Three Essays in Macroeconomics by

Chase G. Coleman

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Economics

Stern School of Business

New York University

May, 2019

Laura L. Veldkamp

Acknowledgments

First and foremost, I am grateful for a truly exceptional committee consisting of Laura Veldkamp, Thomas Sargent, and Stan Zin. They have each played a unique role in my development as an economist by helping my learn to direct my curiosity, teaching me the tools needed to do economic research, and providing encouragement when I didn't know what to do next. My life is greatly improved because of the time they were willing to share with me.

I am grateful for the collaborative environment that I was able to work in while here at NYU and am lucky to have learned an incredible number of things from various students and professors, including, but not limited to, Felipe Alvarez, Katka Borovickova, Tom Cooley, Richard Davies, Fiona Feng, Axelle Ferriere, Sonia Gilbukh, Victoria Gregory, Timothy Hills, Malika Krishna, Elliot Lipnowski, Roxana Mihet, Simon Mongey, Jonathan Payne, Alberto Polo, Kim Ruhl, Maher Said, Daniel Stackman, Balint Szoke, Venky Venkateswaran, Mike Waugh, Larry, White, Peifan Wu.

Dave Backus, though no longer with us, also played a vital role in my development as an economist and as a person — I often miss him and frequently reflect on the lessons I learned from him.

The Sargent Reading Group has served as an effective tool for me to learn about a wide range of topics in economics — I am grateful to those who have helped organize it each while I attended (Joseba Martinez and Balint Szoke) and to each of the particants who took it seriously by coming prepared with interesting papers and asking insightful, though sometimes difficult, questions.

Outside of NYU, I have also benefitted from a number of wonderful colleagues, including, but not limited to, Rick Evans, Lilia Maliar, Serguei Maliar, Matt McKay, Kerk Phillips, Jim Savage, Simon Scheidegger, John Stachurski, and Natasha Watkins.

This work is dedicated to my wife, Jessica Coleman, and my parents, Stephanie and Gregory Coleman. Each of these three individuals have encouraged me in my education, driven and motivated my curiosity, and supported the paths that these pursuits have lead me to take, often at the price of their own convenience.

Abstract

This dissertation consists of three chapters. These chapters focus on

Chapter 1, ‘Pareto weights as wedges in two-country models‘, was written with Dave Backus, Axelle Ferriere, and Spencer Lyon. In this chapter we study how in models with recursive preferences, endogenous variation in Pareto weights would be interpreted as wedges from the perspective of a frictionless model with additive preferences. We describe the behavior of the relative Pareto weight in a two-country world and explore its interaction with consumption and the real exchange rate.

Chapter 2, ‘Global Solution Methods for Macroeconomic Models‘, was written jointly with Spencer Lyon, Lilia Maliar, and Serguei Maliar. This chapter first discusses 7 global solution methods in the context of a simple neoclassical growth model and then introduces a new 8th global solution method in the context of a non-trivial New-Keynesian model. We first highlight how algorithm choice, even more than programming language, plays an important role in the speed that a model can be solved. The new method introduced in this paper is capable of solving a New-Keynesian model with an 8-dimensional state-space in mere seconds which makes it possible to perform model calibration.

In Chapter 3, ‘The Cost of Income-Driven Repayment for Student Loans,’ I investigate a form of student loan repayment known as income-driven repayment. This type of loan repayment has received a significant amount of political support recently in the U.S. and there are new proposals that would make it so that all future student loans would be repaid under income-driven plans. I use a detailed model that incorporates pre, intra, and post college decisions to explore how income-driven might change enrollment and debt accu-

mulation decisions, and how these changes might affect the required government subsidy to support the student loan program. I find that income-driven repayment would increase the cost of running the student loan program by 15% and that a significant portion of this cost comes from debt accumulated by those who are unlikely to graduate from college.

Table of Contents

List of Figures

List of Tables

List of Appendices

Introduction

Introduction

Chapter 1

Pareto weights as wedges in two-country models

1.1 Introduction

We explore the effects of recursive preferences and risk in an otherwise standard two-country exchange economy. We focus on the behavior of the (relative) Pareto weight, which characterizes consumption allocations across countries. When preferences are additive over time and across states, as they typically are, the Pareto weight is constant in frictionless environments. But when preferences are recursive and agents consume different goods, the Pareto weight can fluctuate even in frictionless environments. This variation in the Pareto weight acts like a wedge from the perspective of an additive model. Among the potential byproducts are changes in the behavior of consumption and the exchange rate.

The natural comparison is with models that use capital market frictions to account for the anomalous features of the standard model. Baxter and Crucini (1995), Corsetti et al. (2008), Heathcote and Perri (2002), Kehoe and Perri (2002), and Kose and Yi (2006) are well-known examples. The frictions in these papers might be viewed as devices to produce variation in the Pareto weight, which is then reflected in prices and quantities. In the language of Chari et al. (2007), variations in Pareto weights would appear as wedges in the frictionless model. The question for both approaches is whether these wedges are similar to those we observe when we confront frictionless models with evidence. Ultimately we would like to understand how the two approaches compare, but for now we're simply trying to understand the behavior of the Pareto weight in models with recursive preferences.

We build in an obvious way on earlier work with multi-good economies by Colacito and Croce (2013, 2011); Colacito et al. (2014); Kollmann (2015), and Tretvoll (2011, 2015, 2018). We show how their models work and introduce some modest extensions. We also build on the fundamental work on recursive risk-sharing by Anderson (2005), Borovička (2016), and Collin-dufresne et al. (2015). These papers study one-good worlds, and in that respect are simpler than work with multi-good international models, but they lay out the mathematical structure of recursive risk-sharing problems. The last paper in the list also describes an effective computational method that we adapt to our environment.

One byproduct is a clearer picture of what drives the dynamics of the Pareto weights. In many one-good economies, Pareto weights aren't stable. Eventually one agent consumes everything. One of the insights of Colacito and Croce (2013) is that home bias and imperfect substitutability between goods can produce stable processes for Pareto weights and consumption shares. That's true here, as well, but we also show how changes in risk aversion and intertemporal substitutability affect the dynamics of the Pareto weight. Relative to the additive case, increasing risk aversion or decreasing intertemporal substitution generates more persistence in the real exchange rate. Whether this persistence is welcome depends on one's view of the evidence.

Risk is a particularly interesting object in this context. Random fluctuations in the relative supply of foreign and domestic goods also affect demand — with recursive preferences — through their impact on future utility. As in many dynamic models, it's not clear how (if?) we might separate the concepts of supply and demand. A change in the conditional variance of future endowments, however, works only though the second channel; supplies (endowments) do not change. Risk affects allocations of goods through its impact on the Pareto weight without any direct impact on their supply.

All of these results are based on global numerical solutions to the Pareto problem. These solutions take much more computer time than the perturbation methods used in most related work, but they come with greater assurance that the solution is accurate even in states far from the mean of the distribution.

Notation and terminology. We use Latin letters for variables and Greek letters for parameters. Variables without time subscripts are means of logs. We use the term steady state to refer to the mean of the log of a variable. Thus steady state x refers to the mean value of $\log x_t$.

1.2 A recursive two-country economy

We study an exchange version of the Backus et al. (1994) two-country business cycle model. Like their model, ours has two agents (one for each country), two intermediate goods ("apples" and "bananas"), and two final goods (one for each country). Unlike theirs, ours has (i) exogenous output of intermediate goods, (ii) a unit root in productivity, (iii) recursive preferences, and (iv) stochastic volatility in productivity growth in one country. The first is for convenience. The second allows us to produce realistic asset prices. We focus on the last two, specifically their role in the fluctuations in consumption and exchange rates.

*P*references. We use the recursive preferences developed by Epstein and Zin (1989); Kreps and Porteus (1978), and Weil (1989). Utility from date t on in country j is denoted U_{jt}. We define utility recursively with the time aggregator V:

$$U_{jt} \quad = \quad V[c_{jt}, \mu_t(U_{jt+1})] \quad = \quad [(1-\beta)c_{jt}^{\rho} + \beta\mu_t(U_{jt+1})^{\rho}]^{1/\rho}, \tag{1.1}$$

where c_{jt} is consumption in country j and μ_t is a certainty equivalent function. The parameters are $0 < \beta < 1$ and $\rho \leq 1$. We use the power utility certainty equivalent function,

$$\mu_t(U_{jt+1}) \quad = \quad \left[E_t(U_{jt+1}^{\alpha})\right]^{1/\alpha}, \tag{1.2}$$

where E_t is the expectation conditional on the state at date t and $\alpha \leq 1$. Preferences reduce to the traditional additive case when $\alpha = \rho$.

Both V and μ are homogeneous of degree one (hd1). The two functions together have the property that if consumption is constant at c from date t on, then $U_{jt} = c$.

In standard terminology, $1/(1-\rho)$ is the intertemporal elasticity of substitution (IES) (between current consumption and the certainty equivalent of future utility) and $1-\alpha$ is risk aversion (RA) (over future utility). The terminology is somewhat misleading, because changes in ρ affect future utility, the thing over which we are risk averse. As in other multi-good settings, there's no clean separation between risk aversion and substitutability.

*T*echnology. Each country specializes in the production of its own intermediate good, "apples" in country 1 and "bananas" in country 2. In the exchange case we study, production in country j equals its exogenous productivity:

$$y_{jt} \quad = \quad z_{jt}. \tag{1.3}$$

Intermediate goods can be used in either country. The resource constraints are

$$a_{1t} + a_{2t} \quad = \quad y_{1t} \tag{1.4}$$

$$b_{1t} + b_{2t} \quad = \quad y_{2t}, \tag{1.5}$$

where b_{1t} is the quantity of country 2's good imported by country 1 and a_{2t} is the quantity of country 1's good imported by country 2.

