,z))^{-\frac{1}{\theta}-1}\partial_a \mathbf{D}_{vZ}(a,z)Z_0 := \mathbf{D}_{cZ}(a,z)Z_0$.

[26] Note that this is separate from the state dependence we just discussed which is concerned with how the distribution may affect the *linear* dynamics of the system.

tion method eliminates any potential sign- and size-dependence. This can again be seen clearly from the impulse response of initial aggregate consumption in (3.20) which is linear in the aggregate shock Z_0. This immediately rules out size- and sign-dependence in the response of aggregate consumption to the aggregate shock.[27]

In future work we hope to make progress on relaxing this feature of our solution method. Extending our first-order perturbation method to higher orders would again help in this regard. Another idea is to leverage the linear model solution together with parts of the full non-linear model to simulate the model in a way that preserves these nonlinearities. In particular one could use the fully nonlinear Kolmogorov Forward equation in (3.14) instead of the linearized version in (3.16) to solve for the path of the distribution for times $t > 0$: $d\mathbf{g}_t/dt = \mathbf{A}\left(\mathbf{v}_t;\mathbf{p}_t\right)^{\mathrm{T}}\mathbf{g}_t$. This procedure allows us to preserve *size-dependence* after the initial impact $t > 0$ because larger shocks potentially induce non-proportional movements in the individual state space, and therefore different distributional dynamics going forward.[28]

Small versus Large Aggregate Shocks Another generic feature of linearization techniques is that the linearized solution is expected to be a good approximation to the true non-linear solution for small aggregate shocks and less so for large ones. Section 3.2.4 below documents that our approximate dynamics of the distribution is accurate for the typical calibration of TFP shocks in the Krusell and Smith (1998) model, but breaks down for very large shocks.

3.2.4 Performance of Linearization in Krusell-Smith Model

In order to compare the performance of our method to previous work, we solve the model under the parameterization of the JEDC comparison project Den Haan, Judd and Julliard (2010). A unit

[27] Note that expression (3.20) only holds at $t = 0$. At times $t > 0$, the distribution also moves $\widehat{g}_t(a,z) \neq 0$. The generalization of (3.20) to $t > 0$ is $\widehat{C}_t \approx \int \widehat{c}_t(a,z)g(a,z)dadz + \int c(a,z)\widehat{g}_t(a,z)dadz$. Since both $\widehat{c}_t(a,z)$ and $\widehat{g}_t(a,z)$ will be linear in Z_t so will be $\widehat{C}_t$, again ruling out size- and sign-dependence.

[28] An open question is under what conditions this procedure would be consistent with our use of linear approximations to solve the model. One possible scenario is as follows: even though the time path for the distribution differs substantially computed using the non-linear Kolmogorov Forward equation, the time path for prices may still be well approximated by the linearized solution. Hence, the error in the HJB equation from using the linearized prices may be small.

of time is one quarter. We set the rate of time preference $\rho = 0.01$ and the coefficient of relative risk aversion $\theta = 1$. Capital depreciates at rate $\delta = 0.025$ per quarter and the capital share is $\alpha = 0.36$. We set the levels of idiosyncratic labor productivity z_L and z_H following Den Haan, Judd and Julliard (2010).

One difference between our model and Den Haan, Judd and Julliard (2010) is that we assume aggregate productivity follows the continuous-time, continuous-state Ornstein-Uhlenbeck process (3.1) rather than the discrete-time, two-state Markov chain in Den Haan, Judd and Julliard (2010). To remain as consistent with Den Haan, Judd and Julliard (2010)'s calibration as possible, we choose the approximate quarterly persistence $\text{corr}(\log Z_{t+1}, \log Z_t) = e^{-\eta} \approx 1 - \eta = 0.75$ and the volatility of innovations $\sigma = 0.007$ to match the standard deviation and autocorrelation of Den Haan, Judd and Julliard (2010)'s two-state process.[29]

In our approximation we set the size of the individual asset grid $I = 100$, ranging from $a_1 = 0$ to $a_I = 100$. Together with the two values for idiosyncratic productivity, the total number of grids is $N = 200$ and the total size of the dynamic system (3.16) is 400.[30]

Table 3.1 shows that our linearization method solves the Krusell and Smith (1998) model in approximately one quarter of one second. In contrast, the fastest algorithm documented in the comparison projection by Den Haan (2010) takes over seven minutes to solve the model – more than 1500 times slower than our method (see Table 2 in Den Haan (2010)).[31] In Section 3.3 we solve the model in approximately 0.1 seconds using our model-free reduction method.

Accuracy of Linearization The key restriction that our method imposes is linearity with respect to the aggregate state variables Z_t and $\widehat{\mathbf{g}}_t$. We evaluate the accuracy of this approximation using the error metric suggested by Den Haan (2010). The Den Haan error metric compares the dynamics of the aggregate capital stock under two simulations of the model for $T = 10,000$ periods. The

[29] Another difference is that Den Haan, Judd and Julliard (2010) allows the process for idiosyncratic shocks to depend on the aggregate state. We set our idiosyncratic shock process to match the average transition probabilities in Den Haan, Judd and Julliard (2010).

[30] In this calculation, we have dropped one grid point from the distribution using the restriction that the distribution integrates to one. Hence there are $N = 200$ equations for $\widehat{\mathbf{v}}_t$, $N - 1 = 199$ equations for $\widehat{\mathbf{g}}_t$ and one equation for Z_t.

[31] As discussed by Den Haan (2010), there is one algorithm (Penal) that "is even faster, but this algorithm does not solve the actual [Krusell-Smith] model specified."

Table 3.1: Run Time for Solving Krusell-Smith Model

	Full Model
Steady State	0.082 sec
Derivatives	0.021 sec
Linear system	0.14 sec
Simulate IRF	0.024 sec
Total	**0.27 sec**

Notes: Time to solve Krusell-Smith model once on `MacBook Pro 2016` laptop with `3.3 GHz` processor and `16 GB RAM`, using `Matlab R2016b` and our code toolbox. "Steady state" reports time to compute steady state. "Derivatives" reports time to compute derivatives of discretized equilibrium conditions. "Linear system" reports time to solve system of linear differential equations. "Simulate IRF" reports time to simulate impulse responses reported in Figure 3.1. "Total" is the sum of all these tasks.

Table 3.2: Maximum den Haan Error in $\%$

St. Dev Productivity Shocks ($\%$)	**Maximum den Haan Error** ($\%$)
0.01	0.000
0.1	0.001
0.7	**0.049**
1.0	0.118
5.0	3.282

Notes: Maximum percentage error in accuracy check suggested by Den Haan (2010). The error is the percentage difference between the time series of aggregate capital under our linearized solution and a nonlinear simulation of the model, as described in the main text. The bold face row denotes the calibrated value $\sigma = 0.007$.

first simulation computes the path of aggregate capital K_t from our linearized solution (3.17). The second simulation computes the path of aggregate capital K_t^* from simulating the model using the nonlinear dynamics (3.3) as discussed in Section 3.2.3. We then compare the maximum log difference between the two series,

$$\epsilon^{\text{DH}} = 100 \times \max_{t \in [0,T]} |\log K_t - \log K_t^*| .$$

Den Haan originally proposed this metric to compute the accuracy of the forecasting rule in the Krusell and Smith (1998) algorithm; in our method, the linearized dynamics of the distribution $\mathbf{g}_t$ are analogous to the forecasting rule.

Our method gives a maximum percentage error $\epsilon^{\text{DH}} = 0.049\%$, implying that households in our model make small errors in forecasting the distribution. Our method is three times as accurate as the Krusell and Smith (1998) method, which is the most accurate algorithm in Den Haan (2010) and gives $\epsilon^{\text{DH}} = 0.16\%$. Table 3.2 shows that, since our method is locally accurate, its accuracy decreases in the size of the shocks σ. However, because aggregate shocks are small empirically, it provides exceptional accuracy in our calibration.

3.3 Model Reduction

Solving the system (3.16) is feasible (and in fact extremely fast) because the Krusell and Smith (1998) model is relatively small. However, in larger models like the two-asset model in Section 3.4, the required matrix decomposition becomes prohibitively expensive. In order to solve these more general models we must therefore reduce the size of the system. Furthermore, even in smaller models like Krusell and Smith (1998), model reduction makes estimation feasible by reducing the size of the associated filtering problem.[32]

[32]Mongey and Williams (2016) use a discrete-time relative of our method without model reduction to estimate a small heterogeneous firm model. Winberry (2016) provides an alternative parametric approach for reducing the distribution and also uses it to estimate a small heterogeneous firm model.

We develop a model-free reduction method to reduce the size of the system while preserving accuracy. Under our approach we project the high-dimensional distribution $\widehat{\mathbf{g}}_t$ and value function $\widehat{\mathbf{v}}_t$ onto low-dimensional subspaces and solve the resulting low-dimensional linear system. This section describes how we choose these projections and characterize their dynamics. The main challenge is reducing the distribution, which we discuss in Sections 3.3.1 to 3.3.3. Section 3.3.4 describes how we reduce the value function. Section 3.3.5 puts the two together to solve the reduced model and describes the numerical implementation. Section 3.3.6 shows that our reduction method performs well in the Krusell and Smith (1998) model.

To simplify notation, in the remainder of Section 3.3, we denote by $\mathbf{v}_t, \mathbf{g}_t$ and $\mathbf{p}_t$ *deviations from steady state* of the value function, distribution and prices, i.e. we "drop the hats" from $\widehat{\mathbf{v}}_t, \widehat{\mathbf{g}}_t$ and $\widehat{\mathbf{p}}_t$. Note that this change of notation applies to the present section only. We will remind the reader whenever this could cause confusion.

3.3.1 Overview of Distribution Reduction

The basic insight we exploit is that only a small subset of the information in $\mathbf{g}_t$ is necessary to accurately forecast the path of prices $\mathbf{p}_t$. In fact, in the discrete time version of this model, Krusell and Smith (1998) show that just the mean of the asset distribution $\mathbf{g}_t$ is sufficient to forecast $\mathbf{p}_t$ according to a forecast-error metric. However, the success of their reduction strategy relies on the economic properties of the model, so it is not obvious how to generalize it to other environments. We use a set of tools from the engineering literature known as *model reduction* to generalize Krusell and Smith (1998)'s insight in a model-free way, allowing the computer to compute the features of the distribution necessary to accurately forecast $\mathbf{p}_t$.[33]

[33]The following material is based on lecture notes by Amsallem and Farhat (2011), which in turn build on a book by Antoulas (2005). Lectures 3 and 7 by Amsallem and Farhat (2011) and Chapters 1 and 11 in Antoulas (2005) are particularly relevant. All lecture notes for Amsallem and Farhat (2011) are available online at `https://web.stanford.edu/group/frg/course_work/CME345/` and the book by Antoulas (2005) is available for free at `http://epubs.siam.org/doi/book/10.1137/1.9780898718713`. Also see Reiter (2010) who applies related ideas from the model reduction literature in order to reduce the dimensionality of a linearized discrete-time heterogeneous agent model.

It is important to note that the vector $\mathbf{p}_t$ does not need to literally consist of prices; it is simply the vector of objects we wish to accurately describe. In practice, we often also include other variables of interest, such as aggregate consumption or output, to ensure that the reduced model accurately describes their dynamics as well.

We say that the distribution *exactly reduces* if there exists a $k_{\mathcal{S}}$-dimensional time-invariant subspace $\mathcal{S}$ with $k_{\mathcal{S}} << N$ such that, for all distributions $\mathbf{g}_t$ that occur in equilibrium,

$$\mathbf{g}_t = \gamma_{1t}\mathbf{x}_1 + \gamma_{2t}\mathbf{x}_2 + ... + \gamma_{k_{\mathcal{S}}t}\mathbf{x}_{k_{\mathcal{S}}},$$

where $\mathbf{X}_{\mathcal{S}} = [\mathbf{x}_1, ..., \mathbf{x}_{k_{\mathcal{S}}}] \in \mathbb{R}^{N \times k_{\mathcal{S}}}$ is a basis for the subspace $\mathcal{S}$ and $\gamma_{1t}, ..., \gamma_{k_{\mathcal{S}}t}$ are scalars. If we knew the time-invariant basis $\mathbf{X}_{\mathcal{S}}$, this would decrease the dimensionality of the problem because the distribution would be characterized by the $k_{\mathcal{S}}-$dimensional vector of coefficients γ_t.

Typically exact reduction as described above will not hold, so we instead must estimate a *trial basis* $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_k] \in \mathbb{R}^{N \times k}$ such that the distribution *approximately reduces*, i.e.,

$$\mathbf{g}_t \approx \gamma_{1t}\mathbf{x}_1 + \gamma_{2t}\mathbf{x}_2 + ... + \gamma_{kt}\mathbf{x}_k,$$

or, in matrix form, $\mathbf{g}_t \approx \mathbf{X}\gamma_t$. We denote the resulting approximation of the distribution by $\widetilde{\mathbf{g}}_t = \mathbf{X}\gamma_t$ and the approximate prices by $\widetilde{\mathbf{p}}_t = \mathbf{B}_{pg}\widetilde{\mathbf{g}}_t + \mathbf{B}_{pZ}Z_t$. Starting with a large system in terms of the distribution $\mathbf{g}_t$, the goal is to obtain a smaller system in terms of the coefficients γ_t that implies behavior of $\widetilde{\mathbf{p}}_t$ that is "close" to the behavior of $\mathbf{p}_t$ in the true non-reduced model. This discussion also makes clear that model reduction involves two related tasks. First, given a trial basis $\mathbf{X}$, we must compute the coefficients γ_t. Second, we must choose the basis $\mathbf{X}$ itself. We describe these steps below.

In a special case our problem maps exactly into the prototypical problem in the model reduction literature. This special case assumes away a crucial part of the economics we are interested in studying but is nevertheless useful for understanding the connection to that literature. In particular, suppose that individuals' *decision rules were exogenous*, i.e. that the matrices $\mathbf{D}_{vg}$ and

$\mathbf{D}_{vZ}$ in (3.17) were exogenously given.[34] In that case our dynamical system would become purely backward looking. In particular, from the second and fourth equations of (3.17) and recalling our convention in this section to drop hats from variables

$$\begin{aligned} \frac{\mathrm{d}\mathbf{g}_t}{\mathrm{d}t} &= \mathbf{C}_{gg}\mathbf{g}_t + \mathbf{C}_{gZ}Z_t \\ \mathbf{p}_t &= \mathbf{B}_{pg}\mathbf{g}_t + \mathbf{B}_{pZ}Z_t, \end{aligned} \tag{3.21}$$

where $\mathbf{C}_{gg} = \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg} + \mathbf{B}_{gv}\mathbf{D}_{vg}$ and $\mathbf{C}_{gZ} = \mathbf{B}_{gp}\mathbf{B}_{pZ} + \mathbf{B}_{gv}\mathbf{D}_{vZ}$. This backward-looking system has exactly the same form as the problems studied in the model reduction literature: there is a low-dimensional vector of inputs (here aggregate productivity Z_t) that affects a high-dimensional vector of endogenous state variables (here the distribution $\mathbf{g}_t$) that in turn determines a low-dimensional vector of output variables (here prices $\mathbf{p}_t$).[35] That is, the system maps a low-dimensional object Z_t into another low-dimensional object $\mathbf{p}_t$. The problem is that the "intermediating" variable $\mathbf{g}_t$ is high-dimensional. The goal of the model reduction literature is to replace $\mathbf{g}_t$ with a lower-dimensional object γ_t while retaining a good approximation of the output variable $\mathbf{p}_t$. With exogenous decision rules our task would be easy because the model reduction literature provides a whole battery of tools for doing exactly this. The difficulty is that decision rules are, of course, not exogenously given and therefore the system of interest is not purely backward-looking.

Instead, our system of interest (3.16) involves an equation for the value function $\mathbf{v}_t$. For some of our analysis, it is helpful to restate the system in a form that is closer to the notation used in the

[34] Exogenous decision rules more typically relate the value function to prices, i.e. $\mathbf{v}_t = \mathbf{D}_{vp}\mathbf{p}_t$. But prices $\mathbf{p}_t = \mathbf{B}_{pg}\mathbf{g}_t + \mathbf{B}_{pZ}Z_t$ in turn depend on the distribution $\mathbf{g}_t$ and productivity Z_t. Hence so do the decision rules: $\mathbf{v}_t = \mathbf{D}_{vg}\mathbf{g}_t + \mathbf{D}_{vZ}Z_t$ with $\mathbf{D}_{vg} = \mathbf{D}_{vp}\mathbf{B}_{pg}$ and $\mathbf{D}_{vZ} = \mathbf{D}_{vp}\mathbf{D}_{pZ}$.

[35] The system (3.21) is called a *linear time invariant (LTI) system*. Equivalently, instead of assuming that decision rules are exogenous, we could have assumed that there is no feedback from individuals' decisions to the distribution $\mathbf{B}_{gv} = 0$. In that case the system (3.16) again becomes a backward-looking system of the LTI form (3.21), now with $\mathbf{C}_{gg} = \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg}$ and $\mathbf{C}_{gZ} = \mathbf{B}_{gp}\mathbf{B}_{pZ}$.

model reduction literature as

$$\begin{bmatrix} \mathbb{E}_t[\mathrm{d}\mathbf{v}_t] \\ \mathrm{d}\mathbf{g}_t \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{vv} & \mathbf{B}_{vp}\mathbf{B}_{pg} \\ \mathbf{B}_{gv} & \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg} \end{bmatrix} \begin{bmatrix} \mathbf{v}_t \\ \mathbf{g}_t \end{bmatrix} \mathrm{d}t + \begin{bmatrix} \mathbf{B}_{vp}\mathbf{B}_{pZ} \\ \mathbf{B}_{gp}\mathbf{B}_{pZ} \end{bmatrix} Z_t \mathrm{d}t, \tag{3.22}$$
$$\mathbf{p}_t = \mathbf{B}_{pg}\mathbf{g}_t + \mathbf{B}_{pZ}Z_t.$$

and given the exogenous stochastic process for productivity (3.4). This way of restating the system makes clear the connection to the prototypical problem in the model reduction literature (3.21): the system still maps a low-dimensional input Z_t into a low-dimensional output $\mathbf{p}_t$. The difference is that there are now an additional intermediating variable, the forward-looking $\mathbf{v}_t$.

3.3.2 Distribution Reduction in Simplified Model

Before tackling the distribution reduction in the full linearized model (3.22), we first consider a simplified version. We make three simplifying assumptions. All three are made for purely expositional and pedagogical reasons and we relax them in Section 3.3.3. First, we assume that $Z_t = 0$ for all t, i.e. there is no aggregate uncertainty. This implies that we can drop the equation expectation operator in (3.22). Second, we assume that $\mathbf{p}_t = p_t$ is a scalar. This emphasizes that the price vector we are trying to approximate is a low-dimensional object. Third, we assume that there is no feedback from the price to the evolution of the distribution, so $\mathbf{B}_{gp} = 0$. Under these assumptions, we obtain the following simplified version of the system (3.22)

$$\begin{bmatrix} \dot{\mathbf{v}}_t \\ \dot{\mathbf{g}}_t \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{vv} & \mathbf{b}_{vp}\mathbf{b}_{pg} \\ \mathbf{B}_{gv} & \mathbf{B}_{gg} \end{bmatrix} \begin{bmatrix} \mathbf{v}_t \\ \mathbf{g}_t \end{bmatrix}, \tag{3.23}$$
$$p_t = \mathbf{b}_{pg}\mathbf{g}_t,$$

where $\mathbf{b}_{vp}$ is a $N \times 1$ vector and $\mathbf{b}_{pg}$ is a $1 \times N$ vector. This simplified formulation of the problem focuses on the role of the distribution in determining the current price and, through the evolution of the distribution over time, the path of future prices. In Section 3.3.3 we will return to reducing

the distribution in the full model using the insights developed from studying the simplified system (3.23).

Given the absence of aggregate shocks, $Z_t = 0$, we are concerned with deterministic transition dynamics from an exogenously given initial distribution $\mathbf{g}_0$ that differs from the steady state distribution. Starting from $\mathbf{g}_0$, the distribution $\mathbf{g}_t$ then converges to the steady state deterministically. In later sections, we will instead compute impulse responses driven by shocks to Z_t but starting from steady state. Because certainty equivalence with respect to aggregate shocks holds in the linearized models, the two analyses are closely related and the deterministic transition dynamics also characterize the impulse response dynamics following an aggregate shock.