Agents consume final goods, composites of the intermediate goods defined by the Armington aggregator h:

$$c_{1t} \quad = \quad h(a_{1t}, b_{1t}) \quad = \quad \left[(1-\omega)a_{1t}^{\sigma} + \omega b_{1t}^{\sigma}\right]^{1/\sigma} \tag{1.6}$$

$$c_{2t} \quad = \quad h(b_{2t}, a_{2t}) \tag{1.7}$$

with $0 \leq \omega \leq 1$ and $\sigma \leq 1$. The elasticity of substitution between the two intermediate goods is $1/(1-\sigma)$. The function h is also hd1.

We will typically use $\omega < 1/2$, which puts more weight on the home good in the production of final goods. This "home bias" in final goods production is essential. If

$\omega = 1/2$, the two final goods are the same. In this case, we have identical hd1 utility functions, and any optimal allocation involves a constant Pareto weight and proportional consumption paths.

*S*hocks. Fluctuations in this economy reflect variation in the productivities cum endowments z_{jt} and the conditional variance of the first one. Logs of productivities have unit roots and are cointegrated:

$$\begin{bmatrix} \log z_{1t+1} \\ \log z_{2t+1} \end{bmatrix} = \begin{bmatrix} \log g \\ \log g \end{bmatrix} + \begin{bmatrix} 1-\gamma & \gamma \\ \gamma & 1-\gamma \end{bmatrix} \begin{bmatrix} \log z_{1t} \\ \log z_{2t} \end{bmatrix} + \begin{bmatrix} v_t^{1/2} w_{1t+1} \\ v^{1/2} w_{2t+1} \end{bmatrix} \tag{1.8}$$

with $0 < \gamma < 1/2$. The only asymmetry in the model is the conditional variance of productivity. The conditional variance v of $\log z_{2t+1}$ is constant. The conditional variance v_t of $\log z_{1t+1}$ is AR(1):

$$v_{t+1} = (1-\varphi_v)v + \varphi_v v_t + \tau w_{3t+1}. \tag{1.9}$$

The innovations $\{w_{1t}, w_{2t}, w_{3t}\}$ are standard normals and are independent of each other and over time.

1.3 Features and parameter values

We describe some of the salient features of the model and the benchmark parameter values we use later on.

Features

*C*onsumption frontier. The resource constraints define a possibilities frontier for consumption. We picture several examples in Figure A.1 with $z_{1t} = z_{2t} = 1$. In each case, we compute the maximum quantity c_{1t} consistent with a given quantity c_{2t}, the resource constraints (1.4,1.5), and the Armington aggregators (1.6,1.7). The shape depends on the aggregator. When $\omega = 1/2$, the two final goods are the same and the tradeoff is linear.

When $\omega \neq 1/2$, the frontier is concave. The degree of concavity depends on the elasticity of substitution $1/(1-\sigma)$ in the Armington aggregator.

In a competitive equilibrium, the slope of the consumption frontier is (minus) the relative price of consumption in the two countries, $p_{2t}/p_{1t} = e_t$, the real exchange rate. From the figure, we can imagine variation in this price produced either by moving along the frontier or by changing the frontier through movements in the quantities of intermediate goods.

*M*arginal rates of substitution. Competitive equilibria and Pareto optima equate agents' marginal rates of substitution. With recursive preferences, the intertemporal marginal rate of substitution of agent j is

$$m_{jt+1} \quad = \quad \beta \left(\frac{c_{jt+1}}{c_{jt}} \right)^{\rho-1} \left(\frac{U_{jt+1}}{\mu_t(U_{jt+1})} \right)^{\alpha-\rho} . \tag{1.10}$$

The last term summarizes the impact of recursive preferences. If $\alpha = \rho$, preferences are additive and the term disappears. Otherwise anything that affects future utility can play a role in the marginal rate of substitution, hence in allocations. For example, a change in risk affects future utility which, in turn, alters optimal consumption allocations and market clearing prices. Persistence is critical here, because more persistent shocks have a larger impact on future utility. The discount factor β is similar: the larger it is, the greater the weight on future utility and the greater the impact on the marginal rate of substitution.

Note, too, the dynamics built into the recursive term. Its log is a risk adjustment plus white noise. The log of the numerator is

$$\log U_{jt+1} \quad = \quad E_t(\log U_{jt+1}) + \left[\log U_{jt+1} - E_t(\log U_{jt+1})\right],$$

the mean plus a white noise innovation. The log of the denominator is

$$\begin{aligned} \log \mu_t(U_{jt+1}) \quad &= \quad \alpha^{-1} \log E_t(e^{\alpha \log U_{jt+1}}) \\ &= \quad E_t(\log U_{jt+1}) + \alpha^{-1}\left[\log E_t(e^{\alpha \log U_{jt+1}}) - E_t(\alpha \log U_{jt+1})\right]. \end{aligned}$$

The term in square brackets is the entropy of U^{α}_{jt+1} and is positive. We multiply by α to get what we term the risk adjustment, which is negative if α is. The difference then is the innovation minus the risk adjustment. The innovation increases the volatility of the intertemporal marginal rate of substitution, which is the primary source of success in asset pricing applications.

The marginal rate of substitution (1.10) is measured in units of agent j's consumption good, whose price is p_{jt}. We refer to the relative price of the two consumption goods as the real exchange rate: $e_t = p_{2t}/p_{1t}$. The two marginal rates of substitution are then connected by $m_{2t+1} = (e_{t+1}/e_t)m_{1t+1}$. We'll derive this in the next section, but it should be evident here that the dynamics of the exchange rate reflect the marginal rates of substitution. If the two marginal rates of substitution are close to white noise, as we suggested, then the depreciation rate e_{t+1}/e_t has the same property.

*P*roductivity dynamics. One way to think about our log productivity process (1.8) is that their average is a martingale and their difference is stable. Denote half the sum and half the difference by

$$\begin{aligned} \log \bar{z} &= (\log z_1 + \log z_2)/2 \\ \log \widehat{z} &= (\log z_1 - \log z_2)/2. \end{aligned}$$

Then we can express the underlying productivities by $\log z_1 = \log \bar{z} + \log \widehat{z}$ and $\log z_2 = \log \bar{z} - \log \widehat{z}$. Equation (1.8) implies that half the sum,

$$\log \bar{z}_{t+1} = \log g + \log \bar{z}_t + (v_t^{1/2} w_{1t+1} + v^{1/2} w_{2t+1})/2,$$

is a martingale with drift. The difference,

$$\log \widehat{z}_{t+1} = (1-2\gamma) \log \widehat{z}_t + (v_t^{1/2} w_{1t+1} - v^{1/2} w_{2t+1})/2, \qquad (1.11)$$

is stable, which tells us $\log z_{1t}$ and $\log z_{2t}$ are cointegrated. Given the linear homogeneity of the model, changes in $\bar{z}$ affect consumption quantities proportionately, with no effect on their relative price, the exchange rate. Changes in relative productivity $\widehat{z}$, however, affect consumption quantities differentially and therefore affect the real exchange rate as well.

*P*areto problems. We compute competitive equilibria in this environment by finding Pareto optimal allocations and their supporting prices. In a two-agent Pareto problem, we maximize one agent's utility subject to (i) the other agent getting at least some promised level of utility (the promise-keeping constraint) and (ii) the productive capacity of the economy (the resource constraints and shocks). The Lagrangian for this problem is

$$\mathcal{L} \;=\; U_{1t} + \lambda_t (U_{2t} - \overline{U}) + \text{resource constraints and shocks},$$

with λ_t the multiplier on the promise-keeping constraint. If utility functions are strictly concave, this is equivalent to traditional Mantel-Negishi maximization of their weighted average,

$$\theta_{1t} U_{1t} + \theta_{2t} U_{2t},$$

with positive Pareto weights $(\theta_{1t}, \theta_{2t})$. Evidently λ_t in the previous problem plays the same role as θ_{2t}/θ_{1t}. We refer to λ_t as the Pareto weight, although in terms of the latter version we might call it the relative Pareto weight.

*T*ransforming utility. We find it convenient to use an hd1 time aggregator, but with additive preferences (the special case $\rho = \alpha$) it's more common to transform utility to $U^*_{jt} = U^{\rho}_{jt}/\rho$. With this transformation, equation (1.1) becomes

$$U^*_{jt} \;=\; (1-\beta) c^{\rho}_{jt}/\rho + \beta \big[E_t (U^{*\alpha/\rho}_{jt+1}) \big]^{\rho/\alpha}. \qquad (1.12)$$

When $\rho = \alpha$ this takes the familiar additive form.

The transformation also changes the look of derivatives. When we represent preferences with U_{jt}, marginal utility is

$$\partial U_{jt} / \partial c_{jt} \;=\; U^{1-\rho}_{jt} (1-\beta) c^{\rho-1}_{jt}.$$

When we use U^*_{jt}, marginal utility takes the simpler form

$$\partial U^*_{jt} / \partial c_{jt} \;=\; (1-\beta) c^{\rho-1}_{jt}.$$

We'll use this insight later on to simplify some of the expressions we get using the hd1 form of the time aggregator. This includes the Pareto weight, which is defined for a specific utility function.