Given the connection to the model reduction literature discussed in Section 3.3.1, we first explain our approach for reducing the distribution under the assumption that decision rules are exogenous (Sections 3.3.2 and 3.3.2). We then explain how to relax this assumption in Section 3.3.2. Under the assumption that decision rules are exogenous, we can substitute out the value function $\mathbf{v}_t = \mathbf{D}_{vg}\mathbf{g}_t$ in (3.23) and obtain the following purely backward-looking system:

$$\begin{aligned} \dot{\mathbf{g}}_t &= \mathbf{C}_{gg}\mathbf{g}_t, \quad \mathbf{C}_{gg} := \mathbf{B}_{gg} + \mathbf{B}_{gv}\mathbf{D}_{vg} \\ p_t &= \mathbf{b}_{pg}\mathbf{g}_t. \end{aligned} \tag{3.24}$$

Note that the system is also a special case of (3.21). As already noted, model reduction involves two related tasks. First, given a trial basis $\mathbf{X}$, we must compute the coefficients γ_t. Second, we must choose the basis $\mathbf{X}$ itself. We describe these steps in the next two subsections.

Computing System in Terms of Coefficients γ_t Given Basis $\mathbf{X}$

Mathematically, we project the distribution $\mathbf{g}_t$ onto the subspace spanned by the basis $\mathbf{X} \in \mathbb{R}^{N \times k}$. In particular we can write the requirement that $\mathbf{g}_t \approx \mathbf{X}\gamma_t$ as

$$\mathbf{g}_t = \mathbf{X}\gamma_t + \varepsilon_t, \tag{3.25}$$

where $\varepsilon_t \in \mathbb{R}^N$ is a residual. The formulation (3.25) is a standard linear regression in which the distribution $\mathbf{g}_t$ is the dependent variable, the basis vectors $\mathbf{X}$ are the independent variables, and the coefficients γ_t are to be estimated.

Just as in ordinary least squares, we can estimate the projection coefficients γ_t by imposing the orthogonality condition $\mathbf{X}^{\mathrm{T}}\varepsilon_t = 0$, giving the familiar formula

$$\gamma_t = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{g}_t. \tag{3.26}$$

A sensible basis will be orthonormal, so that $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1} = \mathbf{I}$, further simplifying (3.26) to $\gamma_t = \mathbf{X}^{\mathrm{T}}\mathbf{g}_t$.[36] We can compute the evolution of this coefficient vector by differentiating (3.26) with respect to time to get

$$\dot{\gamma}_t = \mathbf{X}^{\mathrm{T}}\dot{\mathbf{g}}_t = \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}(\mathbf{X}\gamma_t + \varepsilon_t) \approx \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\gamma_t$$

The hope is that the residuals ε_t are small and so the last approximation is good. Assuming this is the case, we have the reduced analogue of (3.24)

$$\begin{aligned} \dot{\gamma}_t &= \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\gamma_t \\ \widetilde{p}_t &= \mathbf{b}_{pg}\mathbf{X}\gamma_t. \end{aligned} \tag{3.27}$$

Summing up, assuming we have the basis $\mathbf{X}$, this projection procedure takes us from the system of differential equations involving the N-dimensional vector $\mathbf{g}_t$ in (3.24) to a system involving only the k-dimensional vector γ_t in (3.27). We now turn to choosing a good basis $\mathbf{X}$.

[36] The model reduction literature also presents alternatives to our "least squares" approach to computing the coefficients γ_t. In particular, one can also estimate γ_t using what amounts to an instrumental variables strategy: one can define a second subspace spanned by the columns of some matrix $\mathbf{Z}$ and impose the orthogonality condition $\mathbf{Z}^{\mathrm{T}}\varepsilon_t = 0$. This yields an alternative estimate $\gamma_t = (\mathbf{Z}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{Z}^{\mathrm{T}}\mathbf{g}_t$. Mathematically, this is called an *oblique projection* (as opposed to an orthogonal projection) of $\mathbf{g}_t$ onto the k-dimensional subspace spanned by the columns $\mathbf{X}$ along the kernel of $\mathbf{Z}^{\mathrm{T}}$. See Amsallem and Farhat (2011, Lecture 3) and Antoulas (2005) for more detail on oblique projections.

Computing Basis X

Mechanically increasing the size of the basis $\mathbf{X}$ will improve the approximation of the distribution $\mathbf{g}_t$; in the limit where $\mathbf{X}$ spans $\mathbb{R}^N$, we will not reduce the distribution at all. The challenge in the model reduction literature is to compute an accurate approximation of the distribution $\mathbf{g}_t$ with as small a basis $\mathbf{X}$ as possible.

We choose the basis $\mathbf{X}$ so that the dynamics of the approximate prices $\widetilde{p}_t$ in (3.27) match those of the true prices p_t in (3.24) as accurately as possible.[37] In continuous time, it is natural to operationalize the notion "accurately matching the path of prices p_t" by matching the Taylor series approximation of p_{t+s} around p_t:

$$p_{t+s} \approx p_t + \dot{p}_t s + \frac{1}{2}\ddot{p}_t s^2 + ... + \frac{1}{(k-1)!}p_t^{(k-1)} s^{k-1}, \tag{3.28}$$

where $\dot{p}_t$, $\ddot{p}_t$, and $p_t^{(k-1)}$ denote time derivatives. From (3.24) we can write these time derivatives as

$$\begin{aligned}
\dot{p}_t &= \mathbf{b}_{pg}\dot{\mathbf{g}}_t = \mathbf{b}_{pg}\mathbf{C}_{gg}\mathbf{g}_t \\
\ddot{p}_t &= \mathbf{b}_{pg}\mathbf{C}_{gg}^2\mathbf{g}_t \\
&\vdots \\
p_t^{(k-1)} &= \mathbf{b}_{pg}\mathbf{C}_{gg}^{k-1}\mathbf{g}_t,
\end{aligned}$$

[37] Our approach for choosing the basis $\mathbf{X}$ is a simplified version of what the model reduction literature calls "moment matching." See Amsallem and Farhat (2011, Lecture 7) and Antoulas (2005, Chapter 11). It is also the continuous-time analogue of what Reiter (2010) terms "conditional expectation approach" (see his Section 3.2.2).

or, in vector form,

$$\mathcal{P}_t := \begin{bmatrix} p_t \\ \dot{p}_t \\ \ddot{p}_t \\ \vdots \\ p_t^{(k-1)} \end{bmatrix} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\mathbf{g}_t, \quad \text{where} \quad \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg}) := \begin{bmatrix} \mathbf{b}_{pg} \\ \mathbf{b}_{pg}\mathbf{C}_{gg} \\ \mathbf{b}_{pg}\mathbf{C}_{gg}^2 \\ \vdots \\ \mathbf{b}_{pg}\mathbf{C}_{gg}^{k-1} \end{bmatrix}. \tag{3.29}$$

The $k \times N$-dimensional matrix $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$ is known as the *observability matrix* of the system (3.24).[38] Using this notation, we can write the kth order Taylor expansion of p_{t+s} as

$$p_{t+s} \approx \left[1, s, \tfrac{1}{2}s^2, ..., \tfrac{1}{(k-1)!}s^{k-1}\right] \mathcal{P}_t.$$

The key insight is that choosing the transpose of the observability matrix $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$ as the basis allows us to exactly match the vector of derivatives $\mathcal{P}_t$ and, therefore, the kth-order Taylor expansion of p_{t+s} even though we know only the reduced state vector γ_t. That is, with $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})^{\mathrm{T}}$, the kth-order Taylor series expansion of approximate prices $\widetilde{p}_{t+s}$ exactly equals that of p_{t+s}. To see this, denote by $\widetilde{\mathcal{P}}_t$ the vector of time derivatives of the approximate prices $\widetilde{p}_t$ which satisfies $\widetilde{\mathcal{P}}_t = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\mathbf{X}\gamma_t$. Next, substitute $\mathbf{g}_t = \mathbf{X}\gamma_t + \varepsilon_t$ into (3.29) to get

$$\mathcal{P}_t = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\mathbf{X}\gamma_t + \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\varepsilon_t,$$

where again ε_t is the residual from the regression. By construction, the residuals must satisfy $\mathbf{X}^{\mathrm{T}}\varepsilon_t = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\varepsilon_t = 0$, implying that $\mathcal{P}_t = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})\mathbf{X}\gamma_t = \widetilde{\mathcal{P}}_t$.

[38] Observability of a dynamical system is an important concept in control theory introduced by Rudolf Kalman, the inventor of the Kalman filter. It is a measure of how well a system's states (here $\mathbf{g}_t$) can be inferred from knowledge of its outputs (here p_t). For systems like ours observability can be directly inferred from the observability matrix $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$ with $k = N$. Note that some texts refer only to $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$ with $k = N$ as "observability matrix" and to the matrix with $k < N$ as "partial observability matrix."

Distribution Reduction with Endogenous Decision Rules

So far we have assumed that the decision rules $\mathbf{D}_{vg}$ were exogenous. This counterfactual assumption allowed us to substitute out the value function $\mathbf{v}_t = \mathbf{D}_{vg}\mathbf{g}_t$ and to apply results from the model reduction literature without modification. Of course, in our economic applications, the decision rules are generally endogenous and we need to solve the model precisely in order to find these decision rules.

We continue to work with our simplified problem from before, but now with the version (3.23) that includes the equation for the value function $\mathbf{v}_t$. This makes explicit that the forward-looking decision rules have yet to be found. Apart from this change, our distribution reduction procedure follows exactly the same steps as above. First as in Section 3.3.2, given a trial basis $\mathbf{X}$, we derive a system in terms of the coefficients γ_t which is now given by

$$\begin{aligned} \begin{bmatrix} \dot{\mathbf{v}}_t \\ \dot{\gamma}_t \end{bmatrix} &= \begin{bmatrix} \mathbf{B}_{vv} & \mathbf{b}_{vp}\mathbf{b}_{pg}\mathbf{X} \\ \mathbf{X}^{\mathrm{T}}\mathbf{B}_{gv} & \mathbf{X}^{\mathrm{T}}\mathbf{B}_{gg}\mathbf{X} \end{bmatrix} \begin{bmatrix} \mathbf{v}_t \\ \gamma_t \end{bmatrix} \\ \widetilde{p}_t &= \mathbf{b}_{pg}\mathbf{X}\gamma_t. \end{aligned} \tag{3.30}$$

As before, projecting onto the subspace spanned by $\mathbf{X}$ takes us from the system of differential equations involving the N-dimensional vector $\mathbf{g}_t$ in (3.23) to a system involving only the k-dimensional vector γ_t. This smaller system can then be solved using the same techniques described in Section 3.2. In particular, for any basis $\mathbf{X}$ we can compute approximate decision rules $\widetilde{\mathbf{v}}_t = \mathbf{D}_{v\gamma}\gamma_t$.

It remains to choose a good trial basis $\mathbf{X}$. In Section 3.3.2 we have shown that if decision rules $\mathbf{D}_{vg}$ were exogenous, then the efficient choice of basis would be $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})^{\mathrm{T}}$ with $\mathbf{C}_{gg} = \mathbf{B}_{gg} + \mathbf{B}_{gv}\mathbf{D}_{vg}$. However, this choice of basis was only dictated by the concern of *efficiently* approximating the distribution with as small a basis as possible; it is always possible to increase *accuracy* by adding additional orthogonal basis vectors. Therefore, setting $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{B}_{gg})^{\mathrm{T}}$, i.e. ignoring feedback from individuals' decisions to the distribution by effectively setting $\mathbf{B}_{gv} = 0$, will not be efficient but may still be accurate. In practice, we have found in both the Krusell and

Smith (1998) and the two-asset model in Section 3.4 that this choice leads to accurate solutions for high enough order k of the observability matrix.

In cases where choosing $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{B}_{gg})^{\mathrm{T}}$ is not accurate even for as high an order k as numerically feasible, we suggest an iterative procedure. First, we solve the reduced model (3.30) based on the inaccurate basis choice $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{B}_{gg})^{\mathrm{T}}$. This yields decision rules $\mathbf{D}_{v\gamma}$ defining a mapping from the reduced distribution γ_t to the value function. We then use these to construct an approximation to the true decision rules $\mathbf{D}_{vg}$ (which map the full distribution to the value function), namely $\widetilde{\mathbf{D}}_{vg} = \mathbf{D}_{v\gamma}\mathbf{X}^{\mathrm{T}}$.[39] We then set $\mathbf{X} = \mathcal{O}(\mathbf{b}_{pg}, \mathbf{B}_{gg} + \mathbf{B}_{gv}\widetilde{\mathbf{D}}_{vg})^{\mathrm{T}}$ and solve the model again based on the new reduction. If the second reduction gives an accurate solution, we are done; if not, we continue the iteration. Although we have no theoretical guarantee that this iteration will converge, in practice we have found that it does.

Choosing k and Internal Consistency with Endogenous Decision Rules

A key practical step in reducing the distribution is choosing the order of the observability matrix k, which determines the size of the basis $\mathbf{X}$. For backward-looking systems like (3.24), we showed that a basis of order k implies that the path of reduced prices $\widetilde{p}_t$ matches the k-th order Taylor expansion of the path of true prices p_t, providing a natural metric for assessing accuracy.[40] However, this logic does not carry through to forward-looking systems like (3.23), leaving unclear what exactly a basis of order k captures.

In the finite limit when $k = N$, any linearly independent basis spans all of $\mathbb{R}^N$ so the distribution is not reduced at all and the reduced model is vacuously accurate. Hence, a natural procedure is to increase k until the dynamics of reduced prices converge. In practice, this convergence is often monotonic. However, we cannot prove convergence is always monotonic, leaving open the question of what exactly the reduced model captures for a given order k.

[39] Recall from (3.26) that the projection of $\mathbf{g}_t$ onto $\mathbf{X}$ defines the reduced distribution as $\gamma_t = \mathbf{X}^{\mathrm{T}}\mathbf{g}_t$. Hence the optimal decision rule can be written as $\widetilde{\mathbf{v}}_t = \mathbf{D}_{v\gamma}\gamma_t = \mathbf{D}_{v\gamma}\mathbf{X}^{\mathrm{T}}\mathbf{g}_t = \widetilde{\mathbf{D}}_{vg}\mathbf{g}_t$ where $\widetilde{\mathbf{D}}_{vg} = \mathbf{D}_{v\gamma}\mathbf{X}^{\mathrm{T}}$.

[40] Recall that in general p_t includes both prices and other observables of interest to the researcher.

We suggest an *internal consistency* metric to assess the extent to which the reduced model satisfies the model's equilibrium conditions. The spirit of our internal consistency check is similar to Krusell and Smith (1998)'s R^2 forecast-error metric and Den Haan (2010)'s accuracy measure discussed in Section 3.2.4: if agents make decisions based on the price path implied by the reduced distribution, but we aggregate those decisions against the true full distribution, do the prices generated by the true distribution match the forecasts?

Concretely, our internal consistency check consists of three steps. First, we compute households' decisions based on the reduced distribution, $\widetilde{\mathbf{v}}_t = \mathbf{D}_{v\gamma}\gamma_t$. Second, we use these decisions to compute the nonlinear dynamics of the full distribution $\mathbf{g}_t^*$ – not the reduced version γ_t – and its implied prices p_t^*

$$\begin{aligned} p_t^* &= \mathbf{b}_{pg}\mathbf{g}_t^* \\ \dot{\mathbf{g}}_t^* &= \mathbf{A}(\widetilde{\mathbf{v}}_t, p_t^*)\mathbf{g}_t^*, \end{aligned}$$

where $\mathbf{A}(\widetilde{\mathbf{v}}_t, p_t^*)$ is the nonlinear transition matrix implied by the decision rules $\widetilde{\mathbf{v}}_t$ and price p_t^*. The third step of our internal accuracy check is to assess the extent to which the dynamics of p_t^* matches the dynamics implied by the reduced system $\widetilde{p}_t$. If the two paths are close, households in the reduced model could not significantly improve their forecasts by using additional information about the distribution. Once again, we compare the maximum log deviation of the two paths

$$\epsilon = \max_{t \geq 0} \left| \log \widetilde{p}_t - \log p_t^* \right|.$$

3.3.3 Distribution Reduction in Full Model

There are three main complications to reducing the distribution in the full model (3.22) relative to the simplified system (3.23) we analyzed in Section 3.3.2. First, there are aggregate productivity shocks. Second, the price vector $\mathbf{p}_t$ is in general $\ell \times 1$ with $\ell \geq 1$. Third, there is feedback from prices to the evolution of the distribution, so $\mathbf{B}_{gp} \neq 0$.

Introducing these complications to the distribution reduction muddies notation but does not alter the logic of Section 3.3.2. We therefore apply exactly the same steps to the full linearized model (3.22) as we did to the simplified version (3.23). We again first fix a basis $\mathbf{X}$ and derive a law of motion for the coefficient vector γ_t (the reduced distribution), and then choose a basis $\mathbf{X}$ that is constructed to perform well in forecasting the evolution of future prices. Following the logic of section 3.3.2, the dynamic system after reducing the distribution is

$$\begin{aligned}\begin{bmatrix}\mathbb{E}_t[\mathrm{d}\mathbf{v}_t]\\ \mathrm{d}\gamma_t\end{bmatrix} &= \begin{bmatrix}\mathbf{B}_{vv} & \mathbf{B}_{vp}\mathbf{B}_{pg}\mathbf{X}\\ \mathbf{X}^{\mathrm{T}}\mathbf{B}_{gv} & \mathbf{X}^{\mathrm{T}}(\mathbf{B}_{gg}+\mathbf{B}_{gp}\mathbf{B}_{pg})\mathbf{X}\end{bmatrix}\begin{bmatrix}\mathbf{v}_t\\ \gamma_t\end{bmatrix}\mathrm{d}t + \begin{bmatrix}\mathbf{B}_{vp}\mathbf{B}_{pZ}\\ \mathbf{X}^{\mathrm{T}}\mathbf{B}_{gp}\mathbf{B}_{pZ}\end{bmatrix}Z_t\mathrm{d}t, \\ \widetilde{\mathbf{p}}_t &= \mathbf{B}_{pg}\mathbf{X}\gamma_t + \mathbf{B}_{pZ}Z_t.\end{aligned} \tag{3.31}$$

Our goal is to choose $\mathbf{X}$ such that the time path of approximate prices $\widetilde{\mathbf{p}}_t$ generated by this system is close to the vector of true prices $\mathbf{p}_t$ in (3.22) *for any realization of the stochastic process* Z_t. Our starting point is again the case where there is no feedback from individuals' choices to the distribution $\mathbf{B}_{gv} = 0$. As discussed in Section 3.3.2 this can potentially be improved upon by means of an iterative strategy that iterates on the approximate decision rule $\widetilde{\mathbf{D}}_{vg}$.

Owing to the presence of aggregate shocks Z_t, the full derivation showing how to choose $\mathbf{X}$ is somewhat more involved and can be found in Appendix 3.7.3. The logic is, however, the same as in Section 3.3.2. The vector of time derivatives of prices $\mathcal{P}_t$ can again be written in terms of an observability matrix $\mathcal{O}$ that has the same structure as (3.29) but with $\mathbf{b}_{pg}\mathbf{C}_{gg}$ replaced by $\mathbf{B}_{pg}(\mathbf{B}_{gg}+\mathbf{B}_{gp}\mathbf{B}_{pg})$.[41] If there are ℓ prices and we want to match a kth order Taylor-series expansion of $\mathbf{p}_{t+s}$, we need to match $k_g = \ell \times k$ time derivatives and hence the observability matrix $\mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg}+\mathbf{B}_{gp}\mathbf{B}_{gp})$ is now of dimension $k_g \times N$. Therefore, we choose $\mathbf{X} = \mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg}+\mathbf{B}_{gp}\mathbf{B}_{gp})^{\mathrm{T}}$ as the basis for our projection method. Finally note that our strategy for reducing the distribution kept the endogenous state variable $\mathbf{g}_t$ separate from the exogenous state variable Z_t, and only reduced the dimensionality of the the former. This is because the two types of state

[41] In the Taylor-series expansion there are now additional terms involving Z_t – hence the complication.

variables play fundamentally different roles: the productivity process Z_t is an exogenous input into the system whereas the distribution $\mathbf{g}_t$ is internal to the system.