Parameter values

We make only a modest effort to use realistic parameter values. The goal instead is to highlight the effects of recursive preferences and stochastic volatility with parameter values in the ballpark of those used elsewhere in the literature. We summarize these choices in Table A.1. The time interval is one quarter.

*P*references. We use $\rho = -1$ (implying an IES of one-half) and $\alpha = -9$ (implying risk aversion of 10). The former is a common value in business cycle modeling; Kydland and Prescott (1982), for example. The latter is widely used in asset pricing; Bansal and Yaron (2004) is the standard reference. The key feature of this configuration is that $\alpha - \rho < 0$. We set $\beta = 0.98$.

*T*echnology. The Armington aggregator plays a central role here, specifically the elasticity of substitution $1/(1-\sigma)$ between foreign and domestic intermediate goods. A wide range of elasticities have been used in the literature. Some earlier work used elasticities greater than one. Colacito and Croce (2011, 2013) and Kollmann (2015) use an elasticity of one. Heathcote and Perri (2002, Section 3.2), Tretvoll (2018, Table 1), and Tretvoll (2015, Table 3) suggest smaller values. We start with an elasticity of one ($\sigma = 0$), but consider other values, particularly when we explore the interaction of the elasticity and the dynamics of the Pareto weight.

Given a choice of σ, we set the share parameter ω like this. First-order conditions equate prices to marginal products:

$$\begin{aligned} p_{1t} &= c_{1t}^{1-\sigma}(1-\omega)a_{1t}^{\sigma-1} \\ p_{2t} &= c_{1t}^{1-\sigma}\omega b_{1t}^{\sigma-1} \end{aligned}$$

In a symmetric steady state with import share $s_m = b_1/(a_1+b_1)$ and relative price $p_{2t}/p_{1t} = 1$, the ratio of these two equations implies

$$\left(\frac{1-\omega}{\omega}\right) = \left(\frac{1-s_m}{s_m}\right)^{1-\sigma}. \tag{1.13}$$

We set $s_m = 0.1$. Given a value for σ, the import share nails down ω. One consequence of this calculation is that the parameter ω approaches one-half as σ approaches one (and the elasticity of substitution approaches infinity).

*S*hocks. The mean growth rate is $\log g = 0.004$: 0.4% per quarter. The number comes from Tallarini (2000, Table 4) and is estimated with US data. We set the persistence parameter γ that governs productivity dynamics equal to 0.1, which implies an autocorrelation of $1 - 2\gamma = 0.8$ for $\log \widehat{z}$. Rabanal et al. (2011, Table 5) estimate γ to be less than 0.01, which implies significantly greater persistence. The stochastic volatility process (1.9) is based on Jurado et al. (2015) as described in Backus et al. (2015, Section 5.3).

1.4 Solving the recursive Pareto problem

We compute a competitive equilibrium indirectly as a Pareto optimum, a standard approach in this literature. The equilibirum prices then show up as Lagrange multipliers on the resource constraints. Similar models have been studied by Colacito and Croce (2013), Kollmann (2015), and Tretvoll (2011).

Pareto problem

We solve the Pareto problem recursively with a slight change in notation: We represent utility of agent 1 by J, the value function, and the utility of agent 2 by U, without the "2" subscript. The state at date t is then the exogenous variables $s_t = (z_{1t}, z_{2t}, v_t)$ plus the utility promise U_t made to agent 2. The Bellman equation is

$$\begin{aligned} J(U_t, s_t) &= \max_{\{c_{1t}, U_{t+1}\}} V\{c_{1t}, \mu_t[J(U_{t+1}, s_{t+1})]\} \\ \text{s.t.} \quad & V\{c_{2t}, \mu_t(U_{t+1})\} \geq U_t \\ & \text{plus resource constraints and shocks.} \end{aligned}$$

The resource constraints include (1.4,1.5) for intermediate goods and (1.6,1.7) for final goods. The shocks follow (1.8,1.9).

We use Lagrange multipliers λ_t on promised utility and $(q_{1t}, q_{2t}, p_{1t}, p_{2t})$ on the resource constraints. The first-order conditions are then

$$
\begin{aligned}
c_{1t}: &\quad p_{1t} &=& \; J_t^{1-\rho}(1-\beta)c_{1t}^{\rho-1} && (1.14)\\
c_{2t}: &\quad p_{2t} &=& \; \lambda_t U_t^{1-\rho}(1-\beta)c_{2t}^{\rho-1} && (1.15)\\
a_{1t}: &\quad q_{1t} &=& \; p_{1t}c_{1t}^{1-\sigma}(1-\omega)a_{1t}^{\sigma-1} && (1.16)\\
b_{1t}: &\quad q_{2t} &=& \; p_{1t}c_{1t}^{1-\sigma}\omega b_{1t}^{\sigma-1} && (1.17)\\
a_{2t}: &\quad q_{1t} &=& \; p_{2t}c_{2t}^{1-\sigma}\omega a_{2t}^{\sigma-1} && (1.18)\\
b_{2t}: &\quad q_{2t} &=& \; p_{2t}c_{2t}^{1-\sigma}(1-\omega)b_{2t}^{\sigma-1} && (1.19)\\
U_{t+1}: &\quad J_t^{1-\rho}\beta\mu_t(J_{t+1})^{\rho-\alpha}J_{t+1}^{\alpha-1}J_{Ut+1} &=& \; -\lambda_t U_t^{1-\rho}\beta\mu_t(U_{t+1})^{\rho-\alpha}U_{t+1}^{\alpha-1}. && (1.20)
\end{aligned}
$$

Note, in particular, that equation (1.20) applies to promises U_{t+1} in every state at $t+1$. It's many equations, not just one.

The envelope condition for U_t is

$$J_{Ut} \;=\; -\lambda_t,$$

which we use to replace J_{Ut+1} with λ_{t+1} in (1.20).

*T*ransforming the Pareto weight. We can simplify the solution by transforming the Pareto weight λ_t. A Pareto weight is defined for specific a utility function; if we transform the utility function, we transform the Pareto weight along with it. The natural benchmark for the Pareto weight is the additive case, in which it's constant. Additive preferences are traditionally expressed using utility $U_t^* = U_t^\rho/\rho$ and $J_t^* = J_t^\rho/\rho$. The associated Pareto weight is

$$\lambda_t^* \;=\; \lambda_t U_t^{1-\rho}/J_t^{1-\rho}.$$

We refer to λ_t^* as the additive Pareto weight — or simply the Pareto weight.

With this substitution, we can clearly see the impact of recursive preferences. The first-order condition (1.20) becomes

$$\beta\left(\frac{J_{t+1}}{\mu_t(J_{t+1})}\right)^{\alpha-\rho}\lambda_{t+1}^* \;=\; \lambda_t^*\,\beta\left(\frac{U_{t+1}}{\mu_t(U_{t+1})}\right)^{\alpha-\rho}. \tag{1.21}$$

In the additive case, $\alpha = \rho$, this reduces to $\lambda^*_{t+1} = \lambda^*_t$ and the Pareto weight is constant. This is, of course, well known, but it's nice to know we're on the right track.

With recursive preferences, the Pareto weight need not be constant, although it's an open question how important this is quantitatively.

*C*onsumption and the exchange rate. The same transformation of the Pareto weight changes equations (1.14,1.15) to

$$(1-\beta)c_{1t}^{\rho-1}/p_{1t} \quad = \quad \lambda^*_t(1-\beta)c_{2t}^{\rho-1}/p_{2t}$$

or

$$p_{2t}/p_{1t} \quad = \quad \lambda^*_t(c_{2t}/c_{1t})^{\rho-1}. \tag{1.22}$$

If the prices of final goods are equal, as they are in a one-good world, the first equation tells us to equate weighted marginal utilities across agents.

In the second equation, the left side is the real exchange rate e_t. The right side is the product of the Pareto weight and the consumption ratio. In the additive case, the Pareto weight is constant and we have a linear relation between the logs of the real exchange rate and the consumption ratio. In the data, there is little sign of such a relation. See, among many others, Backus and Smith (1993); Chari et al. (2002); Corsetti et al. (2008); Colacito and Croce (2013); Kollmann (1995), and Tretvoll (2011). We might say that the data suggests a wedge between the price and consumption ratios that is represented in the recursive model by the Pareto weight.

If we further simplify the model to have a single good, then the real exchange rate is one in all states. In the additive case, λ^*_t is constant and equation (1.22) tells us that the ratio of consumptions is also constant. Any variation in consumption by one agent is exactly mirrored by the other. This is, of course, counterfactual, and one of the standard anomalies of international business cycle models.

*M*arginal rates of substitution. The Pareto problem equates marginal rates of substitution, but since the agents consume different goods this involves the relative price

$e_t = p_{2t}/p_{1t}$. The ratio of equation (1.22) at dates t and $t+1$ is

$$\beta \left(\frac{c_{1t+1}}{c_{1t}} \right)^{\rho-1} \left(\frac{e_{t+1}}{e_t} \right) \quad = \quad \left(\frac{\lambda^*_{t+1}}{\lambda^*_t} \right) \beta \left(\frac{c_{2t+1}}{c_{2t}} \right)^{\rho-1} .$$

Combining this with (1.21) gives us

$$\beta \left(\frac{c_{1t+1}}{c_{1t}} \right)^{\rho-1} \left(\frac{J_{t+1}}{\mu_t(J_{t+1})} \right)^{\alpha-\rho} \left(\frac{e_{t+1}}{e_t} \right) \quad = \quad \beta \left(\frac{c_{2t+1}}{c_{2t}} \right)^{\rho-1} \left(\frac{U_{t+1}}{\mu_t(U_{t+1})} \right)^{\alpha-\rho}$$

as noted earlier. Note the role of $\alpha - \rho$. If the difference is zero, the recursive term drops out. Otherwise the sign of the impact of future utility depends on the sign of $\alpha - \rho$.

1.5 Properties of the exchange economy

We compute an accurate global solution to the scaled Pareto problem by methods described in Appendix A.1. We describe its properties here. We start with the Pareto weight, then go on to explore the dynamics of the Pareto weight, the connection between consumption and the real exchange rate, and the responses of consumption and other variables to changes in various state variables.

*T*he Pareto frontier. One of the outputs of the numerical solution is the value function J, a function of promised utility U and the exogenous state variables. Given values for the exogenous state variables, this gives us the Pareto frontier: the highest utility of agent 1 (J) consistent with a given level of utility for agent 2 (U) and the productive capacity of the economy.

We describe the Pareto frontier in Figure A.2 with state variables $z_{1t} = z_{2t} = \widehat{z}_t = 1$ and $v_t = v$. The outer curve in the figure is the consumption frontier, which echoes Figure A.1. The inner curve is the Pareto frontier. We see that it has much the same shape. It's inside the consumption frontier largely because of risk: utility is below consumption because risk reduces utility. If we increase risk aversion to 50 ($\alpha = -49$, not shown), it shifts in further.

Changes in the state variables change both frontiers. Movements in productivity and output shift the frontiers — both of them — in and out ($\bar{z}_t$, which affects the two intermediate goods proportionately) or twist them ($\widehat{z}_t$, which affects the two goods differently). Changes in risk twist the Pareto frontier, since risk affects the two goods differently, but not the consumption frontier, since it has no effect on quantities of intermediate goods.

*D*ynamics of the Pareto weight. We see the impact of recursive preferences in Figure A.3, where we graph $\log \lambda_t^*$ against time for a (very long) simulation of the model. The flat horizontal line refers to the additive case ($\alpha = \rho = -1$). As we know, the Pareto weight doesn't change in this case. The other line refers to the recursive case, and we see clear variation in the Pareto weight. We also see that the variation is both large and very persistent.

This touches on a question that's been discussed extensively: Is the Pareto weight stable, or does one agent eventually consume everything? Anderson (2005) and Borovička (2016) document some of the difficulties of establishing stability in similar one-good settings. Colacito and Croce (2011) prove stability in a two-good world with elasticities of substitution between goods and over time equal to one. Colacito and Croce (2013), Kollmann (2015), and Tretvoll (2011, 2015, 2018) solve similar models numerically and report that the solutions are stable. We also find that they're stable, but extremely persistent.

We get a sense of how stability works in Figure A.4, where we plot the expected change in the log Pareto weight against its level. The exogenous state variables here have been set equal to their means. In the additive case, the expected (and actual) change is zero. The log Pareto weight is a martingale with no variance. With greater risk aversion, mean reversion becomes evident. If the Pareto weight is below its steady state value of one ($\log \lambda_t^* = 0$), it's expected to increase. If above, it's expected to decrease. The effect is stronger when we increase risk aversion to 50. There is also an evident nonlinearity in the solution, as there is in Colacito and Croce (2011, Figure 5), but most of it occurs in regions of the state space we rarely reach.

The elasticity of substitution between foreign and domestic intermediate goods also plays a role in persistence. See Figure A.5. With smaller values, mean reversion is slower. And with larger values, it's faster. As the elasticity increases, the line flattens out and we approach the one-good world with a constant Pareto weight.

The intertemporal elasticity of substitution also has an effect, but with the numbers we've chosen the effect is smaller. See Figure A.6. Evidently smaller values of ρ, and larger values of the IES $[1/(1-\rho)]$, lead to flatter lines.

*C*onsumption and exchange rate. We noted earlier that the relation between the log consumption ratio $[\log(c_{2t}/c_{1t})]$ and the log of the real exchange rate $[\log e_t = \log(p_{2t}/p_{1t})]$ is mediated by the log Pareto weight $(\log \lambda_t^*)$. See equation (1.22). In the additive case, the Pareto weight is constant and we have a perfect linear relationship between the two variables. We see exactly this in the line in Figure A.7.

The scatter of points in the same figure represents the recursive case, where the Pareto weight acts like a wedge from the perspective of the additive model. With our numbers, the variation in the Pareto weight is enough to change a negative correlation of minus one between the consumption ratio and exchange rate to a slight positive correlation. Colacito and Croce (2013), Kollmann (2015), and Tretvoll (2011) show the same. If we increase risk aversion $1-\alpha$ to 50, the correlation becomes strongly positive. In the recursive model, we can produce any correlation we like by varying the risk aversion parameter.

Recursive preferences also have an impact on exchange rate dynamics as the persistence in the Pareto weight is reflected in the real exchange rate. We see in Figure A.8 that the additive model is much less persistent: The half-life (where the autocorrelation function equals one-half) is about a year. By five years, the autocorrelation is essentially zero. Exchange rate dynamics reflect, in this case, the modest persistence of relative productivity $\widehat{z}_t$. With recursive preferences, the exchange rate is much more persistent. In fact with these parameter values, it's virtually a martingale. We can reproduce any level of persistence we like by varying risk aversion between the two cases. There's a range of opinion, summarized nicely by Crucini and Shintani (2008), about how much persistence we need for the model to be realistic. The larger point is that recursive preferences are a device that can deliver persistence in real exchange rates and macroeconomic variables in general. It's what an older literature would call a propagation mechanism.

*R*esponses to productivity and volatility shocks. We get another perspective on the model's dynamics from impulse responses. Starting at the steady state, we increase one of the exogenous state variables by one standard deviation at date one, simulate the model for several periods, and compute the mean dynamics of all the variables in the model. This goes somewhat beyond traditional impulse responses in linear models in which the subsequent innovations are turned off.

In Figure A.9 we describe responses to an increase in (the log of) relative productivity $\widehat{z}_t$. The effect on future values of $\log \widehat{z}_t$ declines at a constant rate as described by equation (1.11). Neither average productivity $\bar{z}_t$ nor volatility v_t change, so this implies an increase in $\log z_{1t}$ and an equal decrease in $\log z_{2t}$. The quantity of apples goes up, and the quantity of bananas goes down. Because of home bias, consumption goes up in country 1 (the apple eaters) and down in country 2 (banana eaters). The exchange rate rises as scarce bananas become more expensive.

All of this would be true in the additive case as well. What's different is the response of the Pareto weight. It goes up as we compensate the agent in country 2 with promises of higher future consumption. This effect eventually wears off, but it does so very slowly.

In Figure A.10 we describe the responses to an increase in volatility v_t. Here there's no change in the quantities of intermediate goods. In the additive case, there would be virtually no effect. In the recursive case, utility falls, but it falls more for country 1 because of its home bias in favor of the good whose supply has become riskier. The social planner responds by decreasing the Pareto weight on agent 2. Consumption therefore rises in country 1 and falls in country 2. The real exchange rate falls. This is entirely a demand-side effect. By increasing the weight on agent 1, the demand for apples goes up and the demand for bananas goes down. The magnitudes are small, but it's an interesting effect that we would like to explore further in a production economy, where supply can respond to changes in market conditions.

1.6 Open questions

We have documented the behavior of the Pareto weight in a relatively simple environment. We showed, as others have, that recursive preferences can change the quantitative properties of the model in useful ways. The behavior of exchange rates, in particular, is much different from the additive case.

Beyond this, we are left with a number of open questions:

- What parameters govern the persistence of the Pareto weight? Can we be more precise about the impact of risk aversion and intertemporal substitution on the dynamics of the Pareto weight? Of the substitutability of foreign and domestic goods in the Armington aggregator?
- How would this change in a production economy? Production offers opportunities to respond to changes in exogenous variables, particularly changes in risk. If production in one country becomes more risky, do we shift production to the other country? Does capital flow to the less risky country? Are the magnitudes plausible?
- How do changes in the Pareto weight generated by frictions and recursive preferences compare? Are they similar or different? Are the two mechanisms complements or substitutes?