Approximate Multicollinearity Our choice of basis $\mathbf{X}$ is numerically unstable due to approximate multicollinearity; as in standard regression, high degree standard polynomials are nearly collinear due to the fact that, for large k, $\mathbf{B}_{pg}(\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{k-2} \approx \mathbf{B}_{pg}(\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{k-1}$, leaving the necessary projection of the distribution onto $\mathbf{X}$ numerically intractable.

We overcome this challenge by relying on a *Krylov subspace method*, an equivalent but more numerically stable class of methods.[42] For any $N \times N$ matrix $\mathbf{A}$ and $N \times 1$ vector $\mathbf{b}$, the order-k Krylov subspace is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{b}) = \text{span}\left(\left\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, ..., \mathbf{A}^{k-1}\mathbf{b}\right\}\right).$$

From this definition it can be seen that the subspace spanned by the columns of $\mathbf{X} = \mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{\mathrm{T}}$ is simply the order-k Krylov subspace generated by $(\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{\mathrm{T}}$ and $\mathbf{B}_{pg}^{\mathrm{T}}$, i.e. $\mathcal{K}_k(\mathbf{B}_{gg}^{\mathrm{T}} + \mathbf{B}_{gp}^{\mathrm{T}}\mathbf{B}_{pg}^{\mathrm{T}}, \mathbf{B}_{pg}^{\mathrm{T}})$. Therefore, the projection of $\mathbf{g}_t$ on $\mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{\mathrm{T}}$ is equivalent to the projection of $\mathbf{g}_t$ onto this Krylov subspace.

There are many methods for computing the bases of Krylov subspaces in the literature. One important feature of all these methods is that they take advantage of the sparsity of the underlying matrices.[43] We have found that one particular method, *deflated block Arnoldi iteration*, is a robust procedure. Deflated block Arnoldi iteration has two advantages for our application. First, it is a stable procedure to orthogonalize the columns of the basis $\mathbf{X}$ and eliminate the approximate multicollinearity. Second, the *deflation* component handles multicollinearity that can arise even with non-deflated block Arnoldi iteration.

[42] See Antoulas (2005, Chapter 11) and Amsallem and Farhat (2011, Lecture 7).

[43] Even though $\mathbf{B}_{gg}$ is sparse and $\mathbf{B}_{gp}$ and $\mathbf{B}_{pg}$ are only $\ell \times N$, the matrix $\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg}$ which actually enters the system (3.31) is $N \times N$ and not sparse (because $\mathbf{B}_{gp}\mathbf{B}_{pg}$ is $N \times N$ not sparse). In the two-asset model in Section 3.4, $N = 66,000$, and even storing this matrix is not feasible. Fortunately it is never actually necessary to compute this full matrix; instead, it is only necessary to compute $\mathbf{B}_{pg}(\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})$ which involves the action of $\mathbf{B}_{gp}\mathbf{B}_{pg}$ on a thin $\ell \times N$ matrix $\mathbf{B}_{pg}$ and can be computed as $(\mathbf{B}_{pg}\mathbf{B}_{gp})\mathbf{B}_{pg}$.

3.3.4 Value Function Reduction

After reducing the dimensionality of the distribution $\mathbf{g}_t$, we are left with a system of dimension $N + k_g$ with $k_g << N$ (recall $k_g = \ell \times k$ where ℓ is the number of prices and k is the order of the approximation according to which the basis $\mathbf{X}$ is chosen). Although this is considerably smaller than the original system which was of size $2N$, it is still large because it contains N equations for the value function at each point along the individual state space. In complex models, this leaves the linear system too large for matrix decomposition methods to be feasible.[44]

We therefore also reduce the dimensionality of the distribution $\mathbf{v}_t$. Just like in our method for reducing the distribution $\mathbf{g}_t$, we project the (deviation from steady state of the) value function $\mathbf{v}_t$ onto a lower-dimensional subspace. As before, an important question is how to choose the basis for this projection. We choose it by appealing to the theory for approximating smooth functions and approximate $\mathbf{v}_t$ using splines. In most models, the value function is sufficiently smooth that a low-dimensional spline provides an accurate approximation. In particular, any spline approximation can be written as the projection

$$\mathbf{v}_t \approx \mathbf{X}_v \nu_t,$$

where $\mathbf{X}_v$ is a $N \times k_v$ matrix defining the spline knot points and ν_t are the k_v coefficients at those knot points.[45] Given this linear projection the coefficients are given by $\nu_t = (\mathbf{X}_v^{\mathrm{T}}\mathbf{X}_v)^{-1}\mathbf{X}_v^{\mathrm{T}}\mathbf{v}_t = \mathbf{X}_v^{\mathrm{T}}\mathbf{v}_t$, where we have used that we typically choose an orthonormal $\mathbf{X}_v$ so that $\mathbf{X}_v^{\mathrm{T}}\mathbf{X}_v = \mathbf{I}$.

It is worth emphasizing the symmetry with our distribution reduction method, the projection (3.25). In order to do so we add a g-subscript to the basis in the distribution reduction for the remainder of the paper and write (3.25) as $\mathbf{g}_t \approx \mathbf{X}_g \gamma_t$. Hence from now on $\mathbf{X}_g$ denotes the basis in the reduction of the distribution $\mathbf{g}_t$ and $\mathbf{X}_v$ denotes the basis in the reduction of the value function $\mathbf{v}_t$. It is also important to note that we are approximating the deviation of the value function from its steady state value, not the value function itself (the reader should recall our convention in the

[44] One way to overcome this challenge is to use sparse matrix methods to find just the k eigenvalues associated with the stable eigenvectors. This is much faster than computing the full matrix decomposition necessary to obtain the full set of eigenvectors. However, it is slower than the approach we pursue in this subsection.

[45] Note that, in general, the number of coefficients is different from the number of knot points.

present section to drop hat subscripts from variables that are in deviation from steady state for notational simplicity).

We have found that non-uniformly spaced quadratic splines work well for three reasons. First, the non-uniform spacing can be used to place more knots in regions of the state space with high curvature, allowing for an efficient dimensionality reduction. Second, the quadratic spline preserves monotonicity and concavity between knot points, which is important in computing first-order conditions. Third, and related, the local nature of splines implies that they avoid creating spurious oscillations at the edges of the state space (Runge's phenomenon) which often occurs with global approximations like high-degree polynomials.

It is important to note the difference between approximating the deviations of the value function from steady state using quadratic splines – which we do – versus solving for the steady state value using quadratic splines – which we do not do. The finite difference method we use to compute the steady state does not impose that the value function is everywhere differentiable, which is potentially important for capturing the effects of non-convexities. However, after having computed the steady state value functions, it is typically the case that they have kinks at a finite number of points and are well-approximated by smooth functions between these points. It is then straightforward to fit quadratic splines between the points of non-differentiability.

3.3.5 Putting It All Together: A Numerical Toolbox

Summarizing the previous sections, we have projected the distribution $\mathbf{g}_t$ onto the subspace spanned by $\mathbf{X}_g$ and the value function $\mathbf{v}_t$ onto the subspace spanned by $\mathbf{X}_v$. Now we simply need to keep track of the $k_v \times 1$ coefficient vector ν_t for the value function and the $k_g \times 1$ coefficient vector γ_t for the distribution. Because knowledge of these coefficients is sufficient to reconstruct the full value function and distribution, we will also sometimes refer to ν_t as the reduced value

function and to γ_t as the reduced distribution. Our original system (3.16) is now reduced to

$$\mathbb{E}_t \begin{bmatrix} \mathrm{d}\nu_t \\ \mathrm{d}\gamma_t \\ \mathrm{d}Z_t \end{bmatrix} = \begin{bmatrix} \mathbf{X}_v^{\mathrm{T}} \mathbf{B}_{vv} \mathbf{X}_v & \mathbf{X}_v^{\mathrm{T}} \mathbf{B}_{vp} \mathbf{B}_{pg} \mathbf{X}_g & \mathbf{X}_v^{\mathrm{T}} \mathbf{B}_{vp} \mathbf{B}_{pZ} \\ \mathbf{X}_g^{\mathrm{T}} \mathbf{B}_{gv} \mathbf{X}_v & \mathbf{X}_g^{\mathrm{T}} (\mathbf{B}_{gg} + \mathbf{B}_{gp} \mathbf{B}_{pg}) \mathbf{X}_g & \mathbf{X}_g^{\mathrm{T}} \mathbf{B}_{gp} \mathbf{B}_{pZ} \\ \mathbf{0} & \mathbf{0} & -\eta \end{bmatrix} \begin{bmatrix} \nu_t \\ \gamma_t \\ Z_t \end{bmatrix} \mathrm{d}t. \tag{3.32}$$

We have provided a numerical toolbox implementing the key steps in our computational method at the `github` page associated with this project.[46] Broadly, the user provides two files: one which solves for the steady state and another which evaluates the model's equilibrium conditions. Our toolbox then implements the following algorithm (we here revert back to denoting deviations from steady state with hat superscripts):

1. Compute the steady state values of $\mathbf{v}$, $\mathbf{g}$ and $\mathbf{p}$.

2. Compute a first-order Taylor expansion of the equilibrium conditions (3.14) around steady state using automatic differentiation, yielding the system (3.16) in terms of deviations from steady state $\widehat{\mathbf{v}}_t, \widehat{\mathbf{g}}_t, \widehat{\mathbf{p}}_t$ and Z_t.

3. If necessary, reduce the model, yielding the system (3.32) in terms of (ν_t, γ_t, Z_t).

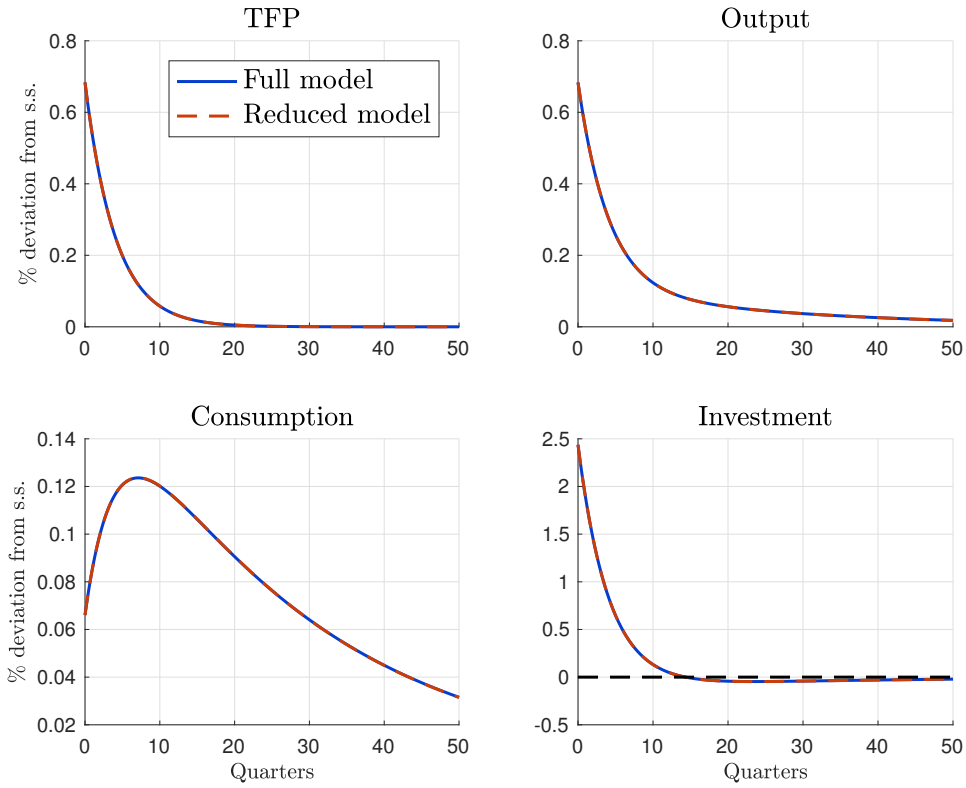
   (a) Distribution reduction: compute the basis $\mathbf{X}_g = \mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{\mathrm{T}}$ using deflated Arnoldi iteration and project $\widehat{\mathbf{g}}_t$ on $\mathbf{X}_g$ to obtain the reduced distribution γ_t.

   (b) Value function reduction: compute the spline basis $\mathbf{X}_v$ and project $\widehat{\mathbf{v}}_t$ on $\mathbf{X}_v$ to obtain the reduced value function ν_t.

4. Solve the system (3.16) or, if reduced, (3.32).

5. Simulate the system to compute impulse responses and time-series statistics.

Figure 3.1: Impulse Responses to TFP Shock in Krusell-Smith Model



Notes: We simulate the model by discretizing the time dimension with step size $\mathrm{d}t = 0.1$. We define an impulse response as the response of the economy to a one standard deviation innovation to the TFP process over an instant of time $\mathrm{d}t$. "Full model" refers to model solved without model reduction and "reduced model" with reduction, using $k_g = 5$ (forecasting $\ell = 5$ objects with a $k = 1$-order Taylor series approximation) and $k_v = 24$.

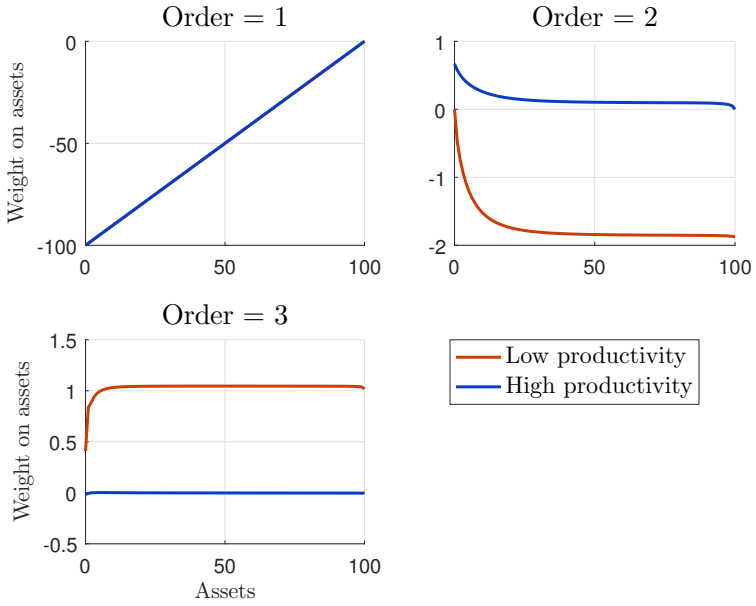
3.3.6 Model Reduction in Krusell-Smith Model

The Krusell and Smith (1998) is a useful environment for evaluating our model reduction methodology because it is possible to solve the full unreduced model as a benchmark. We are able to substantially reduce the size of the system: projecting the distribution on an observability matrix of order $k = 1$ and approximating the value function at 24 spline knot points provides an extremely accurate approximation of the model's dynamics.[47] Figure 3.1 shows that the impulse responses

[46] Currently at: `https://github.com/gregkaplan/phact`.

[47] More precisely, we choose the observability matrix so as to forecast $\ell = 5$ equilibrium objects (namely the wage and the interest rate, plus the three equilibrium aggregates we are most interested in: aggregate output, consumption, investment) to order $k = 1$ resulting in a reduced distribution γ_t of dimension $k_g = \ell \times k = 5$, and we approximate the

Figure 3.2: Basis Vectors in Distribution Reduction



Notes: The columns of $\mathbf{X}_g = \mathcal{O}(\mathbf{B}_{pg}, \mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{B}_{pg})^{\mathrm{T}}$ up to order $k = 4$. These correspond to the basis vectors in the approximated distribution $\mathbf{g}_t \approx \gamma_{1t}\mathbf{x}_{g,1} + ... + \gamma_{4t}\mathbf{x}_{g,4}$.

of key aggregate variables in the reduced model are almost exactly identical to the full, unreduced model, despite approximating the $N = 400$ dimensional dynamic system with a 30-dimensional system.[48]

The fact that we can reduce the distribution with an order observability matrix of order $k = 1$ is consistent with Krusell and Smith (1998)'s finding of "approximate aggregation" using a computationally distinct procedure and accuracy measure. In fact, as Figure 3.2 shows, a $k = 1$ order approximation of the distribution returns precisely the mean. The top left panel of the figure plots the basis vector associated with $k = 1$, split into two 100-dimensional vectors corresponding to the two values for idiosyncratic productivity. It shows that indeed the first basis vector $\mathbf{x}_{g,1} = \left[\begin{smallmatrix}\mathbf{a}\\ \mathbf{a}\end{smallmatrix}\right]$, implying that $\gamma_t = \mathbf{x}_{g,1}^{\mathrm{T}}\mathbf{g}_t = \left[\begin{smallmatrix}\mathbf{a}\\ \mathbf{a}\end{smallmatrix}\right]^{\mathrm{T}}\mathbf{g}_t = \widehat{K}_t$, the (deviation from steady state of the) mean of the distribution. The remaining panels plot the higher-order elements of $\mathbf{X}_g$, which quickly converge

value function at 12 spline knot points in the wealth dimension resulting in a reduced value function ν_t of dimension $k_v = 2 \times 12 = 24$.

[48]There are $k_v = 12 \times 2 = 24$ points for the value function, $k_g = k \times \ell = 1 \times 5 = 5$ points for the distribution because we are tracking five elements of the $\mathbf{p}_t$ vector, and 1 point for TFP Z_t.

Table 3.3: Run Time for Solving Krusell-Smith Model

	Full Model	**Reduced Model**
Steady State	0.082 sec	0.082 sec
Derivatives	0.021 sec	0.021 sec
Dim reduction	$\times$	0.007 sec
Linear system	0.14 sec	0.002 sec
Simulate IRF	0.024 sec	0.003 sec
Total	**0.267 sec**	**0.116 sec**

Notes: Time to solve Krusell-Smith model once on `MacBook Pro 2016` laptop with `3.3 GHz` processor and `16 GB RAM`, using `Matlab R2016b` and our code toolbox. "Full model" refers to solving model without model reduction and "reduced model" with reduction, using $k_g = 1$ and $k_v = 12$. "Steady state" reports time to compute steady state. "Derivatives" reports time to compute derivatives of discretized equilibrium conditions. "Dim reduction" reports time to compute both the distribution and value function reduction. "Linear system" reports time to solve system of linear differential equations. "Simulate IRF" reports time to simulate impulse responses reported in Figure 3.1. "Total" is the sum of all these tasks.

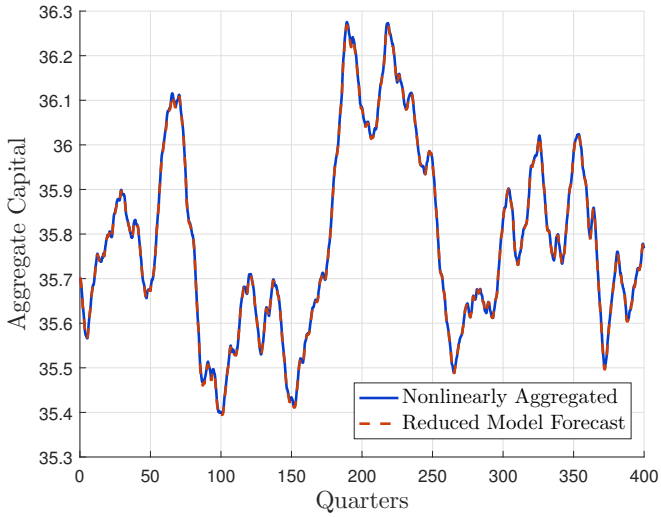
to constants that do not add information to the approximation. Hence, our model-free reduction method confirms Krusell and Smith (1998)'s approximate aggregation result in this simple model.