Chapter 2

Global Solution Methods for Macroeconomic Models

2.1 Introduction

There is a growing interest among economists in global nonlinear solution methods. Value function discretization is the most well-known example of a global nonlinear solution method, but there are a variety of other iterative and Euler-equation based methods that differ in the way they approximate, interpolate, integrate and construct nonlinear solutions. These methods solve economic models on grids of points that cover a relevant area of the state space, and they generally produce more accurate solutions than perturbation methods which typically build approximations around a single point.[1] Also, global methods are useful for solving some models in which perturbation methods are either not applicable or their applications are limited.[2] Finally, the recent developments of numerical techniques

[1] The accuracy of local perturbation solutions can decrease rapidly away from steady state even in relatively smooth models; see Kollmann et al. (2011). In models with stronger nonlinearities, such as new Keynesian models, approximation errors can reach hundreds of percent under empirically relevant parameterizations; see Judd et al. (2017). In models of labor search, a global solution is important for accurate capturing even first moments of equilibrium dynamics; see Petrosky-Nadeau and Zhang (Forthcoming)

[2] For example, perturbation methods are not well suitable for analyzing sovereign default problems (e.g., Arellano (2008)) and portfolio choice problems (e.g., Hasanhodzic and Kotlikoff (2013)). There are perturbation-based methods that can deal with occasionally binding constraints but they are limited to the first-order approximation. For example, see Laséen and Svensson (2011) and Guerrieri and Iacoviello (2015).

for dealing with high-dimensional data has greatly extended the class of models that can be studied with global solution methods. In particular, it is now possible to construct global nonlinear solutions to economic models with dozens of state variables that are intractable with conventional global solution methods (such as value-function discretization) due to the curse of dimensionality.[3]

However, global nonlinear solution methods are more difficult to automate than perturbation methods, and their implementation requires more substantial programming from researchers. In particular, there is still no consensus in the profession on what software to use for implementing global solution methods, unlike for perturbation methods where a common choice is the Dynare or IRIS platforms. Because of this, an important aspect of the implementation of a global solution method is the choice of programming language to use. Traditionally, Matlab is used in economics, although some researchers have also used Fortan and C. Recently, Python and Julia have begun to see a more widespread use in the economics literature.[4]

In this paper, we pursue two goals: First, we provide a comparison between solution methods implemented in Matlab, Python and Julia in the context of two popular applications: a neoclassical growth model and a new Keynesian model. Our goal is to help economic researchers choose the programming language that is best suited to their own situation, and, if needed, help them transition from one programming language to another. The readers can see and compare how the same algorithm is implemented in different languages and, as a result, will understand some of the unique aspects of each language and choose the one which best fits their needs.[5] The implementation, structure and number of lines of code are similar across all three languages.

Second, we provide a carefully documented collection of routines for Matlab, Python

[3] See Maliar and Maliar (2014) for a survey of numerical techniques that are tractable in problems with high dimensionality and see Lepetyuk et al. (2017) for an example of constructing a global nonlinear solution to a large-scale projection model of the Bank of Canada.

[4] For example, Python and Julia are used in the *Quantitative economics* lecture site of Tom Sargent and John Stachurski. Python is used to produce the *Heterogeneous Agents Resources and toolKit* (HARK) by the team led by Christopher Carroll.

[5] Additionally, we provide carefully elaborated notebooks with a description of our code and algorithms on the QuantEcon Notebook site…

and Julia that can be used for constructing global nonlinear solutions with a special emphasis on modern techniques for problems with high dimensionality. In particular, our code includes routines for constructing random and quasi-random grids, low-cost monomial integration methods, approximating functions using complete polynomials, as well as routines for computing the numerical accuracy of solutions. Much of this code is generic in a way which allows it to be easily portable to other applications including problems with kinks and occasionally binding constraints. Our examples are solved using a variety of solution techniques including: conventional policy function iteration, conventional value function iteration, an Euler equation method, the endogenous grid method of Carroll (2006), and several variants of the envelope condition method of Maliar and Maliar (2015).

Aruoba and Fernández-Villaverde (2015) is closely related to our work. In their paper, the authors compare 18 different programming languages by implementing the same algorithm – value-function discretization – to solve the stochastic neoclassical growth model and by reporting the CPU time needed to solve the model. They find that the choice of a programming language plays an important role in computation time: the fastest language in their study (C++) solves the model over 450 times faster than the slowest language (R). The present work extends Aruoba and Fernández-Villaverde (2015) by analyzing a variety of algorithms and also by solving a larger (in terms of the number of state variables) model. Similar to Aruoba and Fernández-Villaverde (2015), we find that the choice of a programming language may have important effects in terms of speed, however, we do not observe such huge differences across the three languages considered. In addition to speed comparisons, we also discuss the key features of each language that might be useful for economists.

We build on the existing Matlab codes accompanying Arellano et al. (2016) and Maliar and Maliar (2015). We improve on these codes, and provide new routines that have not been analyzed previously in the literature. First, for the neoclassical growth model, we implement seven different methods in all three programming languages, whereas Arellano et al. (2016) is limited to Matlab implementation of the envelope condition method. Furthermore, for the new Keynesian model, we modify the method proposed in Maliar and Maliar (2015) to operate on random and quasi-random grids covering a fixed hypercube, instead of operating on the high probability area of the state space. The Matlab code developed in the present paper achieves an almost a 60-time speed up over the original code in Maliar and Maliar (2015), while still producing highly accurate solutions. Our code is sufficiently fast to be used for the estimation of a moderately-large new Keynesian model with 8 state variables. It takes us just a few seconds to construct the solution, including the model with active zero lower bound on the nominal interest rate.

Our overall experience with each language was comparable, though it is useful to recognize that each language has it specific strengths. Both Julia and Python are open-source which facilitates transparency and reproducibility. For reasons to be discussed later, Julia outperforms both Matlab and Python in algorithms that require numerical solvers or optimizers. Python benefits from an active community and strong package ecosystem which makes finding the right tools easy. Finally, Matlab benefits from extensive documentation, technical support, and various built in toolboxes. Ultimately, the best choice of programming language depends on various features including: model size, solution complexity, ease of writing, and co-author preferences. Regardless of the programming language chosen, our experience shows that transitioning between each of the three languages we consider should not require a substantial learning effort.

The rest of the paper is organized as follows. In Section 2, we describe the three languages, Matlab, Python and Julia. In Section 3, we outline seven algorithms we implement to solve a variant of the standard neoclassical stochastic growth model. In Section 4, we present an algorithm for solving a medium-scale new Keynesian model with a zero lower bound on nominal interest rates and describe the results of our analysis. Finally, in Section 5 we conclude.

2.2 Programming languages in economics

In this section, we provide a brief description of the three programming languages discussed in this paper, MATLAB, Python and Julia. We provide step-by-step instructions of how to install Julia and Python and a description of the accompanying software in our corresponding notebooks; see QuantEcon Notebook site.

Matlab

In our experience, Matlab is the programming language that most economists currently use. The most compelling reason for this is that it is a “batteries included”solution. What we mean by “batteries included”is that Matlab provides a code editing and execution environment, plotting capabilities, and a vast catalog of pre-written numerical routines. For example, in this project, we were able to leverage built-in routines to construct Sobol sequences (quasi-Monte Carlo grids) and for doing numerical optimization. Another benefit is that, because so many other economists already use Matlab, collaborating on research code with others is efficient. There are two main examples of this. The first is that Dynare, which is mostly used via it’s Matlab interface, has become widely used across academia and policy institutions facilitating communication and sharing of models. The second is that because many algorithms and routines have been previously written in Matlab, it should be easy for researchers to adapt these pieces of code to new projects.

Matlab, however, is not a perfect language for economists. Some of its downsides are:

- *Commercial.* As a commercial product, in order to use Matlab users must first obtain a non-free license. This can be costly if an institution does not already have a license available for use.
- *Closed source.* Users cannot look at the Matlab source code to study (or change) the implementation of built-in routines.
- *Restricted parallel programming possibilities (due to limited license availability).* As computer hardware becomes more powerful and computer resources in the cloud

become more accessible, economists are increasingly implementing parallel versions of their algorithms. While basic parallelism is convenient in Matlab, running code at scale requires one license per process. This increases the cost of large scale computations. [6]

- *Limited notion of community packages.* Apart from the official Matlab toolboxes, Matlab does not have a well-developed system of add-on packages.[7] Using external libraries requires users to download and place the files in the same directory as the scripts they are running or to manage the path Matlab searches when calling functions. Although not insurmountable, this friction discourages good software development by making it difficult to build and reuse modular libraries.
- *Mainly a numerical programming language.* Tasks outside of this purview can be more difficult than in a more general programming language.

Python

While Matlab has long been widely used in economics, Python has been steadily gaining users in economics during recent years. Some of Python's most compelling selling points are as follows:

- *A mature and rich package ecosystem.* At the time of writing, Python has 112,232 registered packages that perform a variety of tasks and can be easily downloaded and installed.
- *Advanced data manipulation and numerics libraries.* For example, `pandas` is a very powerful library containing data structures and algorithms for doing data analysis. `numpy` and `scipy` are the base of array-based numerical computing in Python and

[6] As an example, if one wanted to run a Matlab program on 16 optimized processors on the Amazon Compute Cloud, then in addition to paying $0.80/hour for the processor time, it would cost an additional $0.20/hour per processor to license each processor bringing the total from $0.80/hour to $4.00/hour. For details see Matlab's pricing site and Amazon's pricing site.

[7] Matlab does however host an open file exchange called Matlab Central. Matlab Central has 365,000 current contributors and code that is available to do many tasks that are not included in the base Matlab package, but they do not provide the same package management services that are available within Python or Julia.

provide linear algebra, optimization, and other foundational numerical algorithms. `matplotlib` is the main plotting library.

- *Many users and ample learning resources.* Python is one of the top five most widely used programming languages in the world. This large user base is beneficial to economists because of the libraries mentioned above and the sheer volume of learning materials and resources available online for learning Python. As one point of data, at the time of writing there are 696,984 questions tagged with Python on `http://stackoverflow.com`[8], while there are only 68,403 for Matlab and 2,659 for Julia.
- *Wide adoption in the private sector.* This is beneficial because industry resources are being applied to enhance the language and create powerful tools economists can leverage in their computational code.
- *Open source.* This means that the source code is publicly available on the internet and it is 100 percent free to view, install, distribute and use.