With or without dimensionality reduction, our method solves and simulates the model in less than 0.3 seconds. Table 3.3 reports the running time of using our `Matlab` code suite on a desktop PC. Although reduction is not necessary to solve this simple model, it nevertheless reduces running time by more than 50% and takes approximately 0.1 seconds.[49] In the two-asset model in Section 3.4, model reduction is necessary to even solve the model.

Our internal consistency check confirms the fact that the distribution reduction is accurate; the maximum log deviation is 0.065%, which is twice as small as the most accurate algorithm in the JEDC comparison Den Haan (2010). Recall that in the unreduced model that the maximum log deviation is $0.049\$$, capturing the error due to linearization. Hence, the additional error due to

[49]Recall that the fastest algorithm in the JEDC comparison Den Haan (2010) is more 7 minutes, or 3500 times longer.

Figure 3.3: Internal Consistency Check



Notes: Two series for aggregate capital that enter the internal consistency check ϵ. "Reduced model forecast" computes the path $\widetilde{K}_t$ implied by the reduced linear model. "Nonlinear model forecast" computes the path K_t^* from updating the distribution according to the nonlinear KFE (3.3).

our model reduction is extremely small. Figure 3.3 plots the two series for a random 400-quarter period of simulation and shows that the two series are extremely close to each other.[50]

3.4 Aggregate Consumption Dynamics in Two-Asset Model

While the Krusell and Smith (1998) model is a useful pedagogical tool for explaining our computational method, it performs poorly in reproducing key features of the distribution of household-level wealth and consumption, such as marginal propensities to consume. We now apply our method to solve a two-asset heterogeneous agent model inspired by Kaplan and Violante (2014) and Kaplan, Moll and Violante (2016). We parameterize the model in order to match key features of the joint distribution of income, wealth, and marginal propensities to consume – properties that one-asset models have difficulty in matching and lead to a failure of approximate aggregation. The failure of approximate aggregation, together with the model's size, render it an ideal setting to illustrate the

[50] Den Haan (2010) refers to this type of figure as the "fundamental accuracy plot."

power of our method since, to the best of our knowledge, it cannot be solved using other existing methods.

We demonstrate the usefulness of the model in two ways. First, in Section 3.4.4 we show that when the model is parameterized to match *household-level* data, it also matches key features of the joint dynamics of *aggregate* consumption and income. The model performs substantially better than a representative agent model and at least as well as a simple two-agent spender-saver model in the spirit of Campbell and Mankiw (1989), which has been reversed engineered to match the aggregate facts. We explain how differences across households in their consumption response to aggregate shocks drives these findings.

The second way we illustrate illustrate the usefulness of the two-asset model is, in Section 3.5, to demonstrate how the model can be used to understand the distributional consequences of aggregate shocks, thus paving the way for a complete analysis of the transmission of shocks to inequality. We show how these forces can be particularly large in the presence of shocks that impact some households more than others, for example when different types of workers play different roles in production.

3.4.1 Model

The household side of the model is a simplified version of Kaplan, Moll and Violante (2016), so we refer the interested reader to that paper for more details. The firm side follows the standard real business cycle model with persistent shocks to productivity growth.

Environment

Households There is a unit mass of households indexed by $j \in [0, 1]$. At each instant of time households hold liquid assets b_{jt}, illiquid assets a_{jt}, and have labor productivity z_{jt}. Households die with an exogenous Poisson intensity ζ and upon death give birth to an offspring with zero wealth $a_{jt} = b_{jt} = 0$ and labor productivity drawn from its ergodic distribution. There are perfect annuity markets, implying that the wealth of deceased households is distributed to other house-

holds in proportion to their asset holdings. Each household has preferences over consumption c_{jt} represented by the expected utility function

$$\mathbb{E}_0 \int_0^\infty e^{-(\rho+\zeta)t} \frac{c_{jt}^{1-\theta}}{1-\theta} \mathrm{d}t.$$

A household with labor productivity z_{jt} earns labor income $w_t z_{jt}$ and pays a linear income tax at rate τ. Each household also receives a lump-sum transfer from the government TQ_t, where T is a constant and Q_t is aggregate productivity (described below). Labor productivity follows a discrete state Poisson process, taking values from the set $z_{jt} \in \{z_1, ..., z_J\}$. Households switch from state z to state z' with Poisson intensity $\lambda_{zz'}$.

The liquid asset b_{jt} pays a rate of return r_t^b. Households can borrow in liquid assets up to an exogenous limit $\underline{b}Q_t$. The interest rate on borrowing is $r_t^{b-} = r_t^b + \kappa$ where $\kappa > 0$ is a wedge between borrowing and lending rates. Let $r_t^b(b_t)$ denote the interest rate function taking both of these cases into account.

The illiquid asset a_{jt} pays a rate of return r_t^a. Households cannot borrow in the illiquid asset. The asset is illiquid in the sense that a household with illiquid assets a_{it} must pay a flow cost $\chi(d_{jt}, a_{jt})Q_t$ in order to transfer assets at the rate d_{jt} between the liquid and illiquid accounts. The transaction cost function is given by[51]

$$\chi(d, a) = \chi_0 |d| + \chi_1 \left| \frac{d}{a} \right|^{\chi_2} a.$$

The role of the linear component $\chi_0 > 0$ is to generates an inaction region in households' optimal deposit policies. The role of the convex component $(\chi_1 > 0, \chi_2 > 1)$ is to ensure that deposit rates are finite, $|d_t| < \infty$ and hence household's holdings of assets never jump. Scaling the convex term by illiquid assets a delivers the desirable property that marginal costs $\chi_d(d, a)$

[51] Because the transaction cost at a = 0 is infinite, in computations we replace the term a with $\max\{a, \underline{a}\}$, where the threshold $\underline{a} > 0$ is a small value (2% of quarterly GDP, which is around $500). This guarantees that costs remain finite even for households with $a = 0$.

are homogeneous of degree zero in the deposit rate d/a so that the marginal cost of transacting depends on the fraction of illiquid assets transacted, rather than the raw size of the transaction.

The laws of motion for liquid and illiquid assets are

$$\begin{aligned}\frac{\mathrm{d}b_{jt}}{\mathrm{d}t} &= (1-\tau)w_t z_{jt} + TQ_t + r_t^b(b_{jt})b_{jt} - \chi(d_{jt}, a_{jt})Q_t - c_{jt} - d_{jt} \\ \frac{\mathrm{d}a_{jt}}{\mathrm{d}t} &= r_t^a a_{jt} + d_{jt}.\end{aligned}$$

Firm There is a representative firm with the Cobb-Douglas production function

$$Y_t = K_t^{\alpha}(Q_t\overline{L})^{1-\alpha},$$

where K_t is the aggregate capital stock, Q_t is aggregate productivity, and $\overline{L}$ is aggregate labor supply which is constant by assumption.

Unlike in previous sections we consider shocks to the growth rate of productivity $Z_t := \mathrm{d}\log Q_t$ rather than the level of productivity. Aggregate productivity growth follows the Ornstein-Uhlenbeck process

$$\begin{aligned}\mathrm{d}\log Q_t =& Z_t \mathrm{d}t \\ \mathrm{d}Z_t =& -\nu Z_t \mathrm{d}t + \sigma \mathrm{d}W_t,\end{aligned}$$

where $\mathrm{d}W_t$ is an innovation to a standard Brownian motion.

Government There is a government which balances its budget each period. Since the labor tax rate τ and lump-sum transfer rate T are constant, we assume that government spending G_t adjusts each period to satisfy the government budget constraint

$$\int_0^1 \tau w_t z_{jt} \mathrm{d}j = G_t + \int_0^1 TQ_t \mathrm{d}j. \tag{3.33}$$

Government spending G_t is not valued by households.

Asset Market Clearing The aggregate capital stock is the total amount of illiquid assets in the economy,

$$K_t = \int_0^1 a_{jt} \mathrm{d}j.$$

We assume the market for capital is competitive, so the return on the illiquid asset r_t^a is simply the rental rate of capital. The supply of liquid assets is fixed exogenously at $B_t = B^* Q_t$, where B^* is the steady state demand for liquid assets given $r_b^* = 0.005$ and $Q_t = 1$ (discussed below). For simplicity we assume that interest payments on debt come from outside the economy.

Equilibrium

We characterize the equilibrium recursively. The household-level state variables are illiquid asset holdings a, liquid asset holdings b, and labor productivity z. The aggregate state variables are aggregate productivity Q_t and the cross-sectional distribution of households over their individual state $g_t(a, b, z)$. As in Section 3.2, we denote an equilibrium object conditional on a particular realization of the aggregate state $(g_t(a, b, z), Q_t)$ with a subscript t.

Households The household's Hamilton-Jacobi-Bellman equation is given by

$$\begin{aligned}(\rho + \zeta) v_t(a, b, z) = &\max_{c,d} \frac{c^{1-\theta}}{1-\theta} + \partial_b v_t(a, b, z)(TQ_t + (1-\tau) w_t e^z + r_t^b(b) b \\ &- \chi(d, a) Qt - c - d) + \partial_a v_t(a, b, z)(r_t^a a + d) \\ &+ \sum_{z'} \lambda_{zz'}(v_t(a, b, z') - v_t(a, b, z)) + \frac{1}{\mathrm{d}t} \mathbb{E}_t[\mathrm{d}v_t(a, b, z)]. \end{aligned} \tag{3.34}$$

The cross-sectional distribution $g_t(a, b, z)$ satisfies the Kolmogorov forward equation

$$\begin{aligned}\frac{\mathrm{d}g_t(a,b,z)}{\mathrm{d}t} =& -\partial_a\left(s_t^a(a,b,z)g_t(a,b,z)\right) - \partial_b\left(s_t^b(a,b,z)g_t(a,b,z)\right) \\ & - \sum_{z'}\lambda_{zz'}g_t(a,b,z) + \sum_{z'}\lambda_{z'z}g_t(a,b,z),\end{aligned} \tag{3.35}$$

where s_t^a and s_t^b are the optimal drifts in illiquid and illiquid assets implied by (3.34).

Firms The equilibrium conditions for the production side are the firm optimality conditions, together with the process for aggregate productivity:

$$\begin{aligned} r_t^a =& \alpha K_t^{\alpha-1}(Q_t\overline{L})^{1-\alpha} - \delta \\ w_t =& (1-\alpha)K_t^{\alpha}Q_t^{1-\alpha}\overline{L}^{-\alpha} \\ \mathrm{d}\log Q_t =& Z_t\mathrm{d}t \\ \mathrm{d}Z_t =& -\nu Z_t\mathrm{d}t + \sigma\mathrm{d}W_t. \end{aligned}$$

Market Clearing Capital market clearing is given by

$$K_t = \int a\mathrm{d}a g_t(a,b,z)\mathrm{d}b\mathrm{d}z.$$

Liquid asset market clearing is given by

$$\int b g_t(a,b,z)\mathrm{d}a\mathrm{d}b\mathrm{d}z = B^*Q_t.$$

Given these conditions, together with the government budget constraint (3.33), the market for output clears by Walras' law.

Detrending Due to the non-stationary process for productivity, we work with de-trended versions of the equilibrium objects and condition. We show how to represent the economy in de-trended form in Appendix 3.7.4.

3.4.2 Calibration

We first calibrate the steady state without aggregate shocks to match key features of the cross-sectional distribution of household income and balance sheets. Our calibration closely follows Kaplan, Moll and Violante (2016). We then calibrate stochastic process for aggregate productivity to match properties of the dynamics of equilibrium output.

Exogenously Set Paramaters We choose the quarterly death rate $\zeta = 1/180$ so that households live 45 years on average. We set the tax rate $\tau = 30\%$ and set the lump sum transfer T to 10% of steady-state output. Given our labor productivity process, this policy implies that in steady state around 35% of households receive a net transfer from the government, consistent with the Congressional Budget Office (2013). We interpret borrowing in the liquid asset as unsecured credit and therefore set the borrowing limit $\underline{b}$ at one times average quarterly labor income. We set the coefficient of relative risk aversion $\theta = 1$, implying log utility.

We set the capital share in production $\alpha = 0.4$ and the annual depreciation rate on capital $\delta = 0.075$. With an equilibrium steady-state ratio of capital to annual output of 3.0 (see below) this implies an annual return on illiquid assets r^a of 5.8%. We set the steady-state annual return on liquid assets to 2%.

Labor productivity shocks We model the (logarithm of) idiosyncratic labor productivity as the sum of two independent components

$$\log z_{jt} = z_{1,jt} + z_{2,jt}. \tag{3.36}$$

Table 3.4: Targeted Labor Income Moments

Moment	Data	Model Estimated	Model Discretized
Variance: annual log earns	0.70	0.70	0.76
Variance: 1yr change	0.23	0.23	0.21
Variance: 5yr change	0.46	0.46	0.46
Kurtosis: 1yr change	17.8	16.5	17.3
Kurtosis: 5yr change	11.6	12.1	10.9
Frac 1yr change $< 10\%$	0.54	0.56	0.64
Frac 1yr change $< 20\%$	0.71	0.67	0.70
Frac 1yr change $< 50\%$	0.86	0.85	0.86

Notes: Moments of the earning process targeted in the calibration. "Data" refers to SSAA data on male earnings from Guvenen et al. (2015). "Model Estimated" refers to the continuous process (3.36) and (3.37). "Model Discretized" refers to discrete Poisson approximation of the process used in model computation.

Each process is assumed to follow the jump-drift process

$$\mathrm{d}z_{i,jt} = -\beta_i z_{i,jt}\mathrm{d}t + \mathrm{d}J_{i,jt}. \tag{3.37}$$

Jumps arrive at a Poisson arrival rate λ_i. Conditional on a jump, a new log-earnings state $z_{j,it}$ is drawn from a normal distribution with mean zero and variance σ_j^2. Between jumps, each process drifts toward zero at rate β_i.[52] Jump-drift processes of this form are closely related to discrete-time AR(1) processes, with the modification that shocks arrive at random, rather than known, dates. The parameters σ_i govern the size of the shocks, the parameters β_i govern the persistence of the shocks, and the parameters λ_i govern the frequency of arrival of shocks.

Allowing for random arrival of shocks is important both for matching observed household portfolios of liquid versus illiquid assets, and for matching the leptokurtic nature of observed annual income growth rates. If earnings shocks are transitory and frequent (high β high λ), households

[52] See Kaplan, Moll and Violante (2016) for a formal description of these process.

Table 3.5: Targeted Labor Income Moments

Parameter		Component $j = 1$	Component $j = 2$
Arrival rate	λ_j	0.080	0.007
Mean reversion	β_j	0.761	0.009
St. Deviation of innovations	σ_j	1.74	1.53

Notes: Parameters of the income process (3.36) and (3.37) estimated to match the moments in 3.4.

would accumulate a buffer stock of liquid assets to self-insure. But if earnings shocks are persistent and infrequent (low β, low λ), households would prefer to save in high-return illiquid assets and pay the transaction costs to rebalance their portfolio when shocks occur. Recent work by Guvenen et al. (2015) shows that changes in annual labor income are extremely leptokurtic, meaning that most absolute annual income changes are small but a small number are very large. We use the extent of this leptokurtosis, together with standard moments on the variance of log earnings and log earnings growth rates, to estimate the parameters of the earnings process (3.36) and (3.37). The moments we match, together with the fit of the estimated model, are shown in Table 3.4.

The estimated parameters in Table 3.5 indicate that the two jump-drift processes can be broadly interpreted as a transitory and a persistent component, in line with a long literature on estimating earnings processes in discrete time. The transitory component ($j = 1$) arrives on average once every three years and has a half-life of around one quarter. The persistent component ($j = 2$) arrives on average once every 38 years and has a half-life of around 18 years. In the context of an infinite-horizon model the persistent component can be interpreted as a "career shock." We discretize the continuous process (3.37) using 10 points for the persistent component and 3 points for the transitory component. The fit of the discretized process for the targeted moments is shown in Table 3.4.

Table 3.6: Targeted Labor Income Moments

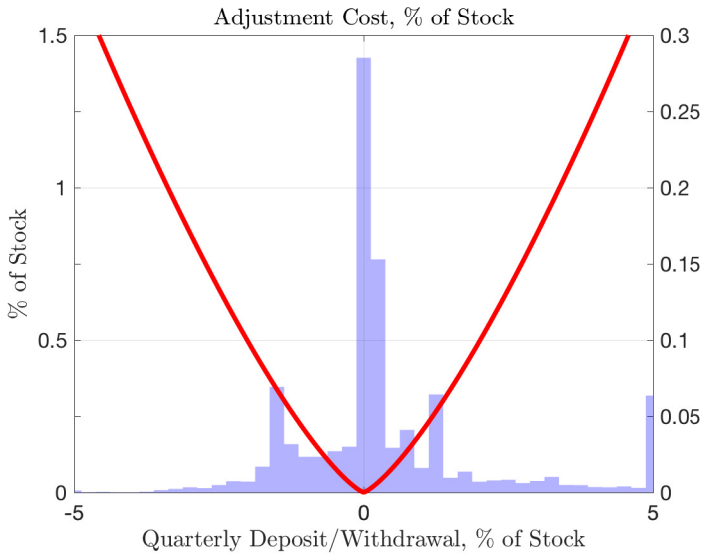
	Target	Model
Mean illiquid assets (multiple of annual GDP)	3.000	3.000
Mean liquid assets (multiple of annual GDP)	0.375	0.378
Frac. with $b = 0$ and $a = 0$	0.100	0.105
Frac. with $b = 0$ and $a > 0$	0.200	0.171
Frac. with $b < 0$	0.150	0.135

Notes: Moments of asset distribution targeted in calibration. Data source: SCF 2004.

Adjustment costs and discount factor The five remaining parameters on the household-side of the model – the discount rate ρ, the borrowing wedge κ, and the parameters of the adjustment cost function χ_0, χ_1, and χ_2 – jointly determine the incentives of households to accumulate liquid and illiquid assets. We therefore choose these parameters to match five moments of household balance sheets from the Survey of Consumer Finances 2004: the mean of the illiquid and liquid wealth distributions, the fraction of poor and wealthy hand-to-mouth households, and the fraction of households with negative assets. We choose to match mean asset holdings so that the aggregate wealth (and hence the capital stock) in the model is comparable with the US economy. We choose to match the fraction of hand-to-mouth households (poor and wealthy) since this is the most important feature of the bottom of the wealth distribution for determining marginal propensities to consume. For details on the classification of liquid and illiquid assets, see Kaplan, Moll and Violante (2016).

Table 3.6 reports the fit with respect to these moments in the calibrated model. The implied annual discount rate is 5.8% annually and the annual borrowing wedge is 8.1% annually. The equilibrium return on illiquid assets r^a is 5.8%. Figure 3.4 plots the calibrated adjustment cost function together with the distribution of quarterly deposits in the steady state. The figure shows adjustment costs as a percentage of the amount being transacted, as a function of the quarterly transaction amount (relative to holdings of illiquid assets). The transaction cost is less than 1% of

Figure 3.4: Calibrated Adjustment Cost Function

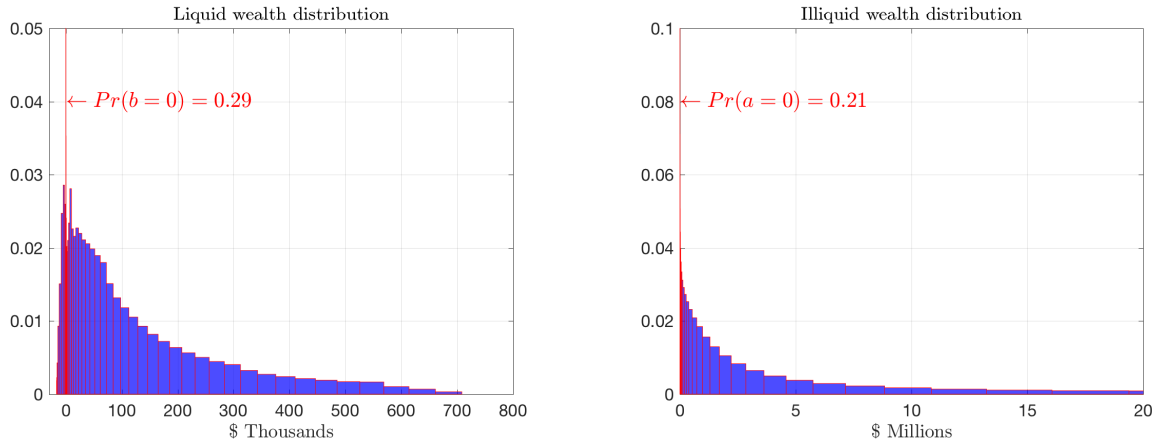


Notes: Solid line plot $\frac{\chi(d,a)}{d}$ as a function of $\frac{d}{a}$, where $\chi(d,a) = \chi_0|d_{jt}| + \chi_1 \left|\frac{d_{jt}}{a_{jt}}\right|^{\chi_2} a_{jt}$. Histogram displays the steady state distribution of adjustments $\frac{d}{a}$.

the transaction for small transactions, and rises to around 10% of the transaction for a quarterly transaction that is 2% of illiquid assets. The function has a kink at $d = 0$, which generates the mass of households who neither deposit nor withdraw in a given quarter.