Python is not perfect, however. Some of its shortcomings are:

- *External libraries.* External libraries are needed for efficient numerical computation. This means that users need to identify which libraries to use, know how to install them properly, and load them into their codes. Fortunately this is not as difficult as it sounds thanks to scientific python distributions like Anaconda that come with Python and all the most common scientific packages installed and pre-configured.
- *Slower than other languages for certain types of algorithms.* This can often be circumvented by vectorizing operations (as in Matlab) or by leveraging tools such as numba or Cython to compile Python code into a fast and efficient machine code. In many cases this is a suitable approach, but it does require extra effort on behalf of the programmer. In other cases, it is difficult or impossible to express only the slower parts of an algorithm using these tools.

[8] This is a popular question-and-answer website for programming.

Julia

The most recently released language of the three is Julia. While Julia has not yet reached a stable 1.0 release, many enthusiastic economists are starting to pick it up because of the benefits that it offers. The list of Julia's strengths is extensive and includes the following:

- *All of Julia is based around just-in-time (JIT) compilation*[9]. Since all Julia code is compiled before it is executed, Julia has the potential to run at the same speed as languages like C or Fortran.[10] [11]
- *Most of Julia is written in Julia.* This offers at least two main benefits: (1) core language features and constructs are written and defined using the same tools available to user code; (2) users can look to the implementation of Julia itself to see best practices and tips to make their own code better. One consequence of this that is often surprising to Matlab or Python users is that a Julia user can write a routine in Julia that performs as well, or often better, than implementations in common libraries. In other words, Julia users do not need to reach to a library for every operation, because writing the subroutine in Julia can be at least as performant.
- *Flexibility.* Of the three languages that we analyze, Julia is the most expressive and flexible. It features advanced computer science concepts such as meta-programming (see below) and multiple dispatch (a choice of which method to execute when a function is applied; such a choice is made using all of a function's arguments, not just the first one).
- *Similarity to Matlab.* Julia's syntax is intentionally similar to that of Matlab. In fact, in many instances, copying and pasting Matlab code produces the same result in Julia as in Matlab. This was an intentional decision intended to allow Matlab users to transition seamlessly from Matlab to Julia.

[9] Just-in-time compilation means that a function is compiled the first time it is called and all subsequent calls to that function are calls to faster.

[10] See `https://julialang.org/benchmarks/` for some example benchmarks.

[11] There has been work to add just-in-time compilation to both Matlab and Python, but, because it isn't native to either language, it does not work as generically as it does in Julia.

- *Built in parallel and multi-threaded programming constructs.* Having support at the language level for both distributed and shared memory parallel processing means running code in serial or in parallel is often a one word change to the user's code. Also, because Julia is open source like Python, its programs can scale up to run on arbitrarily many processes without incurring the additional costs from licensing.
- *Meta-programming.* One relatively advanced and compelling feature that is unique to Julia (amongst the languages considered here) is meta-programming. A language that allows meta-programming treats the code that is executed as data that can be manipulated by the program itself. This means users can write code that writes the code that will be run. The most complete example of how we leveraged this feature for this project was in implementing routines to compute basis functions for complete polynomial. In Python and Matlab, our implementation was quite naive. We allow for a complete polynomial of degree between two and five and write out one `for` loop for each degree. The structure of each loop is identical. In contrast, the Julia's version of this routine allows an arbitrary degree complete polynomial. To achieve this, we leverage Julia's meta-programming capabilities, receive the desired degree of complete polynomial from the user and then instruct Julia to emit code that contains that many for loops – each having the same structure as the hand-written loops in our Python and Matlab versions. This results in the complete polynomial Julia routines having less code written by us and being more flexible and generic.

Despite many promising features, there are still some shortcomings to Julia. These include:

- *Lack of stability.* Julia is still very young, it was first made public in 2011 and has not reached a stable 1.0 release.[12]
- *Slow compile times.* Because the JIT compiler aggressively optimizes the machine code for each function that is executed, it can feel sluggish when doing rapid iterative

[12] The Julia team anticipates releasing version 1.0 in Q2 2018.

development (make a change, run code, make a change, run code, etc…) because you pay the fixed compile cost after each edit.

- *Julia must be able to infer a type for each variable in a function* in order to emit fast code. This is not usually a problem, but sometimes if you are careless then the code will run slower than it should — though, even in the worst case, it is still often as fast as Matlab or Python.
- *The package system is still emerging.* Many of the packages throughout the Julia community are surprisingly advanced given that Julia has still not reached a stable release, but they are usually not as polished as what you would find in the Matlab sponsored toolboxes or the best Python packages. That being said, we have not found package availability to be prohibitive for our work.[13]
- *Lack of tooling.* Unlike the more mature languages Matlab and Python, Julia currently lacks polished developer tools like an integrated development environment (IDE) or debugger. These are actively being worked on and will hopefully emerge soon after the Julia 1.0 release.

Syntax comparison

Julia and Matlab have many syntax elements in common. For example, if one wants to perform matrix multiplication of two matrices A and B then one can simply write `A*B`, but if one wants to perform element wise multiplication then one writes `A.*B`. Additionally, blocks are started with the same words (`for`, `if`, `while`) and terminated with the word `end`. These similarities make it very easy to translate code from Matlab to Julia and vice-versa. The one major difference between Julia and Matlab is that Julia uses square brackets to index into arrays (`A[i, j]`) while Matlab uses parenthesis (`A(i, j)`).

On the other hand, Python differs slightly in its conventions. In Python `A*B` represents element-wise multiplication between matrices while `A@B` represents matrix multipli-

[13] When routines are not available, it often isn't much extra work to code up an implementation by hand; see the discussion above.

cations.[14] Additionally, rather than depend on `end` to denote the end of blocks, Python forces each new block to be indented four spaces from the previous one. Despite these differences, we found that syntax differences did not present a large obstacle for writing code in all three languages.

2.3 Comparison using a neoclassical growth model

The neoclassical growth model is often used as a benchmark to measure the effectiveness of a solution method. We keep with this tradition and analyze the stochastic neoclassical growth model with seven alternative solution methods, which are outlined in Arellano et al. (2016). Each method is implemented in Julia, Matlab, and Python which allows us to see how the performance of different solution methods may vary by programming language.

The model

We consider a dynamic programming problem of finding the value function, V, that solves the Bellman equation,

$$V(k,z) = \max_{c,k'} u(c) + \beta E\left[V\left(k',z'\right)\right] \tag{2.1}$$

$$\text{s.t.} \quad k' = (1-\delta)k + zf(k) - c \tag{2.2}$$

$$\ln z' = \rho \ln z + \varepsilon', \qquad \varepsilon' \sim \mathcal{N}\left(0,\sigma^2\right), \tag{2.3}$$

where k, c and z are capital, consumption and productivity level, respectively; $\beta \in (0,1)$; $\delta \in (0,1]$; $\rho \in (-1,1)$; $\sigma \geq 0$; the utility and production functions, u and f, respectively, are strictly increasing, continuously differentiable and strictly concave. The primes on variables denote next-period values, and $E\left[V\left(k',z'\right)\right]$ is an expectation conditional on state (k,z).

[14] The use of `@` for matrix multiplication was added in Python version 3.5. Readers who have seen or written Python code written for Python versions earlier than version 3.5 might have come across `A.dot(B)` or `np.dot(A, B)` instead.

Optimality conditions. The first order condition (FOC) and envelope condition (EC) of the problem (2.1)-(2.3), respectively, are

$$u'(c) = \beta E\left[V_1\left(k',z'\right)\right] \tag{2.4}$$

$$V_1(k,z) = u'(c)\left[1-\delta+zf'(k)\right]. \tag{2.5}$$

By combining (2.4) and (2.5), we obtain the Euler equation

$$u'(c) = \beta E\left[u'\left(c'\right)\left[1-\delta+z'f'\left(k'\right)\right]\right]. \tag{2.6}$$

Parameterization and implementation details. We parameterize the model (2.1)-(2.3) by using a Constant Relative Risk Aversion (CRRA) utility function, $u(c) = \frac{c^{1-\gamma}-1}{1-\gamma}$, and a Cobb-Douglas production function, $f(k) = Ak^{\alpha}$. We choose parameters to be set at $A = \frac{1/\beta-(1-\delta)}{\alpha}$, $\alpha = 1/3$, $\beta = 0.99$, $\delta = 0.025$, $\rho = 0.95$ and $\sigma = 0.01$. As a solution domain, we use a rectangular, uniformly spaced grid of 10×10 points for capital and productivity between 0.9 and 1.1 for both variables.

We integrate by using a 10-node Gauss-Hermite quadrature rule and approximate the policy and value functions by using complete ordinary polynomials up to degree 5. As an initial guess, we use a linear approximation to the capital policy function. To solve for polynomial coefficients, we use fixed-point iteration.

All computations are performed using Julia v0.6.0, Matlab version 9.2.0 (R2017a), and Python 3.6 on a MacBook Pro with a 2.7 GHz Intel Core i7 processor and 16 GB of RAM. For Julia and Python, the particular package versions can be found in the corresponding notebooks submitted on QuantEcon's Notebook site[15].

[15] The Matlab notebook, Python notebook, Julia notebook

Value iterative methods

We analyze three value iterative algorithms for solving the model: (1) conventional value function iteration method analyzed in e.g., Aruoba and Fernández-Villaverde (2015) ; (2) a variant of envelope condition method (ECM) of Maliar and Maliar (2013) that finds a solution to Bellman equation by iterating on value function; and (3) endogenous grid method (EGM) of Carroll (2006). We present a brief description of each algorithm below. A detailed description of each algorithm is included in Appendix A.

Conventional Value Function Iteration

Conventional VFI constructs the policy rule for consumption by combining FOC (2.4) and budget constraint (2.2):

Algorithm 1. Conventional VFI

```
Given $V$, for each point $(k,z)$, define the following recursion:
    i). Numerically solve for $c$ satisfying $u'(c) = \beta E[V_1((1-\delta)k + zf(k) - c, z')]$.
    ii). Find $k' = (1-\delta)k + zf(k) - c$.
    iii). Find $\widehat{V}(k,z) = u(c) + \beta E[V(k',z')]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.
```

Envelope Condition Method

ECM, proposed in Maliar and Maliar (2013), finds consumption from envelope condition (2.5) and budget constraint (2.2). In this specific model, the consumption function can be constructed analytically:

Algorithm 2. ECM-VF

```
Given $V$, for each point $(k,z)$, define the following recursion:
    i). Find $c = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf'(k)}\right]$.
    ii). Find $k' = (1-\delta)k + zf(k) - c$.
    iii). Find $\widehat{V}(k,z) = u(c) + \beta E[V(k',z')]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.
```

Endogenous Grid Method

Finally, EGM of Carroll (2006) constructs a grid on (k',z) by fixing the future endogenous state variable k' and by treating the current endogenous state variable k as unknown; see also Barillas and Fernández-Villaverde (2007) for a discussion of the EGM method. Since k' is fixed, EGM computes $E[V_1(k',z')]$ up-front and thus can avoid costly interpolation and approximation of expectation in a root finding procedure.

Algorithm 3. EGM of Carroll (2005)

```
Given $V$, for each point $(k',z)$, define the following recursion:
    i). Find $c = u'^{-1}\{\beta E[V_1(k',z')]\}$.
    ii). Solve for $k$ satisfying $k' = (1-\delta)k + zf(k) - c$.
    iii). Find $\widehat{V}(k,z) = u(c) + \beta E[V(k',z')]$.
Iterate on i)-iii) until convergence $\widehat{V} = V$.
```

In Step ii) of EGM, we still need to find k numerically. However, for the studied model, Carroll (2006) shows a change of variables that makes it possible to avoid finding k numerically on each iteration (except of the very last iteration). In order to streamline our code, we do not use this change of variables so the running time of our version of the EGM method will be considerably larger than for the version of the method in Carroll

(2006). However, this fact does not distort our comparison analysis since we use the same implementation in all three languages.

Comparison results of Matlab, Python and Julia for value iterative methods

The results from our comparison of the three iterative methods are in Table B.1. Conventional VFI is the most time consuming in all three languages because it requires us to find the root of (2.4) for each (k,z). Given (k,z), each evaluation of equation (2.4) requires the computer to compute conditional expectation by interpolating over the $t+1$ values. Numerical root finders must do this repeatedly. The reason EGM performs better than VFI is that it is easier to solve equation (2.4) with respect to c given (k',z) than to solve it with respect to c given (k,z) because we only need to evaluate conditional expectation once if we know k'. The ECM method requires no numerical solver which is why it is so efficient relative to the other two methods.[16]

Julia produces solutions between 5 and 50 times faster than Matlab and Python for VFI and EGM. The reasons Julia performs so much better on these two algorithms is because the entire language is built on JIT compilation. This means that the repeated calls to the Julia objective function are cheap (relative to an interpreted language like Matlab or Python) because these subsequent function calls execute already compiled machine code.

Our results for the ECM show similar performance for all three languages. The main reason for this is that the method does not require a non-linear solver to compute the policy for consumption and computations can be vectorized. The vectorization puts the languages on more similar footing because they all end up calling out to the same high performant BLAS routines implemented in C or Fortran.

[16] A version of the envelope condition argument is used in Achdou et al. (2017) to construct a new class of fast and efficient numerical methods for solving dynamic economic models in continuous time.

Policy iterating methods

We analyze four algorithms that construct policy functions for the standard neoclassical stochastic growth model: Algorithm 4 is a conventional policy iteration (PI) method, e.g., Santos and Rust (2004); Algorithm 5 is a variant of ECM that implements policy iteration; Algorithm 6 is a version of ECM that solve for derivative of value function instead of value function itself; Algorithm 7 an Euler equation algorithm. Again we give a brief description of each algorithm but a more detailed description of these algorithms is provided in Appendix A.

Conventional Policy Iteration

Conventional policy iteration constructs a solution to the Bellman equation by iterating on the consumption function using FOC (2.4).

Algorithm 4. Conventional PI

```
Given $C$, for each point $(k,z)$, define the following recursion:
    i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$
    ii). Solve for $V$ satisfying $V(k,z) = u(C(k,z)) + \beta E[V(K(k,z),z')]$
    iii). Find $\widehat{C}$ satisfying $u'\left(\widehat{C}(k,z)\right) = \beta E\left[V_1\left((1-\delta)k + zf(k) - \widehat{C}(k,z), z'\right)\right]$
Iterate on i)-iii) until convergence $\widehat{C} = C$.
```

ECM Policy Iteration

ECM-PI the variant of ECM that performs PI instead of VFI. It constructs a solution to the Bellman equation by iterating on the consumption function using EC (2.5).

Algorithm 5. ECM-PI

```
Given $C$, for each point $(k,z)$, define the following recursion:
    i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$
    ii). Solve for $V$ satisfying $V(k,z) = u(C(k,z)) + \beta E[V(K(k,z),z')]$
    iii). Find $\widehat{C}(k,z) = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf'(k)}\right]$
Iterate on i)-iii) until convergence $\widehat{C} = C$.
```

Derivative Policy Iteration

The ECM analysis also suggests a useful recursion for the derivative of value function. We first construct consumption function $C(k,z)$ satisfying FOC (2.4) under the current value function

$$\beta E\left[V_1\left(k',z'\right)\right] = u'\left(C(k,z)\right), \tag{2.7}$$

and we then use (2.5) to obtain the derivative of value function for next iteration. This leads to a solution method that we call ECM-DVF.

The main difference between ECM-DVF from the previously studied ECM-VF consists in that we iterate on V_1 without computing V on each iteration. We only compute V at the very end, when both V_1 and the optimal policy functions are constructed. Again, neither numerical maximization nor a numerical solver is necessary under ECM-DVF but only direct calculations. Similarly to PI methods, Euler equation methods do not solve for value function but only for decision (policy) functions. One possible decision function is a derivative of value function. Thus, the ECM-DVF recursion (2.7) can be also viewed as an Euler equation written in terms of the derivative of value function.

Algorithm 6. ECM-DVF

```
Given $V_1$ for each point $(k,z)$, define the following recursion:
    i). Find $c = u'^{-1}\left[\frac{V_1(k,z)}{1-\delta+zf^{prime}(k)}\right]$
    ii). Find $k' = (1-\delta)k + zf(k) - c$
    iii). Find $\widehat{V}_1(k,z) = \beta\left[1-\delta+zf'(k)\right]E\left[V_1(k',z')\right]$
Iterate on i)-iii) until convergence $\widehat{V}_1 = V_1$
Given the converged policy functions, find $V$ satisfying $V(k,z) = u(c) + \beta E\left[V(k',z')\right]$
```

Euler Equation Methods

Policy iteration has similarity to Euler equation methods; see Judd (1998), Santos (1999) for a general discussion of such methods. Euler equation methods approximate policy functions for consumption $c = C(k,z)$, capital $k' = K(k,z)$ (or other policy functions) to satisfy (2.2), (2.3) and (2.6). Below, we provide an example of Euler equation method (many other recursions for such methods are possible).

Algorithm 7. Euler equation algorithms

```
Given $C(k,z)$, for each point $(k,z)$, define the following recursion:
    i). Find $K(k,z) = (1-\delta)k + zf(k) - C(k,z)$
    ii). Find $c' = C(K(k,z), z')$
    iii). Find $\widehat{C}(k,z) = u'^{-1}\left\{\beta E\left[u'(c')(1-\delta+z'f'(k'))\right]\right\}$
Iterate on i)-iii) until convergence $\widehat{C} = C$.
```

Comparison results of Matlab, Python and Julia for policy iterative methods

In Table B.2, we see that conventional PI performs the worst because it relies on a numerical solver, but, as with the value iterative methods, this reliance is less problematic for Julia than for Matlab or Python.

Overall, we found that the speed of most algorithms was comparable across all three languages. The exception was that Julia performed significantly better than the other programs whenever there was any numerical optimization or root-finding involved. However, these speedups came at the cost of spending a (relatively small) amount of extra time being careful about how the code was written to ensure that the Julia compiler is able to infer the type of each variable in the most time-intensive parts of the program.

2.4 Comparison using a new Keynesian model

In this section, we solve a stylized medium-scale Keynesian model using equivalent codes implemented in Julia, Matlab, and Python. The model was previously analyzed in Maliar and Maliar (2015). It features Calvo-type price frictions and a Taylor rule with a zero lower bound on the nominal interest rate.

The model

To save on space, we present the model in Appendix B.