Wealth Distribution The distributions of liquid and illiquid wealth are displayed in Figure 3.5. Both distributions are extremely skewed, as in the data. Roughly 28% of households have zero liquid wealth (i.e. are hand-to-mouth), a further 14% have negative liquid wealth and around 21% have positive liquid wealth less than $\$2,000$. The Gini coefficient for liquid wealth is 0.87 and the top 10 percent share is 0.85. Of the hand-to-mouth households, roughly two-thirds (17%) are wealthy hand-to-mouth, meaning that they have positive illiquid assets while one-third (11%) are poor hand-to-mouth, meaning that they have zero liquid and zero illiquid assets. Around 10% of household have positive liquid assets but no illiquid assets, and around 43% have have positive illiquid assets less than $\$10,000$. There is a thick right tail of the illiquid wealth distribution: 3%

Figure 3.5: Liquid and Illiquid Wealth Distribution in Steady State



(a) Liquid assets b

(b) Illiquid assets a

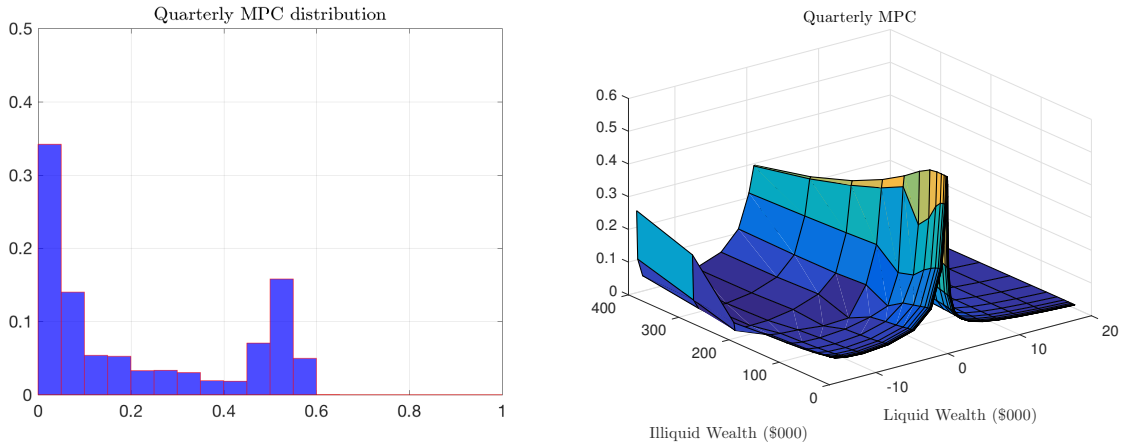
Notes: Steady state distributions of liquid and illiquid wealth.

of households have more than $\$2,000,000$ in illiquid assets and the top 10 percent hold 92% of total illiquid wealth in the economy.

Marginal Propensities to Consume The substantial number of hand-to-mouth households in the model means that the model generates a distribution of MPCs that is in line with empirical evidence. The average quarterly spending out of a \$500 cash windfall is 23%, in the range of the empirical estimates in Johnson, Parker and Souleles (2006) and Parker et al. (2013). The large average consumption response is comprised of a small response for households with positive liquid wealth and a large response (around 0.5) for hand-to-mouth households. This bi-modality can be seen clearly in the left panel of Figure 3.6, and is consistent with recent work by Fagereng, Holm and Natvik (2016).[53] The right panel of 3.6 displays the corresponding response at each point in the space of liquid and illiquid assets (averaged across labor productivity). The figure shows clearly that only households with zero (or very negative) liquid wealth have substantial MPCs, and this is true even for households with illiquid assets.

[53]Fagereng, Holm and Natvik (2016) study consumption responses to lottery winnings using Norwegian administrative data. They find that MPCs are high for households with nearly zero liquid assets, even if the household has positive illiquid assets.

Figure 3.6: Heterogeneity in MPCs Across Households



(a) Distribution in Steady State

(b) MPC Function

Notes: Quarterly MPCs out of a \$500 windfall in steady state. The MPC over a period τ is $MPC_\tau(a,b,z) = \frac{\partial C_\tau(a,b,z)}{\partial b}$, where $C_\tau(a,b,z) = \mathbb{E}\left[\int_0^\tau c(a_t,b_t,z_t)\mathrm{d}t | a_0 = a, b_0 = b, z_0 = z\right]$.

Table 3.7: Targeted Moments of Real GDP Growth

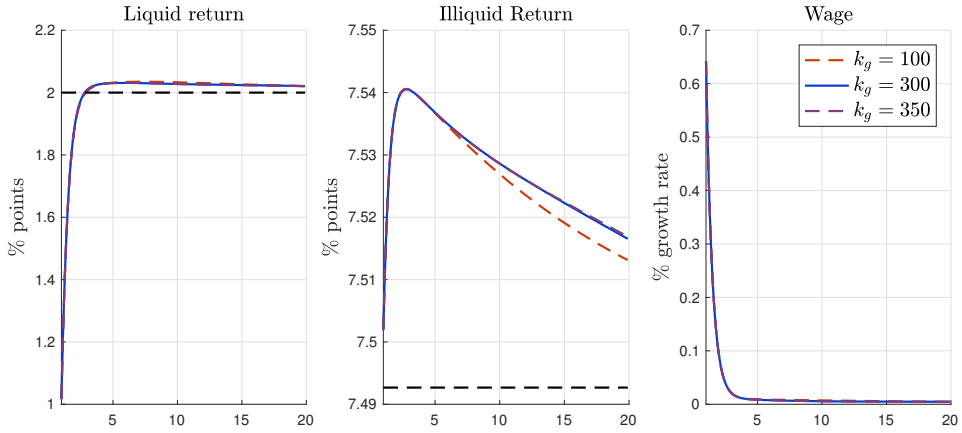
	Data	Model
$\sigma\left(\Delta \log Y_t\right)$	0.89%	0.88%
$\text{Corr}(\Delta \log Y_t, \Delta \log Y_{t-1})$	0.37	0.36

Notes: Targeted moments of real GDP growth, 1953q1 - 2016q2.

Aggregate Income Dynamics Given the parameters of the production function, and the optimal decisions of households, we choose the standard deviation and persistence of aggregate productivity shocks to match the autocorrelation and volatility of equilibrium real GDP growth as reported in Table 3.7. This gives $\nu = 2.15$ and $\sigma = 0.04$.[54]

[54]Most heterogeneous agent models of consumption assume that aggregate income follows a stationary AR(1) process around a deterministic trend. This implies that aggregate income growth is negatively autocorrelated over short horizons, which is at odds with the data.

Figure 3.7: Impulse Responses for Different Orders of Distribution Reduction



Notes: impulse response to a positive Dirac shock of size 1. We simulate the model by discretizing the time dimension with step size d$t = 0.1$ using an implicit updating scheme. We compute the paths of outcome variables by including them in the $\mathbf{p}_t$ vector of variables to compute accurately in the reduction, and use their linearized dynamics.

3.4.3 Computation

Our discretization of the individual state space (a, b, z) contains $N = 66,000$ points, implying implying the total unreduced dynamic system is more than $132,000$ equations in $132,000$ variables.[55,56] We reduce the distribution $\widehat{\mathbf{g}}_t$ using a $k_g = 300$ order observability matrix as our basis. Figure 3.7 shows that the impulse responses of the three prices in the model – the liquid return r_t^b, the illiquid return r_t^a, and the wage w_t – have converged by $k_g = 300$. We reduce the value function $\widehat{\mathbf{v}}_t$ using the spline approximation discussed in Section 3.3.4, bringing the size of the value function down from $N = 66,000$ gridpoints to $k_v = 2,145$ knot points.

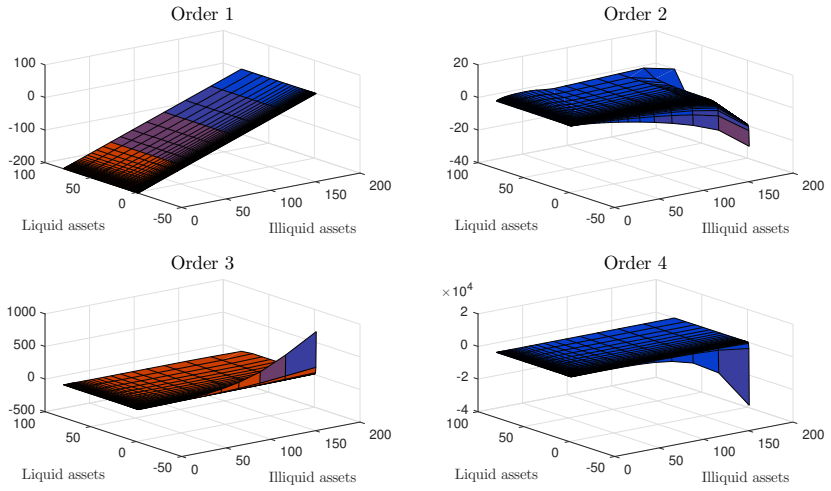
The fact that the two-asset model requires $k_g = 300$ indicates that "approximate aggregation" does not hold.[57] Figure 3.7 shows that even with $k_g = 100$, the reduced model does not correctly

[55]Recall that the dynamic system for the simple Krusell-Smith model was of dimension 400. The reason the two-asset model is so much larger is that the individual state space is three-dimensional. To ensure an accurate approximation of the steady state, we use 33 grid points for labor productivity, 40 points for illiquid assets, and 50 points for liquid assets. The total number of grid points is thus $N = 33 \times 40 \times 50 = 66,000$.

[56]The results presented in this section are for an older and slightly different calibration of the model than described in Section 3.4.2. The final draft will include results for the current calibration, which we do not expect to significantly differ from the results presented here.

[57]Recall that $k_g = 1$ provided an accurate approximation in the simple Krusell and Smith (1998) model.

Figure 3.8: Basis Vectors for Approximating Distribution in Two-Asset Model



Notes: basis vectors of $\mathbf{X}_g$ corresponding to forecasting aggregate capital K_t. Plotted over liquid and illiquid assets for a given value of labor productivity z. Each plot shows the weight placed on regions of the asset space for a given order of approximation. Basis vectors have not been orthonormalized using deflated Arnoldi iteration.

capture the dynamics of the illiquid return r_t^a. Hence, approximating the distribution with a small number of moments would be infeasible in this model.

To illustrate the reason approximate aggregation fails in the two-asset model, Figure 3.8 plots the first three basis vectors in our distribution reduction method. Each panel in the figure plots the basis vector over liquid and illiquid assets given a typical value of labor productivity z necessary for forecasting aggregate capital. Analogously to Figure 3.2 in the Krusell and Smith (1998) model, the first basis vector captures exactly the mean of the illiquid asset distribution corresponding to aggregate capital. The next three basis vectors focus on particular regions of the state space over which individual behavior substantially varies. These regions need to be tracked separately in the distribution reduction.

Overall, our toolbox solves and simulates the model in `5 mins, 49 secs`. Table 3.8 decomposes the total runtime; more than 74% is spent reducing the model and the remaining time is split evenly between solving for the steady state and the remaining tasks. To give a sense of how our method scales with the size of the distribution reduction, Table 3.8 also reports the runtime

Table 3.8: Run Time for Solving Two-Asset Model

	$k_g = 300$	$k_g = 150$
Steady State	47.00 sec	47.00 sec
Derivatives	21.91 sec	21.91 sec
Dim reduction	258.80 sec	79.90 sec
Linear system	17.14 sec	12.66 sec
Simulate IRF	3.76 sec	2.12 sec
Total	**348.61 sec**	**171.58 sec**

Notes: Time to solve the two-asset model on a `MacBook Pro 2016` laptop with `3.3 GHz` processor and `16 GB RAM`, using `Matlab R2016b` and our code toolbox. k_g refers to order of Taylor expansion used to compute basis $\mathbf{X}$. "Steady state" reports time to compute steady state. "Derivatives" reports time to compute derivatives of discretized equilibrium conditions. "Dim reduction" reports time to compute both the distribution and value function reduction. "Linear system" reports time to solve system of linear differential equations. "Simulate IRF" reports time to simulate impulse responses reported in Figure 3.7. "Total" is the sum of all these tasks.

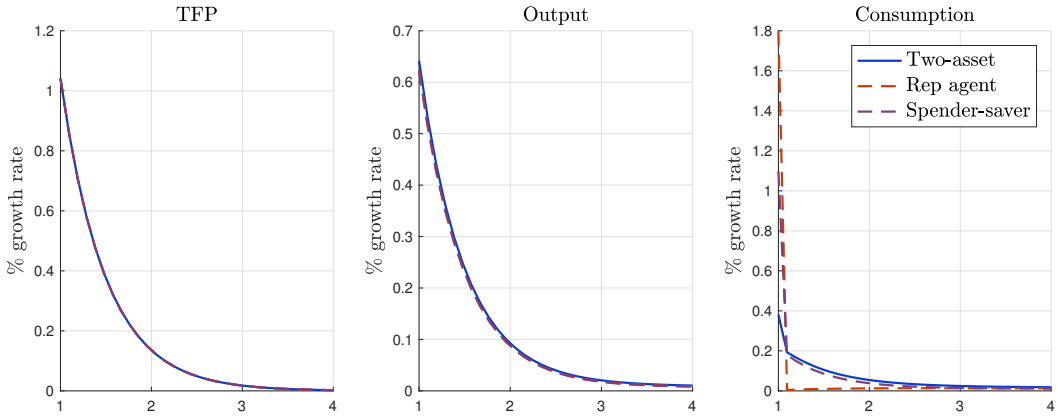
for a $k_g = 150$ order approximation of the distribution. With this smaller approximation of the distribution, the run time is almost halved to `2 mins, 52 secs`.

3.4.4 Aggregate Consumption Dynamics

Having calibrated the model to match features of the distribution of income, wealth, and marginal propensities to consume across households, we now study its implications for the joint dynamics of aggregate consumption and income. The spirit of our analysis is to take the unconditional dynamics of aggregate income as given – recall that we calibrated the process for TFP growth Z_t to match the volatility and persistence of income growth in the data – and study the resulting co-movement of income with consumption. We compare these joint dynamics to the data as well as to simpler models that abstract from household heterogeneity.

Impulse Response to Productivity Growth Shock To fix ideas, Figure 3.9 displays the impulse responses of aggregate output and consumption growth to a positive TFP growth shock Z_t. The shock directly increases output growth as well as increases the wage w_t and illiquid asset return

Figure 3.9: Impulse Responses to TFP Growth Shock in Two-Asset Model



Notes: impulse response to a positive Dirac shock of size 1. We simulate the model by discretizing the time dimension with step size d$t = 0.1$ using an implicit updating scheme. We compute the paths of outcome variables by including them in the $\mathbf{p}_t$ vector of variables to compute accurately in the reduction, and use their linearized dynamics. "Two-asset" model refers to our two-asset model described above. "Representative agent" model refers to the representative agent version of the model described in Appendix **??**, which corresponds to the RBC model with permanent productivity shocks and no labor supply. "Spender-saver" model refers to the spender-saver model described in Appendix **??**, which extends the RBC model to include an exogenous fraction λ of households that consume their income each period. $\lambda = 0.275$ is calibrated to reproduce the IV($\Delta \log C_t$ on $\Delta \log Y_t | \Delta \log Y_{t-1}$) sensitivity coefficient from Table 3.9 below.

r_t^a as in Figure 3.7. These price increases stimulate household income and therefore consumption growth.

Figure 3.9 also compares the impulse responses from our two-asset model with two reference models that abstract from realistic household-level heterogeneity. The first reference is the representative agent version of our model, in which the household side is replaced by one household with the same preferences who can only save in aggregate capital. The second reference model is a spender-saver model in the spirit of Campbell and Mankiw (1989), in which a constant fraction λ of households consumie their income each period, and the remaining $1 - \lambda$ households make a consumption-savings decision as in the representative agent model.

Across all three models, the dynamics of aggregate output are nearly identical by construction. However, the models differ substantially in their implied dynamics of aggregate consumption. The TFP growth shock Z_t causes permanent income to immediately jump to a new level. In the

representative agent model, the permanent-income household therefore increases consumption to a new level as well, causing a one-time increase in consumption growth.[58] The spender-saver model dampens this behavior because $\lambda = 0.275$ of households set their consumption equal to income, decreasing the jump and prolonging the dynamics. Consumption in our two-asset model more closely resembles the spender-saver model in terms of the size of the initial jump and its dynamics.[59]

Sensitivity and Smoothness Following a long tradition in business cycle analysis, we compare the joint dynamics of consumption and income with data through the lens of two sets of facts. The first set of facts, known as *sensitivity*, describe how aggregate consumption growth co-moves with predictable changes in aggregate income growth. We present two measures of sensitivity in the top panel of Table 3.9: (i) the regression coefficient of consumption growth on lagged income growth and (ii) the instrumental variables coefficient of consumption growth on current income growth, instrumented using lagged income growth. In this table, we measure aggregate consumption as the sum of real nondurables and durable services per capita and measure aggregate income as real GDP per capita, both 1953q1 - 2016q2. While both of these statistics capture predictable changes in income – to the extent that past observations are part of agents' information sets – we prefer the instrumental variables coefficient because it focuses on the contemporaneous relationship between consumption and income. The coefficient is approximately 0.5, indicating the 50% of predictable income changes are passed through to consumption.[60]

The second set of facts, known as *smoothness*, refer to the extent of time-series variation in aggregate consumption growth. We present two measures of smoothness in the bottom panel of Table 3.9: (i) the standard deviation of consumption growth relative to that of income growth; and

[58] If the TFP growth shocks Z_t were not persistent and interest rates were constant, the increase in permanent income would be constant and consumption growth would return permanently to zero immediately after the shock. These conditions are not literally met in our model, but they hold approximately.

[59] In our current calibration of the two-asset model, there are a fraction 0.29 of hand-to-mouth consumers in steady state, compared to $\lambda = 0.275$ in the spender-saver model.

[60] Campbell and Mankiw (1989) instrument income growth with a broader set of variables including many lags of past income growth and past interest rates. For our sample period, resulting sensitivity coefficient is nearly identical to our simple instrumental variables coefficient. However, the model analogues of these coefficients are sensitive to assumptions about how interest rates are determined on the production side of the economy.

Table 3.9: Joint Dynamics of Consumption and income

Sensitivity to Income				
	Data		**Models**	
		Rep agent	*Two-Asset*	*Sp-Sa*
Reg($\Delta \log C_t$ on $\Delta \log Y_{t-1}$)	0.184***	0.093	0.251	0.189
	(0.031)			
IV($\Delta \log C_t$ on $\Delta \log Y_t \mid \Delta \log Y_{t-1}$)	0.503***	0.247	0.656	0.505
	(0.083)			
Smoothness				
	Data		**Models**	
		Rep agent	*Two-Asset*	*Sp-Sa*
$\frac{\sigma(\Delta \log C_t)}{\sigma(\Delta \log Y_t)}$	0.518	0.709	0.514	0.676
Corr($\Delta \log C_t, \Delta \log C_{t-1}$)	0.452***	0.153	0.581	0.342
	(0.056)			

Notes: Measures of the sensitivity of aggregate consumption to aggregate income and the smoothness of aggregate consumption. In the data, aggregate consumption C_t is measured as the sum of real nondurable plus durable services, per capita, and aggregate income Y_t is real GDP per capita. Both series are quarterly 1953q1 - 2016q2. "Rep agent" refers to the representative agent model described in Appendix **??**. "Two-asset" refers to the full two-asset model. "Sp-Sa" refers to the spender-saver model described in Appendix **??**. "One-asset" refers to the one-asset version of our model, which assumes all assets are productive capital which is traded subject to $a \geq 0$. The discount factor is recalibrated to match the same aggregate wealth to income ratio as the two-asset model.
Reg($\Delta \log C_t$ on $\Delta \log Y_{t-1}$) refers to the regression coefficient of consumption growth on lagged income growth. IV($\Delta \log C_t$ on $\Delta \log Y_t \mid \Delta \log Y_{t-1}$) refers to the instrumental variables coefficient of consumption growth on income growth, instrumented using lagged income growth. We time-aggregate our continuous time model to the quarterly frequency by computing the simple average within a quarter.

(ii) the autocorrelation of consumption growth. The standard deviation of consumption growth is about half that of income growth, and the autocorrelation of consumption growth (0.45) is slightly larger than the autocorrelation of income growth (0.37).

Campbell and Mankiw (1989) and Ludvigson and Michaelides (2001), among others, argue that these two sets of facts are a useful metric for comparing models of aggregate consumption dynamics with time-series data. As they argue, the representative agent version of our model does not generate a quantitatively realistic degree of sensitivity; both measures are about half the size as in the data. The low sensitivity of this model reflects the fact that the representative household behaves according to the permanent income hypothesis and takes into account predictable changes in income when forming consumption plans.[61]

In contrast, the two-asset model is broadly consistent with both the degree of sensitivity and the degree of smoothness that we measure in the data. The instrumental variables coefficient of consumption growth on income growth is 0.656, within two standard errors of the empirical estimate. As we discuss below, this sensitivity is driven by the presence of hand-to-mouth households who have high MPCs out of changes in income. At the same time, the relative volatility of consumption to income is 0.514, nearly identical to the data.

In fact, our two-asset model is competitive with the spender-saver model, in which the fraction of hand-to-mouth consumers $\lambda = 0.275$ has been explicitly parameterized to match the instrumental variables sensitivity coefficient. Hence, the spender-saver model performs better on sensitivity by construction. However, the spender-saver model overpredicts the volatility of consumption compared to the data, and therefore provides a worse fit along smoothness.

Role of Hand-to-Mouth Households Generating a realistic fraction of hand-to-mouth households is crucial to generating a realistic degree of sensitivity in the two-asset model. As discussed in Figure 3.9, hand-to-mouth households' consumption inherits persistence from the dynamics of aggregate output, while permanent income consumers adjust immediately. Hence, the presence

[61] The representative agent version of our model overpredicts the volatility of consumption relative to income. As discussed in Figure 3.9, the representative household strongly responds to shocks because the shocks directly affect permanent income.

of hand-to-mouth households in our model generates sensitivity of consumption to predictable changes in income.

As Kaplan and Violante (2014) extensively discuss, standard one-asset incomplete markets models, in the spirit of Aiyagari (1994) and Krusell and Smith (1998), do not generate enough high hand-to-mouth households relative to the data. Table 3.9 shows that this model generates corresponding less sensitivity of consumption to income.[62] In the one-asset model, the only high-MPC households are those at the borrowing constraint. However, only a small fraction of households are borrowing constrained in the model because it is relatively easy to save out of the constraint. In the two-asset model, the transaction costs $\chi(d, a)$ make it more costly to save away from the constraint.

Interest Rate Dynamics and the Role of General Equilibrium In response to the positive TFP growth shock Z_t, our two-asset model generates a sharp fall in the liquid return r_t^b. This fall occurs because households demand for liquid savings increases but the supply is fixed. We now show that our results about the sensitivity and smoothness of consumption are robust to two alternative specifications of general equilibrium in which the liquid return is either constant or increases following a TFP growth shock. In the first, we assume that the liquid return r_t^b is fixed and supplied perfectly elastically at that price. In the second, we assume that the liquid asset is also used for physical capital $K_t = \int (a + b) g_t(a, b, z) \mathrm{d}a\mathrm{d}b\mathrm{d}z$, and that there is a constant wedge between r_t^a and r_t^b equal to the steady state difference between the two rates in our benchmark model.

Table 3.10 shows that these alternative specifications change the quantitative magnitudes of our results, but do not alter the qualitative conclusion that the model generates both sensitivity and smoothness. In fact, the alternative models both predict that consumption is too sensitive to income. Fully addressing the role of general equilibrium is outside the scope of this paper. Our

[62] The one-asset model we analyze is a direct simplification of our model in which all assets are productive capital which agents can trade subject to the borrowing constraint $a \geq 0$; there are no other portfolio adjustment costs. We recalibrate the discount rate ρ so that the steady state aggregate wealth to income ratio is the same as in the two-asset model.

Table 3.10: Role of General Equilibrium

Sensitivity to Income				
	Data		**Models**	
		Original	*K liquid*	*Fixed* r_b
Reg($\Delta \log C_t$ on $\Delta \log Y_{t-1}$)	0.184***	0.251	0.364	0.314
	(0.031)			
IV($\Delta \log C_t$ on $\Delta \log Y_t \mid \Delta \log Y_{t-1}$)	0.503***	0.656	0.955	0.793
	(0.083)			
Smoothness				
	Data		**Models**	
		Original	*K liquid*	*Fixed* r_b
$\frac{\sigma(\Delta \log C_t)}{\sigma(\Delta \log Y_t)}$	0.518	0.514	0.606	0.630
Corr($\Delta \log C_t, \Delta \log C_{t-1}$)	0.452***	0.581	0.425	0.556
	(0.056)			

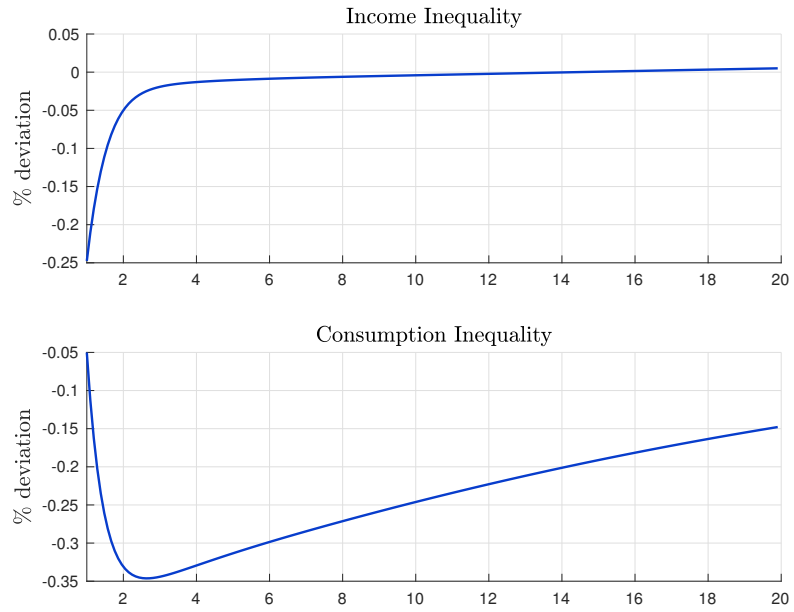
Notes: reproduces Table 3.9 for two alternative general equilibrium specifications of the two-asset model. "K liquid" refers to the model in which aggregate capital is the sum of liquid and illiquid assets, and there is a constant wedge between the two returns equal to the steady state difference in the original two-asset model. "Fixed r_b" refers to the model in which $r_t^b = 0.005$, its steady state value, and bonds are perfectly elastically supplied at that price.

production side is kept purposely simple and therefore does not contain the necessary richness to quantitatively capture the joint determination of interest rates and output in the economy.

3.4.5 Distributional Implications of TFP Growth Shocks

Thus far, we have demonstrated how accurately capturing inequality across households matter for understanding the response of aggregate variables to aggregate shocks. We now briefly explore how aggregate shocks in turn affect the dynamics of inequality. We focus on the dynamics of labor income and consumption inequality. We show that our benchmark two-asset model with TFP growth shocks is ill-suited to this task because the distribution of labor income is constant.

Figure 3.10: Distributional Responses to TFP Growth Shock in Two-Asset Model



Notes: impulse response to a Dirac shock of size 1. We simulate the model by discretizing the time dimension with step size d$t = 0.1$ using an implicit updating scheme. We compute the paths of outcome variables by including them in the $\mathbf{p}_t$ vector of variables to compute accurately in the reduction, and use their linearized dynamics. "Consumption inequality" is the cross-sectional standard deviation of log consumption. "Income inequality" is the cross-sectional standard deviation of log income.

This is a generic feature of models in which efficiency units of labor are perfect substitutes. In Section 3.5 we will we therefore extend the model to include endogenous heterogeneity in labor income across households.

Figure 3.10 plots the impulse response of cross-sectional income and dispersion to a TFP growth shock Z_t in our benchmark two-asset model. The dispersion of total income falls because the TFP shock has a stronger effect on the wage than the return to assets. Low-income households rely disproportionately on labor income and therefore benefits more directly from the TFP shock. In turn, consumption inequality falls, as shown in the bottom panel.

Quantitatively, the time-series variation in income dispersion is 0.37% of its mean and the variation in consumption dispersion is 2.06% of its mean, which is on the order of the time-series variation in aggregate output and consumption. Therefore, even in this benchmark two-asset model,

TFP growth shocks Z_t have a non-negligible effect on inequality. However, as discussed above, the dispersion in labor income $\log w_t + \log z_{jt}$ is constant because it is only driven by the dispersion in the exogenous process for z_{jt}. We now turn to augmenting the production side of our model to generate endogenous movement in labor income inequality.

3.5 Richer Macro and Inequality Interactions

Section 3.4 demonstrated how accurately capturing inequality across households matter for understanding the response of aggregates to aggregate shocks. We focussed on one particular type of aggregate shock, namely a shock to the growth rate of aggregate TFP. We also briefly explored the implications of that shock for the dynamics of inequality but found that our benchmark two-asset model with TFP growth shocks is ill-suited to this task because the distribution of labor income is constant.

Therefore, in the present section we extend the model to include endogenous heterogeneity in labor income across households. In particular, we augment the production function to include high-skill and low-skill workers with different degrees of substitutability with capital. High- and low-skill workers are differentially exposed to aggregate shocks. We show that this augmented production structure can lead to rich interactions between inequality and the macroeconomy. In Sections 3.5.2 and 3.5.3, we illustrate how capital-specific and unskilled-labor specific productivity shocks have strong effects on the distribution of income across households. In turn, the resulting aggregate dynamics are substantially different from the representative agent counterpart. These results therefore provide a counterexample to the main result of Krusell and Smith (1998) who found that, in their particular baseline heterogeneous agent model, the business cycle properties are virtually indistinguishable from those in the representative agent counterpart.

3.5.1 Augmented Production Structure

Following Krusell et al. (2000), we modify the production function to feature high- and low-skill workers and capital-skill complementarity. Unlike the Cobb-Douglas case, factor-specific shocks in this model have differential effects on the distribution of labor income.

Production Function The production function is

$$Y_t = \left[\mu(Z_t^U U_t)^\sigma + (1-\mu)\left(\lambda(Z_t^K K_t)^\rho + (1-\lambda)S_t^\rho\right)^{\frac{\sigma}{\rho}}\right]^{\frac{1}{\sigma}}, \tag{3.38}$$

where Z_t^U is an unskilled labor-specific productivity shock, Z_t^K is a capital-specific productivity shock, U_t is the amount of unskilled labor, and S_t is the amount of skilled labor (all described in more detail below). The elasticity of substitution between unskilled labor and capital, which is equal to the elasticity between unskilled and skilled labor, is $\frac{1}{1-\sigma}$. The elasticity of substitution between skilled labor and capital is $\frac{1}{1-\rho}$. Capital and skilled labor are complements when $\sigma > \rho$, i.e. when capital is more substitutable with unskilled labor than with skilled labor.[63]

We assume there is a simple mapping from labor productivity z_{jt} into skill. Recall that we calibrated the idiosyncratic shock process

$$\log z_{jt} = z_{1,jt} + z_{2,jt}$$

where we found that $z_{1,jt}$ is a transitory shock and $z_{2,jt}$ is a permanent shock. We assume that households with values of $z_{2,jt}$ above a certain threshold are skilled and those below are unskilled. We choose the threshold as the midpoint of our discretization of the shock process.

[63] Krusell et al. (2000) assume that only equipment capital features capital-skill complementarity while structures capital has unitary elasticity of substitution. We omit structures capital for simplicity.

Factor-Specific Shocks We assume that the unskilled labor-specific and capital-specific productivity shocks follow independent Ornstein-Uhlenbeck processes in logs:[64]

$$\begin{aligned} \mathrm{d}\log Z_t^U &= -\eta_U \log Z_t^U \mathrm{d}t + \sigma_U \mathrm{d}W_t^U \\ \mathrm{d}\log Z_t^K &= -\eta_K \log Z_t^K \mathrm{d}t + \sigma_K \mathrm{d}W_t^K. \end{aligned}$$

where η_U and η_K control the rate of mean reversion and σ_K and σ_U control the size of innovations.

Calibration We set the elasticities of substitution to the values estimated in Krusell et al. (2000): $\sigma = 0.401$ and $\rho = -.495$. Since $\sigma > \rho$, the production function features capital-skill complementarity, so capital-specific productivity shocks disproportionately favor skilled labor over unskilled labor. Given these values for the elasticities, we calibrate the factor shares μ and λ so that the steady state labor share is approximately 0.6, as in Section 3.4, and the steady state skill premium is 91%.[65]

We set the rate of mean reversion of the unskilled labor-specific and capital-specific productivity shocks to $\eta_U = \eta_K = 0.25$. We then set the standard deviation of the innovations σ_U and σ_K so that they generate the same increase in output upon impact as a neutral TFP shock with standard deviation 0.007. Hence, with a benchmark Cobb-Douglas production function, the two shocks have the same aggregate impact.

3.5.2 Capital-Specific Productivity Shocks

We first study the effect of a positive capital-specific productivity shock. Because our production function features capital-skill complementarity, high-skill workers benefit disproportionately more from the boom.

[64] We do not assume that these shocks are nonstationary like the TFP shocks in Section 3.4 because, outside of Cobb-Douglas, they do not admit a balanced growth path.

[65] This calibration affects the steady state of our model relative to the moments reported in Section 3.4; in the next draft, we will recalibrate the factor shares and distribution of labor income so that the steady state is unchanged.

Distributional Implications The bottom left panel of Figure 3.11 shows that the capital-specific shock increases high-skill wages by more than low-skill wages due to capital-skill complementarity; in response to the capital-specific shock, the representative firm substitutes toward skilled labor. Hence, labor income inequality increases, generating an increase in consumption inequality as well.

Aggregate Implications The other panels of Figure 3.11 plot the responses of aggregate variables to the capital-specific shock and compare them to the representative household version of the model. Output increases by roughly the same amount in both models. However, consumption increases by substantially more in our model than in the representative agent model due to the presence of high-MPC households. The striking difference between the impulse responses of aggregate consumption in the heterogeneous- and representative agent models provides a counterexample to the main result of Krusell and Smith (1998) that such impulse responses tend to look similar in the two classes of models.

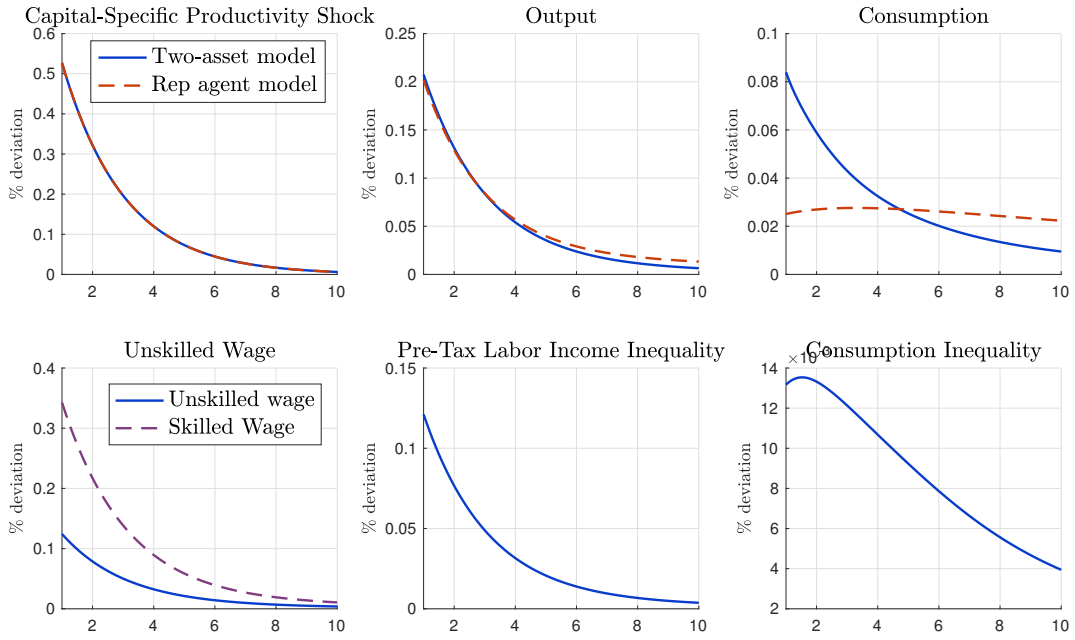
3.5.3 Unskilled-Specific Productivity Shocks

We now study the effect of a negative unskilled labor-specific productivity shock Z_t^U. Because workers are not perfectly substitutable, low-skill workers are disproportionately affected by the bust.

Distributional Implications Similar to Section 3.5.2, the bottom left panel of Figure 3.12 shows that the unskilled-specific shock sharply decreases low-skilled workers' wages, but barely affects high-skilled workers' wages. This effect increases labor income inequality, and therefore consumption inequality following the shock.

Aggregate Implications The other panels of Figure 3.11 plots the impulse responses of aggregate variables to the negative unskilled labor-specific shock and compares them to the representative household version of the model. Output decreases by roughly the same amount in both models,

Figure 3.11: Impulse Responses to Capital-Specific Productivity Shock



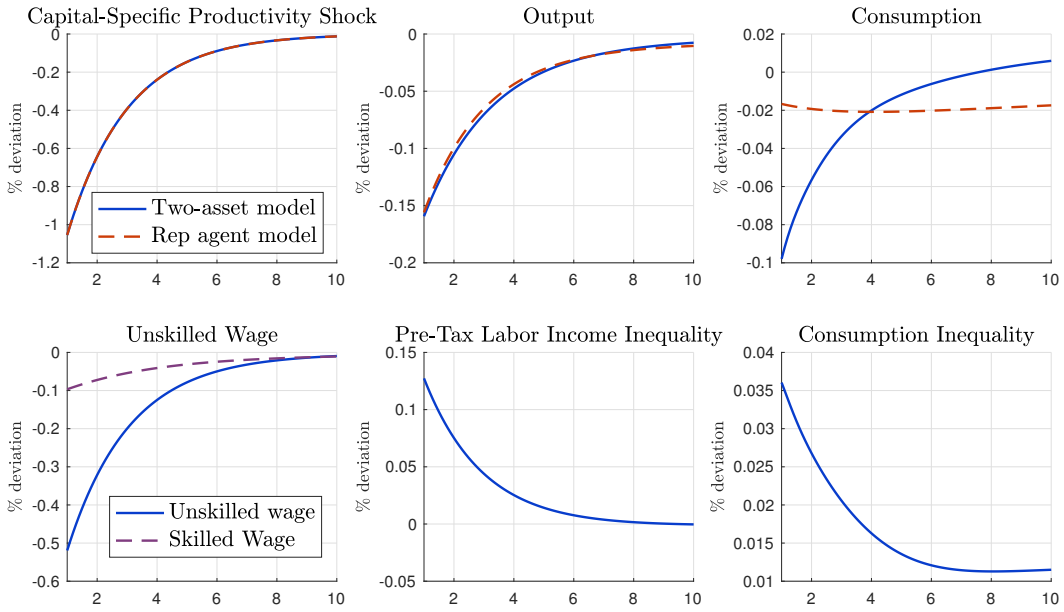
Notes: impulse response to a positive Dirac shock of size 1 to capital-specific productivity Z_t^K. We simulate the model by discretizing the time dimension with step size $\mathrm{d}t = 0.1$ using an implicit updating scheme. We compute the paths of outcome variables by including them in the $\mathbf{p}_t$ vector of variables to compute accurately in the reduction, and use their linearized dynamics. The top panel compares responses in our full two-asset model with the representative agent version of the model. The bottom panel plots responses in the two-asset model only. "Pre-tax labor income inequality" is the cross-sectional standard deviation of log labor income. "Consumption inequality" is the cross-sectional standard deviation of log consumption.

but consumption falls by five times as much in the two-asset model. This occurs because low-skill households have high MPCs out of changes in income, and therefore must cut consumption by more than the representative household. As before, the different impulse responses of aggregate consumption can be viewed as a counterexample to Krusell-Smith's main finding.

3.6 Conclusion

Thus far, heterogenous agent models have not taken off as much as the potential insights afforded by such rich models. There have been two major roadblocks for the takeoff of the heterogeneous

Figure 3.12: Impulse Responses to Unskilled Labor-Specific Productivity Shock



Notes: impulse response to a negative Dirac shock of size 1 to unskilled labor-specific productivity Z_t^U. We simulate the model by discretizing the time dimension with step size $\mathrm{d}t = 0.1$ using an implicit updating scheme. We compute the paths of outcome variables by including them in the $\mathbf{p}_t$ vector of variables to compute accurately in the reduction, and use their linearized dynamics. The top panel compares responses in our full two-asset model with the representative agent version of the model. The bottom panel plots responses in the two-asset model only. "Pre-tax labor income inequality" is the cross-sectional standard deviation of log labor income. "Consumption inequality" is the cross-sectional standard deviation of log consumption.

models. The first roadblock is the difficulty in solving the heterogeneous agent models. This paper presented how a linearization method can be used to make solving such models simple, and exhibited how the model reduction literature from engineering can be used to reduce the state space to solve problems that cannot be solved with existing methods. The second roadblock is the (possible) (mis)conception that heterogeneity does not matter. Hence, the solution method with the toolbox has been applied to two models to show how distributions can affect aggregate variables and how macroeconomy can affect distributions.

3.7 Appendix

3.7.1 Fully Recursive Formulation of

When writing the equilibrium conditions (3.2) to (3.7), we used recursive notation with respect to the idiosyncratic states (a, z) but time-dependent notation with respect to those aggregate states (g, Z). For completeness, this Appendix shows how to write the corresponding equations using fully recursive notation; and how to obtain the hybrid-notation conditions in the main text from the equations using fully recursive notation.

To this end, define the wage and interest rate as a function of the state variables (g, Z)

$$w(g, Z) = (1 - \alpha)e^Z K(g)^\alpha, \tag{3.39}$$

$$r(g, Z) = \alpha e^Z K(g)^{\alpha-1} - \delta, \tag{3.40}$$

$$\text{where } K(g) = \int ag(a, z)dadz \tag{3.41}$$

is the aggregate capital stock as a function of the distribution. Furthermore, define the "Kolmogorov Forward operator" $\mathcal{K}_Z$ that operates on distributions g as

$$(\mathcal{K}_Z g)(a, z) := -\partial_a[s(a, z, g, Z)g(a, z)] - \lambda_z g(a, z) + \lambda_{z'} g(a, z')$$

where $s(a, z, g, Z)$ is the optimal saving policy function (determined below). This operator maps distribution functions g into time derivatives of that distribution. Using this tool, one can for example write the Kolmogorov Forward equation (3.3) compactly as

$$\frac{\mathrm{d}g_t(a, z)}{\mathrm{d}t} = (\mathcal{K}_Z g_t)(a, z).$$

With this machinery in hand, the fully recursive, infinite-dimensional HJB equation is:

$$\begin{aligned}\rho V(a,z,g,Z) = \max_c \; & u(c) + \partial_a V(a,z,g,Z)(w(g,Z)z + r(g,Z)a - c) \\ & + \lambda_z(V(a,z',g,Z) - V(a,z,g,Z)) \\ & + \partial_Z V(a,z,g,Z)(-\eta Z) + \frac{1}{2}\partial_{ZZ} V(a,z,g,Z)\sigma^2 \\ & + \int \frac{\delta V(a,z,g,Z)}{\delta g(a,z)}(\mathcal{K}_Z g)(a,z)\mathrm{d}a\mathrm{d}z\end{aligned} \tag{3.42}$$

The first and second lines in this infinite-dimensional HJB equation capture the evolution of the *idiosyncratic* states (a, z) (just like in (3.2)). The third and fourth lines capture the evolution of the *aggregate* states (g, Z). The third line captures the evolution of aggregate TFP Z with standard "Ito's Formula terms" involving the first and second derivatives of the value function with respect to Z. The fourth line captures the evolution of the distribution g. Since g is a function, it involves the functional derivative of V with respect to g at point (a, z), which we denote by $\delta V/\delta g(a, z)$. The equilibrium in fully recursive notation is then characterized by (3.42) together with (3.39), (3.40) and (3.41).

To understand the last term in (3.42), assume momentarily that the distribution is an N-dimensional vector $\mathbf{g} = (g_1, ..., g_N)$ rather than a function (i.e. an infinite-dimensional object). Then the HJB equation would be

$$\begin{aligned}\rho V(a,z,\mathbf{g},Z) = \max_c \; & u(c) + \partial_a V(a,z,\mathbf{g},Z)(w(\mathbf{g},Z)z + r(\mathbf{g},Z)a - c) \\ & + \lambda_z(V(a,z',\mathbf{g},Z) - V(a,z,\mathbf{g},Z)) \\ & + \partial_Z V(a,z,\mathbf{g},Z)(-\eta Z) + \frac{1}{2}\partial_{ZZ} V(a,z,\mathbf{g},Z)\sigma^2 \\ & + \sum_{i=1}^{N} \frac{\partial V(a,z,\mathbf{g},Z)}{\partial g_i}\dot{g}_i\end{aligned}$$

Since a functional derivative $\delta V/\delta g(a, z)$ is the natural generalization of the partial derivative $\partial V/\partial g_i$ to the infinite-dimensional case, if g is a function rather than a vector we get (3.42).

The equilibrium conditions (3.2) to (3.7) in the main text can be obtained from this system by evaluating "along the characteristic" (g_t, Z_t) that satisfies (3.3) and (3.4). In particular the value function $v_t(a, z)$ in (3.2) is obtained from evaluating (3.42) at (g_t, Z_t), i.e.

$$v_t(a, z) = V(a, z, g_t, Z_t).$$

In particular by Ito's Formula

$$\begin{aligned}\mathrm{d}v_t(a, z) = &\left(\partial_Z V(a, z, g_t, Z_t)(-\eta Z_t) + \frac{1}{2}\partial_{ZZ} V(a, z, g_t, Z_t)\sigma^2\right)\mathrm{dt} + \sigma\partial_Z V(a, z, g_t, Z_t)\mathrm{d}W_t \\ &+ \int \frac{\delta V(a, z, g_t, Z_t)}{\delta g_t(a, z)}(\mathcal{K}_Z g_t)(a, z)\mathrm{d}a\mathrm{d}z\ \mathrm{dt}\end{aligned}$$

and hence using that $\mathbb{E}_t[\mathrm{d}W_t] = 0$

$$\begin{aligned}\frac{1}{\mathrm{d}t}\mathbb{E}_t[\mathrm{d}v_t(a, z)] =&\partial_Z V(a, z, g_t, Z_t)(-\eta Z_t) + \frac{1}{2}\partial_{ZZ} V(a, z, g_t, Z_t)\sigma^2 \\ &+ \int \frac{\delta V(a, z, g_t, Z_t)}{\delta g_t(a, z)}(\mathcal{K}_Z g_t)(a, z)\mathrm{d}a\mathrm{d}z\end{aligned}$$

Similarly, the prices and capital stock in (3.5) to (3.7) are obtained by evaluating (3.39) to (3.41) at (g_t, Z_t), i.e.

$$w_t = w(g_t, Z_t), \quad r_t = r(g_t, Z_t), \quad K_t = K(g_t).$$

3.7.2 Connection to Linearization of Representative-Agent Models

Our solution approach is intimately related to the standard approach of solving representative-agent business-cycle models. For illustration, consider a simple real business cycle (RBC) model. Just like in our heterogeneous-agent models, the equilibrium in this model is characterized by a forward-looking equation for controls, a backward-looking equation for the endogenous state, several static relations and an evolution equation for the exogenous state.

Defining the representative household's marginal utility $\Lambda_t := C_t^{-\gamma}$, the equilibrium conditions can be written as

$$\begin{aligned}
\frac{1}{\mathrm{d}t}\mathbb{E}_t[\mathrm{d}\Lambda_t] &= (\rho - r_t)\Lambda_t \\
\frac{\mathrm{d}K_t}{\mathrm{d}t} &= w_t + r_t K_t - C_t \\
dZ_t &= -\eta Z_t dt + \sigma dW_t \\
r_t &= \alpha e^{Z_t} K_t^{\alpha-1} - \delta \\
w_t &= (1-\alpha) e^{Z_t} K_t^{\alpha}
\end{aligned} \tag{3.43}$$

and where $C_t = \Lambda_t^{-1/\gamma}$. The first equation is the Euler equation. Marginal utility Λ_t is the single control variable. (We could have alternatively written the Euler equation in terms of consumption C_t but working with marginal utility is slightly more convenient). The second equation is the evolution of the aggregate capital stock which is the single endogenous state variable. The third equation is the stochastic process for aggregate productivity which is the exogenous state variable. The last two equations define equilibrium prices. Note that the system (3.14) characterizing the Krusell and Smith (1998) heterogeneous agent model has exactly the same structure as this system for the representative agent model: the discretized value function points $\mathbf{v}_t$ in the heterogeneous agent model play the role of aggregate consumption C_t or marginal utility Λ_t in the RBC model. The discretized distribution $\mathbf{g}_t$ points are endogenous state variables, like aggregate capital K_t in the RBC model. TFP Z_t is an exogenous state variable. Finally, the wage and real interest rate are statically defined variables.

The model's equilibrium conditions can be linearized and the resulting linear system solved exactly as the heterogeneous agent model in the main text.[66] Let hatted variables denote deviations from steady state. Then we have the jump variable $\widehat{\Lambda}_t$, the endogenous state $\widehat{K}_t$, the exogenous

[66] Alternatively, we could *log*-linearize the model. The analysis is analogous.

state Z_t, and the prices $\widehat{\mathbf{p}}_t = (\widehat{r}_t, \widehat{w}_t)$. We can thus write

$$\mathbb{E}_t \begin{bmatrix} \mathrm{d}\widehat{\Lambda}_t \\ \mathrm{d}\widehat{K}_t \\ \mathrm{d}Z_t \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{\Lambda\Lambda} & 0 & 0 & \mathbf{B}_{\Lambda p} \\ \mathbf{B}_{K\Lambda} & \mathbf{B}_{KK} & 0 & \mathbf{B}_{Kp} \\ 0 & 0 & -\eta & 0 \\ 0 & \mathbf{B}_{pK} & \mathbf{B}_{pZ} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\Lambda}_t \\ \widehat{K}_t \\ Z_t \\ \widehat{\mathbf{p}}_t \end{bmatrix} \mathrm{d}t$$

Note that our linearized heterogeneous agent model (3.15) again has exactly the same form as this system. In particular the vector of jump variables $\widehat{\mathbf{v}}_t$ plays the role of $\widehat{\Lambda}_t$ and the vector of endogenous state variables $\widehat{\mathbf{g}}_t$ plays the role of $\widehat{K}_t$.

3.7.3 Model Reduction

In the main text we showed that, in backward-looking systems, our distribution reduction method matches the dynamics of prices up to order k. In this Appendix, we show that this choice also matches the kth order Taylor expansion of the system's impulse response function around $t = 0$.

Deterministic Model

As in the main text consider first the simplified model (3.24) which we briefly restate here:

$$\begin{aligned} \dot{\mathbf{g}}_t &= \mathbf{C}_{gg}\mathbf{g}_t, \\ p_t &= \mathbf{b}_{pg}\mathbf{g}_t. \end{aligned}$$

Solving this for p_t gives

$$\begin{aligned} p_t &= \mathbf{b}_{pg} e^{\mathbf{C}_{gg}t}\mathbf{g}_0 \\ &= \mathbf{b}_{pg}\left[\mathbf{I} + \mathbf{C}_{gg}t + \frac{1}{2}\mathbf{C}_{gg}^2 t^2 + \frac{1}{6}\mathbf{C}_{gg}^3 t^3 + \ldots\right]\mathbf{g}_0 \end{aligned}$$

Now consider the reduced model with

$$\gamma_t = \mathbf{X}^{\mathrm{T}}\mathbf{g}_t, \quad \mathbf{g}_t \approx \mathbf{X}\gamma_t$$

Differentiating with respect to time gives the dynamics

$$\begin{aligned} \dot{\gamma}_t &= \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\gamma_t \\ \widetilde{p}_t &= \mathbf{b}_{pg}\mathbf{X}\gamma_t \end{aligned}$$

and so

$$\begin{aligned} \widetilde{p}_t &= \mathbf{b}_{pg}\mathbf{X}e^{\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}t}\mathbf{X}^{\mathrm{T}}\mathbf{g}_0 \\ &= \mathbf{b}_{pg}\mathbf{X}\left[\mathbf{I} + \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}^{\mathrm{T}}t + \frac{1}{2}(\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}^{\mathrm{T}})^2t^2 + \frac{1}{6}(\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}^{\mathrm{T}})^3t^3 + \ldots\right]\mathbf{X}^{\mathrm{T}}\mathbf{g}_0 \end{aligned}$$

We choose the projection matrix $\mathbf{X}$ to ensure that the dynamics of the reduced $\widetilde{p}_t$ match as closely as possible those of the unreduced p_t. Following insights from the model reduction literature, we take this to mean that Taylor series expansions of p_t and $\widetilde{p}_t$ around $t = 0$ share the first k expansion coefficients. This is ensured by letting $\mathbf{X}$ be a semi-orthogonal basis of the space spanned by the order-k observability matrix $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$:

$$\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg}) := \begin{bmatrix} \mathbf{b}_{pg}^{\mathrm{T}} \\ \mathbf{C}_{gg}^{\mathrm{T}}\mathbf{b}_{pg}^{\mathrm{T}} \\ (\mathbf{C}_{gg}^{\mathrm{T}})^2\mathbf{b}_{pg}^{\mathrm{T}} \\ \vdots \\ (\mathbf{C}_{gg}^{\mathrm{T}})^{k-1}\mathbf{b}_{pg}^{\mathrm{T}} \end{bmatrix}$$

To see this works, we can just consider each term separately in the Taylor series expansions derived above. In all of the following, e_i denotes the ith standard unit vector and $\mathbf{X}^i$ denotes the

ith submatrix of $\mathbf{X}$ (so corresponding to $(\mathbf{C}_{gg}^{\mathrm{T}})^{i-1}\mathbf{b}_{pg}^{\mathrm{T}}$). Now note that first of all we have

$$\begin{aligned}\mathbf{b}_{pg}\mathbf{X}\mathbf{X}^{\mathrm{T}} &= (\mathbf{X}^1)'\mathbf{X}\mathbf{X}^{\mathrm{T}} \\ &= e_1\mathbf{X}^{\mathrm{T}} = (\mathbf{X}^1)' = \mathbf{b}_{pg}\end{aligned}$$

as desired. Next we have

$$\begin{aligned}\mathbf{b}_{pg}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\mathbf{X}^{\mathrm{T}} &= (\mathbf{X}^1)'\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\mathbf{X}^{\mathrm{T}} \\ &= (\mathbf{X}^2)'\mathbf{X}\mathbf{X}^{\mathrm{T}} = e_2'\mathbf{X}^{\mathrm{T}} = (\mathbf{X}^2)' = \mathbf{b}_{pg}\mathbf{C}_{gg}\end{aligned}$$

also as desired. All higher-order terms follow analogously.

Stochastic Model

Solving out prices and the decision rules for the controls v_t, we get the system

$$\begin{aligned}\dot{\mathbf{g}}_t &= (\underbrace{\mathbf{B}_{gg} + \mathbf{B}_{gp}\mathbf{b}_{pg} + \mathbf{B}_{gv}\mathbf{D}_{vg}}_{\mathbf{C}_{gg}})\mathbf{g}_t + (\underbrace{\mathbf{B}_{gv}\mathbf{D}_{vZ}}_{\mathbf{C}_{gZ}})Z_t \\ p_t &= \mathbf{b}_{pg}\mathbf{g}_t + \mathbf{B}_{pZ}Z_t\end{aligned}$$

The dynamics of this stochastic system are characterized by the impulse response function

$$h(t) = \mathbf{b}_{pg}e^{\mathbf{C}_{gg}t}\mathbf{C}_{gZ} + \delta(t)\mathbf{B}_{pZ}$$

with $\delta(t)$ the Dirac delta function. This impulse response function induces the following dynamic behavior:

$$p_t = \underbrace{\mathbf{b}_{pg}e^{\mathbf{C}_{gg}t}\mathbf{g}_0}_{\text{det. part}} + \underbrace{\int_0^t h(t-s)Z_s}_{\text{stoch. part}}$$

As before, we consider a reduced model with

$$\gamma_t = \mathbf{X}^{\mathrm{T}}\mathbf{g}_t, \quad \mathbf{g}_t = \mathbf{X}\gamma_t$$

giving

$$\begin{aligned}\dot{\gamma}_t &= \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\gamma_t + \mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ}Z_t \\ \widetilde{p}_t &= \mathbf{b}_{pg}\mathbf{X}\gamma_t + \mathbf{B}_{pZ}Z_t\end{aligned}$$

This model induces the impulse response function

$$\widetilde{h}(t) = \mathbf{b}_{pg}\mathbf{X}e^{\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}t}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} + \delta(t)\mathbf{B}_{pZ}$$

and so the dynamics

$$\widetilde{p}_t = \mathbf{b}_{pg}\mathbf{X}e^{\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}t}\mathbf{X}^{\mathrm{T}}\mathbf{g}_0 + \int_0^t \widetilde{h}(t-s)Z_s$$

We now proceed exactly as before, with the projection matrix $\mathbf{X}$ chosen as a semi-orthogonal basis of the space spanned by the order-k observability matrix $\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg})$:

$$\mathcal{O}(\mathbf{b}_{pg}, \mathbf{C}_{gg}) := \begin{bmatrix} \mathbf{b}_{pg}^{\mathrm{T}} \\ \mathbf{C}_{gg}^{\mathrm{T}}\mathbf{b}_{pg}^{\mathrm{T}} \\ (\mathbf{C}_{gg}^{\mathrm{T}})^2\mathbf{b}_{pg}^{\mathrm{T}} \\ \vdots \\ (\mathbf{C}_{gg}^{\mathrm{T}})^{k-1}\mathbf{b}_{pg}^{\mathrm{T}} \end{bmatrix}$$

Showing that all terms in the deterministic part are matched is exactly analogous to the argument given above. For the stochastic part, we also do not need to change much. The impact impulse

response $\mathbf{B}_{pZ}$ is matched irrespective of the choice of projection matrix $\mathbf{X}$. Next we have

$$\begin{aligned}\mathbf{b}_{pg}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} &= (\mathbf{X}^1)'\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} \\ &= e_1\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} = (\mathbf{X}^1)'\mathbf{C}_{gZ} = \mathbf{b}_{pg}\mathbf{C}_{gZ}\end{aligned}$$

and finally

$$\begin{aligned}\mathbf{b}_{pg}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} &= (\mathbf{X}^1)'\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gg}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} \\ &= (\mathbf{X}^2)'\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} = e_2'\mathbf{X}^{\mathrm{T}}\mathbf{C}_{gZ} = (\mathbf{X}^2)'\mathbf{C}_{gZ} = \mathbf{b}_{pg}\mathbf{C}_{gg}\mathbf{C}_{gZ}\end{aligned}$$

We are thus matching both the deterministic and the stochastic part of the dynamics up to order k in a Taylor series expansion around time $t = 0$.

3.7.4 Detrending the Two-Asset Model

Many equilibrium objects are nonstationary because productivity shocks are permanent. In this Appendix, we show that the equilibrium can be equivalently represented by a set of normalized objects $v_t(\hat{a}, \hat{b}, z)$, $g_t(\hat{a}, \hat{b}, z)$, $\hat{K}_t$, r_t^a, $\hat{w}_t$, r_t^b, and Z_t such that

1. *Transformed HJB*: $v_t(\hat{a}, \hat{b}, z)$ solves

$$\begin{aligned}(\rho + \zeta - (1-\theta)Z_t)v_t(\hat{a}, \hat{b}, z) = \max_{\hat{c},\hat{d}} \frac{\hat{c}^{1-\theta}}{1-\theta} &+ \partial_b v_t(\hat{a}, \hat{b}, z)(T + (1-\tau)\hat{w}_t e^z + r_t^b(\hat{b})\hat{b} \\ &- \chi(\hat{d}, \hat{a}) - \hat{c} - \hat{d}) + \partial_a v_t(\hat{a}, \hat{b}, z)(r_t^a \hat{a} + \hat{d}) \\ &+ \sum_{z'} \lambda_{zz'}(v_t(\hat{a}, \hat{b}, z') - v_t(\hat{a}, \hat{b}, z)) + \frac{1}{\mathrm{d}t}\mathbb{E}_t[\mathrm{d}\hat{v}_t(\hat{a}, \hat{b}, z)].\end{aligned}$$

   The fact that TFP growth is permanent changes the effective discount factor in the households' HJB equation.

2. *Transformed KFE*: $g_t(\hat{a}, \hat{b}, z)$ evolves according to

$$\begin{aligned}\frac{\mathrm{d}g_t(\hat{a}, \hat{b}, z)}{\mathrm{d}t} =& -\partial_{\hat{a}} s_t^a(\hat{a}, \hat{b}, z) g_t(\hat{a}, \hat{b}, z) - \partial_{\hat{b}} s_t^b(\hat{a}, \hat{b}, z) g_t(\hat{a}, \hat{b}, z) \\ & - \sum_{z'} \lambda_{zz'} g_t(\hat{a}, \hat{b}, z) + \sum_{z'} \lambda_{z'z} g_t(\hat{a}, \hat{b}, z), \text{ where} \\ s_t^b(\hat{a}, \hat{b}, z) =& T + (1-\tau)\hat{w}_t e^z + r_t^b(\hat{b})\hat{b} - \chi(\hat{d}, \hat{a}) - \hat{c} - \hat{d} - \hat{a} Z_t \text{ and} \\ s_t^a(\hat{a}, \hat{b}, z) =& r_t^a \hat{a} + \hat{d} - \hat{b} Z_t.\end{aligned}$$

Permanent TFP shocks change the effective depreciation rate of assets.

3. *Transformed firm conditions*: r_t^a, $\hat{w}_t$, and Z_t satisfy

$$\begin{aligned}r_t^a =& \alpha \hat{K}_t^{\alpha-1} (\overline{L})^{1-\alpha} - \delta \\ \hat{w}_t =& (1-\alpha) \hat{K}_t^{\alpha} \overline{L}^{-\alpha} \\ dZ_t =& -\nu Z_t dt + \sigma dW_t.\end{aligned}$$

4. *Transformed asset market clearing conditions*

$$\begin{aligned}B^* = \hat{B}_t =& \int \hat{b} g_t(\hat{a}, \hat{b}, z) \mathrm{d}\hat{b}\mathrm{d}\hat{a}\mathrm{d}z \\ \hat{K}_t =& \int a g_t(\hat{a}, \hat{b}, z) \mathrm{d}\hat{b}\mathrm{d}\hat{a}\mathrm{d}z\end{aligned}$$

Since aggregate productivity Z_t is nonstationary, we must normalize the model to express the equilibrium in terms of stationary objects. Almost all variables in the model naturally scale with the level of productivity Z_t; for any such variable x_t, let $\hat{x}_t = \frac{x_t}{Z_t}$ denote its detrended version. The one exception to this scheme is the households' value function $v_t(a, b, z)$, which scales with $Z_t^{1-\gamma}$.

HJB Equation Define the detrended value function $\hat{v}_t(a,b,z) = \frac{v_t(a,b,z)}{Z_t^{1-\gamma}}$. Divide both sides of the HJB (3.34) by $Z_t^{1-\gamma}$ and use $\hat{x}_t$ notation where applicable to get

$$\begin{aligned}(\rho+\zeta)\hat{v}_t(a,b,z) =& \max_{c,d} \frac{\hat{c}^{1-\gamma}}{1-\gamma} + \partial_b \hat{v}_t(a,b,z)(TZ_t + (1-\tau)w_t e^z + r_t^b(b)b \\ & \qquad - \chi(d,a)Z_t - c - d) \\ & + \partial_a \hat{v}_t(a,b,z)(r_t^a a + d) \\ & + \sum_{z'} \lambda_{zz'}(\hat{v}_t(a,b,z') - \hat{v}_t(a,b,z)) \\ & + \frac{1}{Z_t^{1-\gamma}} \times \frac{1}{dt}\mathbb{E}_t[dv_t(a,b,z)]. \end{aligned} \tag{3.44}$$

Next, to replace the $\frac{1}{dt}\mathbb{E}_t[dv_t(a,b,z)]$ term, note that by the chain rule

$$\frac{d}{dt}\hat{v}_t(a,b,z) = \frac{\frac{d}{dt}v_t(a,b,z)}{Z_t^{1-\gamma}} + (\gamma-1)d\log Z_t \hat{v}_t(a,b,z),$$

which implies that

$$\frac{1}{Z_t^{1-\gamma}} \times \frac{1}{dt}\mathbb{E}_t[dv_t(a,b,z)] = \frac{1}{dt}\mathbb{E}_t[d\hat{v}_t(a,b,z)] + (1-\gamma)d\log Z_t \hat{v}_t(a,b,z).$$

Plug this back into (3.44) and rearrange to get

$$\begin{aligned}(\rho+\zeta+(\gamma-1)d\log Z_t)\hat{v}_t(a,b,z) =& \max_{c,d} \frac{\hat{c}^{1-\gamma}}{1-\gamma} + \partial_b \hat{v}_t(a,b,z)(TZ_t + (1-\tau)w_t e^z \\ & \qquad + r_t^b(b)b - \chi(d,a)Z_t - c - d) \\ & + \partial_a \hat{v}_t(a,b,z)(r_t^a a + d) + \sum_{z'} \lambda_{zz'}(\hat{v}_t(a,b,z') - \hat{v}_t(a,b,z)) \\ & + \frac{1}{dt}\mathbb{E}_t[d\hat{v}_t(a,b,z)]. \end{aligned} \tag{3.45}$$

The formulation in (3.45) is still not stationary because there are permanent changes in the state variables a and b, the wage w_t, and transaction cost on the right hand side. To address this we

perform a change of variables and characterize the value function in terms of $\hat{a}$ and $\hat{b}$, rather than a and b themselves. Note that

$$\begin{aligned}\partial_b \hat{v}_t(a,b,z) =& \partial_b \hat{v}_t(\frac{a}{Z_t}, \frac{b}{Z_t}, z) = \frac{1}{Z_t}\partial_{\hat{b}} \hat{v}_t(\hat{a},\hat{b},z) \text{ and} \\ \partial_a \hat{v}_t(a,b,z) =& \partial_a \hat{v}_t(\frac{a}{Z_t}, \frac{b}{Z_t}, z) = \frac{1}{Z_t}\partial_{\hat{a}} \hat{v}_t(\hat{a},\hat{b},z).\end{aligned}$$

This implies

$$\begin{aligned}&\partial_b \hat{v}_t(a,b,z)(TZ_t + (1-\tau)w_t e^z + r_t^b(b)b - \chi(d,a)Z_t - c - d) \\ &= \partial_b \hat{v}_t(\hat{a},\hat{b},z)(TZ_t(1-\tau)\hat{w}_t e^z + r_t^b(\hat{b})\hat{b} - \chi(\hat{d},\hat{a}) - \hat{c} - \hat{d})\end{aligned}$$

and

$$\partial_a \hat{v}_t(a,b,z)(r_t^a a + d) = \partial_a \hat{v}_t(\hat{a},\hat{b},z)(r_t^a \hat{a} + \hat{d}).$$

Putting all these results together, we get the final detrended HJB equation

$$\begin{aligned}(\rho + \zeta - (1-\gamma)d\log Z_t)\hat{v}_t(\hat{a},\hat{b},z) = \max_{\hat{c},\hat{d}} \frac{\hat{c}^{1-\gamma}}{1-\gamma} &+ \partial_b \hat{v}_t(\hat{a},\hat{b},z)(T + (1-\tau)\hat{w}_t e^z \\ &+ r_t^b(\hat{b})\hat{b} - \chi(\hat{d},\hat{a}) - \hat{c} - \hat{d}) \\ + \partial_a \hat{v}_t(\hat{a},\hat{b},z)(r_t^a \hat{a} + \hat{d}) + \sum_{z'} \lambda_{zz'}(\hat{v}_t(\hat{a},\hat{b},z') - \hat{v}_t(\hat{a},\hat{b},z)) &+ \frac{1}{dt}\mathbb{E}_t[d\hat{v}_t(\hat{a},\hat{b},z)].\end{aligned} \tag{3.46}$$

KFE The cross-sectional distribution of households over $\hat{a}, \hat{b}, z$ is stationary. We will directly construct the KFE for the distribution over this space. Analogously to (3.35), this is given by

$$\begin{aligned} g_t(\hat{a},\hat{b},z) =& -\partial_{\hat{a}}\dot{\hat{a}}_t(a,b,z)g_t(\hat{a},\hat{b},z) - \partial_{\hat{b}}\dot{\hat{b}}_t(\hat{a},b,z)g_t(\hat{a},\hat{b},z) \\ & - \sum_{z'}\lambda_{zz'}g_t(\hat{a},\hat{b},z) + \sum_{z'}\lambda_{z'z}g_t(\hat{a},\hat{b},z). \end{aligned}$$

By the product rule,

$$\dot{\hat{a}}_t = \frac{\dot{a}_t}{Z_t} - d\log Z_t \hat{a}_t$$

and that from the construction of the modified HJB (3.46) above $\frac{\dot{a}}{Z_t} = r_t^a\hat{a} + \hat{d}$.

Using this result, and the analogous one for $\dot{\hat{a}}_t$, we get the final detrended KFE

$$\begin{aligned} g_t(\hat{a},\hat{b},z) =& -\partial_{\hat{a}}s_t^a(\hat{a},\hat{b},z)g_t(\hat{a},\hat{b},z) - \partial_{\hat{b}}s_t^b(\hat{a},\hat{b},z)g_t(\hat{a},\hat{b},z) \\ & - \sum_{z'}\lambda_{zz'}g_t(\hat{a},\hat{b},z) + \sum_{z'}\lambda_{z'z}g_t(\hat{a},\hat{b},z), \text{ where} \\ s_t^b(\hat{a},\hat{b},z) =& \hat{T}_t + (1-\tau)\hat{w}_t e^z + r_t^b(\hat{b})\hat{b} - \chi(\hat{d},\hat{a}) - \hat{c} - \hat{d} - d\log Z_t\hat{a} \text{ and} \\ s_t^a(\hat{a},\hat{b},z) =& r_t^a\hat{a} + \hat{d} - d\log Z_t\hat{b}. \end{aligned}$$

Other Equilibrium Conditions Detrending the remaining equilibrium conditions is simple:

$$\begin{aligned} r_t^a =& \alpha\hat{K}_t^{\alpha-1}(\overline{L})^{1-\alpha} - \delta \\ \hat{w}_t =& (1-\alpha)\hat{K}_t^{\alpha}\overline{L}^{-\alpha} \\ d\log Z_t =& A_t \\ dA_t =& -\nu A_t + \sigma dW_t. \end{aligned}$$

Bibliography

ACHDOU, Y., AND I. CAPUZZO-DOLCETTA (2010): "Mean field games: Numerical methods," *SIAM Journal on Numerical Analysis*, 48(3), 1136–1162.

ACHDOU, Y., J. HAN, J.-M. LASRY, P.-L. LIONS, AND B. MOLL (2015): "Heterogeneous Agent Models in Continuous Time," Discussion paper, Princeton University.

AHN, S., G. KAPLAN, B. MOLL, T. WINBERRY, AND C. WOLF (2017): *When Inequality Matters for Macro and Macro Matters for Inequality*University of Chicago Press.

AIYAGARI, S. R. (1994): "Uninsured Idiosyncratic Risk and Aggregate Saving," *The Quarterly Journal of Economics*, 109(3), 659–684.

AMSALLEM, D., AND C. FARHAT (2011): "Lecture Notes for CME 345: Model Reduction," `https://web.stanford.edu/group/frg/course_work/CME345/`.

ANTOULAS, A. (2005): *Approximation of Large-Scale Dynamical Systems*. SIAM Advances in Design and Control.

AUCLERT, A. (2014): "Monetary Policy and the Redistribution Channel," Discussion paper, MIT.

BÄCK, J., F. NOBILE, L. TAMELLINI, AND R. TEMPONE (2011): "Stochastic spectral Galerkin and collocation methods for PDEs with random coefficients: a numerical comparison," in *Spectral and High Order Methods for Partial Differential Equations*, pp. 43–62. Springer.

BLANCHARD, O. J., AND C. M. KAHN (1980): "The Solution of Linear Difference Models under Rational Expectations," *Econometrica*, 48(5), 1305–11.

BLOOM, N., M. FLOETOTTO, N. JAIMOVICH, I. SAPORTA-EKSTEN, AND S. TERRY (2014): "Really Uncertain Business Cycles," Discussion paper.

BUNGARTZ, H.-J., AND M. GRIEBEL (2004): "Sparse grids," *Acta numerica*, 13, 147–269.

CAMPBELL, J. (1998): "Entry, Exit, Embodied Technology, and Business Cycles," *Review of Economic Dynamics*, 1(2), 371–408.

CAMPBELL, J. Y., AND N. G. MANKIW (1989): "Consumption, Income and Interest Rates: Reinterpreting the Time Series Evidence," in *NBER Macroeconomics Annual 1989, Volume 4*, NBER Chapters, pp. 185–216. National Bureau of Economic Research.

CARROLL, C., M. WHITE, N. PALMER, D. LOW, AND A. KAUFMAN (2016): "Heterogenous Agents Resources & toolKit," `https://github.com/econ-ark/HARK`.

CARROLL, C. D., J. SLACALEK, AND M. SOMMER (2011): "International Evidence on Sticky Consumption Growth," *The Review of Economics and Statistics*, 93(4), 1135–1145.

CHRISTIANO, L. (1989): "Comment on "Consumption, Income and Interest Rates: Reinterpreting the Time Series Evidence"," in *NBER Macroeconomics Annual 1989, Volume 4*, NBER Chapters, pp. 216–233. National Bureau of Economic Research.

CONGRESSIONAL BUDGET OFFICE (2013): "The Distribution of Federal Spending and Taxes in 2006," Discussion paper, Congress of the United States.

DEN HAAN, W., K. JUDD, AND M. JULLIARD (2010): "Computational Suite of Models with Heterogeneous Agents: Incomplete Markets and Aggregate Uncertainty," *Journal of Economic Dynamics and*

Control, 34(1), 1–3.

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.

DOTSEY, M., R. KING, AND A. WOLMAN (1999): "State-Dependent Pricing and the General Equilibrium Dynamics of Money and Output," *Quarterly Journal of Economics*, pp. 655–690.

FAGERENG, A., M. B. HOLM, AND G. J. NATVIK (2016): "MPC Heterogeneity and Household Balance Sheets," Discussion paper, Statistics Norway.

GERSTNER, T., AND M. GRIEBEL (2003): "Dimension–adaptive tensor–product quadrature," *Computing*, 71(1), 65–87.

GORDON, G. (2011): "Computing dynamic heterogeneous-agent economies: Tracking the distribution," .

GRIEBEL, M. (1998): "Adaptive sparse grid multilevel methods for elliptic PDEs based on finite differences," *Computing*, 61(2), 151–179.

GUVENEN, F., F. KARAHAN, S. OZKAN, AND J. SONG (2015): "What Do Data on Millions of U.S. Workers Reveal about Life-Cycle Earnings Risk?," NBER Working Papers 20913, National Bureau of Economic Research.

JOHNSON, D. S., J. A. PARKER, AND N. S. SOULELES (2006): "Household Expenditure and the Income Tax Rebates of 2001," *American Economic Review*, 96(5), 1589–1610.

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.

KAPLAN, G., B. MOLL, AND G. L. VIOLANTE (2016): "Monetary Policy According to HANK," Working Papers 1602, Council on Economic Policies.

KAPLAN, G., AND G. L. VIOLANTE (2014): "A Model of the Consumption Response to Fiscal Stimulus Payments," *Econometrica*, 82(4), 1199–1239.

KRUEGER, D., AND F. KUBLER (2004): "Computing equilibrium in OLG models with stochastic production," *Journal of Economic Dynamics and Control*, 28(7), 1411–1436.

KRUSELL, P., L. OHANIAN, V. RIOS-RULL, AND G. VIOLANTE (2000): "Capital-Skill Complementarity and Inequality: A Macroeconomic Analysis," *Econometrica*, 68, 1029–1053.

KRUSELL, P., AND A. A. SMITH (1998): "Income and Wealth Heterogeneity in the Macroeconomy," *Journal of Political Economy*, 106(5), 867–896.

LUCAS, R. E. (2003): "Macroeconomic Priorities," *American Economic Review*, 93(1), 1–14.

LUDVIGSON, S. C., AND A. MICHAELIDES (2001): "Does Buffer-Stock Saving Explain the Smoothness and Excess Sensitivity of Consumption?," *American Economic Review*, 91(3), 631–647.

MA, X., AND N. ZABARAS (2009): "An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations," *Journal of Computational Physics*, 228(8), 3084–3113.

MCKAY, A. (2017): "Time-Varying Idiosyncratic Risk and Aggregate Consumption Dynamics," Discussion paper, Boston University.

MCKAY, A., E. NAKAMURA, AND J. STEINSSON (2015): "The Power of Forward Guidance Revisited," NBER Working Papers 20882, National Bureau of Economic Research.

MCKAY, A., AND R. REIS (2013): "The Role of Automatic Stabilizers in the U.S. Business Cycle," NBER Working Papers 19000, National Bureau of Economic Research.

MONGEY, S., AND J. WILLIAMS (2016): "Firm Dispersion and Business Cycles: Estimating Aggregate Shocks Using Panel Data," Working paper, NYU.

PARKER, J. A., N. S. SOULELES, D. S. JOHNSON, AND R. MCCLELLAND (2013): "Consumer Spending and the Economic Stimulus Payments of 2008," *American Economic Review*, 103(6), 2530–53.

PRESTON, B., AND M. ROCA (2007): "Incomplete Markets, Heterogeneity and Macroeconomic Dynamics," NBER Working Papers 13260, National Bureau of Economic Research, Inc.

REITER, M. (2009): "Solving heterogeneous-agent models by projection and perturbation," *Journal of Economic Dynamics and Control*, 33(3), 649–665.

——— (2010): "Approximate and Almost-Exact Aggregation in Dynamic Stochastic Heterogeneous-Agent Models," Economics Series 258, Institute for Advanced Studies.

RUST, J. (1996): "Numerical dynamic programming in economics," *Handbook of computational economics*, 1, 619–729.

SMOLYAK, S. A. (1963): "Quadrature and interpolation formulas for tensor products of certain classes of functions," in *Dokl. Akad. Nauk SSSR*, vol. 4, p. 123.

TERRY, S. (2017): "Alternative Methods for Solving Heterogeneous Firm Models," Discussion paper, Boston University.

WINBERRY, T. (2016): "A Toolbox for Solving and Estimating Heterogeneous Agent Macro Models," Working paper, University of Chicago.