First-order conditions. Here, we summarize the model by showing the set of optimality conditions used to construct the numerical solutions:

$$
\begin{aligned}
S_t &= \frac{\exp\left(\eta_{u,t}+\eta_{L,t}\right)}{\exp\left(\eta_{a,t}\right)} L_t^{\vartheta} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon} S_{t+1}\right\}, && (2.8)\\
F_t &= \exp\left(\eta_{u,t}\right) C_t^{-\gamma} Y_t + \beta\theta E_t\left\{\pi_{t+1}^{\varepsilon-1} F_{t+1}\right\}, && (2.9)\\
\frac{S_t}{F_t} &= \left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}, && (2.10)\\
\Delta_t &= \left[(1-\theta)\left[\frac{1-\theta\pi_t^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_t^{\varepsilon}}{\Delta_{t-1}}\right]^{-1}, && (2.11)\\
C_t^{-\gamma} &= \beta\frac{\exp\left(\eta_{B,t}\right)}{\exp\left(\eta_{u,t}\right)} R_t E_t\left[\frac{C_{t+1}^{-\gamma}\exp\left(\eta_{u,t+1}\right)}{\pi_{t+1}}\right], && (2.12)\\
Y_t &= \exp\left(\eta_{a,t}\right) L_t \Delta_t && (2.13)\\
C_t &= \left(1-\frac{\overline{G}}{\exp\left(\eta_{G,t}\right)}\right) Y_t && (2.14)\\
R_t &= \max\left\{1, R_*\left(\frac{R_{t-1}}{R_*}\right)^{\mu}\left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi}\left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu}\exp\left(\eta_{R,t}\right)\right\}, && (2.15)
\end{aligned}
$$

where S_t and F_t are supplementary variables reflecting profit maximizing conditions of firms; Δ_t is a measure of price dispersion across firms; C_t, Y_t, L_t are consumption, output and labor, respectively; π_t and R_t are inflation and interest rate, respectively; $\pi*$ and $R* = \frac{\pi*}{\beta}$ are the inflation target and the steady state interest rate, respectively; $Y_{N,t} = \left[\frac{\exp(\eta_{a,t})^{1+\vartheta}}{[\exp(\eta_{G,t})]^{-\gamma}\exp(\eta_{L,t})}\right]^{\frac{1}{\vartheta+\gamma}}$ is the natural level of output. The parameters are β (discount factor), θ (fraction of non-reoptimizing firms), ε (inverse of the elasticity of substitution in the final-good production), ϑ and γ (utility-function parameters), $\overline{G}$ (steady-state government spending), and μ, ϕ_y and ϕ_π (Taylor-rule parameters). Each exogenous variable $\eta_{z,t} \in \{\eta_{u,t}, \eta_{L,t}, \eta_{a,t}, \eta_{u,t}, \eta_{B,t}, \eta_{G,t}\}$ follows an AR(1) process $\eta_{z,t+1} = \rho_z \eta_{z,t} + \epsilon_{z,t+1}$ with $\epsilon_{z,t+1} \sim \mathcal{N}\left(0, \sigma_z^2\right)$, $\sigma_z > 0$ and $\rho_z \in (-1,1)$.

Parameterization and implementation details. We parameterize the model by $\beta = 0.99$, $\gamma = 1$, $\vartheta = 2.09$, $\varepsilon = 4.45$, $\theta = 0.83$ and $\overline{G} = 0.23$. The autocorrelation coefficients in six exogenous variables are $\rho_a = 0.95$, $\rho_B = 0.22$, $\rho_R = 0.15$, $\rho_u = 0.92$, $\rho_G = 0.95$, $\rho_L = 0.25$, and the corresponding standard deviations are $\sigma_u = 0.54\%$, $\sigma_G = 0.38\%$,

$\sigma_L = 18.21\%$ $\sigma_a = 0.45\%$, $\sigma_B = 0.23\%$ and $\sigma_R = 0.28\%$. The parameters in the Taylor rule are $\phi_y = 0.07$, $\phi_\pi = 2.21$, and $\mu = 0.82$. These values are used in Maliar and Maliar (2015) and are in line with the estimates in Del Negro et al. (2007) and Smets and Wouters (2007).

As a solution domain, we use random and quasi-random grids constructed inside an 8-dimensional hypercube; we describe the grid construction in Section 4.2. To approximate the equilibrium policy rules, we use a family of ordinary polynomials up to degree 5. To compute conditional expectations in Euler equations (2.8), (2.9) and (2.12), we use a monomial integration rule (either a formula with $2N$ nodes or the one with $2N^2 + 1$ nodes); see Judd et al. (2011) for a detailed description of the monomial integration formulas. To solve for the polynomial coefficients, we use fixed point iteration. Again, Julia, Matlab, and Python software can be found in the corresponding notebooks submitted on QuantEcon[17].

Computational method. We now outline the Euler equation algorithm used to solve the neoclassical model. An extended description of this algorithm is provided in Appendix B.

[17] Matlab notebook, Python notebook, Julia notebook

Algorithm 8. Euler equation algorithms for the new Keynesian model.

```
Given $S(\Delta,R,\eta_z)$, $F(\Delta,R,\eta_z)$ and $C(\Delta,R,\eta_z)$ for each point $(\Delta,R,\eta_z)$, define the following recursion:
    i). Find $\pi$ from $\frac{S}{F}=\left[\frac{1-\theta\pi^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$ and $\Delta'=\left[(1-\theta)\left[\frac{1-\theta\pi^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}}+\theta\frac{\pi^{\varepsilon}}{\Delta}\right]^{-1}$;
    – $Y=\left(1-\frac{\overline{G}}{\exp(\eta_G)}\right)^{-1}C$, and $L=Y\left[\exp(\eta_a)\Delta'\right]^{-1}$;
    – $Y_N=\left[\frac{\exp(\eta_a)^{1+\vartheta}}{[\exp(\eta_G)]^{-\gamma}\exp(\eta_L)}\right]^{\frac{1}{\vartheta+\gamma}}$;
    – $R'=\max\left\{1,R_*\left(\frac{R}{R_*}\right)^{\mu}\left[\left(\frac{\pi}{\pi_*}\right)^{\phi_\pi}\left(\frac{Y}{Y_N}\right)^{\phi_y}\right]^{1-\mu}\exp(\eta_R)\right\}$.
    ii). Find $S(\Delta',R',\eta_z{}')$, $F(\Delta',R',\eta_z{}')$ and $C(\Delta',R',\eta_z{}')$;
    iii). Find $\pi'$ from $\frac{S'}{F'}=\left[\frac{1-\theta\left(\pi'\right)^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$;
    – $\widehat{S}(\Delta,R,\eta_z)=\frac{\exp(\eta_u+\eta_L)}{\exp(\eta_a)}L^{\vartheta}Y+\beta\theta E\left\{(\pi')^{\varepsilon}S'\right\}$;
    – $\widehat{F}(\Delta,R,\eta_z)=\exp(\eta_u)C^{-\gamma}Y+\beta\theta E\left\{\left(\pi'\right)^{\varepsilon-1}F'\right\}$;
    – $\widehat{C}(\Delta,R,\eta_z)=\left\{\frac{\beta\exp(\eta_B)R'}{\exp(\eta_u)}E\left[\frac{(C')^{-\gamma}\exp(\eta_u')}{\pi'}\right]\right\}^{-1/\gamma}$.
Iterate on i)-iii) until convergence $\widehat{F}=F$, $\widehat{S}=S$, $\widehat{C}=C$.
```

Grid techniques for high dimensional applications

Tensor product grids can be successfully used for analyzing small-scale applications but their cost increases exponentially with the number of the state variables (curse of dimensionality). In particular, tensor product grid are expensive even for moderately-large models like our model with 8 state variables. There is a number of alternative grid techniques that do not rely on tensor products and that ameliorate the curse of dimensionality; see Maliar and Maliar (2014) for a survey.

To solve the model (2.8)-(2.15), Maliar and Maliar (2015) construct the grid on the high-probability area of the state space which is approximated by stochastic simulation. However, crude simulated points are not uniformly spaced, in particular, many simulated points are situated close to one another and are redundant for the purpose of the grid construction. Maliar and Maliar (2015) introduce two techniques for selecting a roughly uniformly-spaced subset of simulated points. One technique is clustering analysis: the simulated points are grouped into clusters and the centers of the clusters are used as a grid. Another technique is an ε-*distinguishable set* (EDS) method that selects a subset of points situated at the distance of at least ε from one another. As an example, in Figure B.1 we show a stochastic-simulation grid and epsilon-distinguishable set grid.

Maliar and Maliar (2015) construct the EDS or cluster grids iteratively. In particular, they (i) guess policy functions and use them to simulate the time series solution; (ii) build an EDS or cluster grid; (iii) solve the model on this grid; (iv) then use the new solution to form a new grid. This procedure is repeated until the solution and grid both remain constant on successive iterations. The advantage of this approach is that it focuses on the right area of the state space – a high probability set. However, it requires to have initial guess for the solution (the paper uses linearization solution from Dynare) and it reconstructs the EDS or cluster grid iteratively, which leads to larger running times. Finally, the code is more complicated.

In the present paper, we modify the grid construction relative to the method in Maliar and Maliar (2015). We specifically replace the EDS and cluster grids with random and quasi-random grids covering a fixed multi-dimensional hypercube. To construct the hypercube, we choose the interval $\left[-\frac{2\sigma_z}{\sqrt{1-\rho_z^2}}, \frac{2\sigma_z}{\sqrt{1-\rho_z^2}}\right]$ for each of six exogenous variable $\eta_z \in \{\eta_u, \eta_L, \eta_a, \eta_u, \eta_B, \eta_G\}$, where σ_z and ρ_z are standard deviation and auto-correlation coefficient. For endogenous state variables R and Δ, we chose intervals that cover the high probability set, $\Delta \in [0.95, 1]$ and $R \in [1, 1.05]$.

We consider two alternative grid techniques in the constructed fixed hypercube. One is a *random grid* obtained by drawing an 8-dimensional set of random uncorrelated uniformly distributed points. The other is *quasi-random grids* (also known as *quasi-Monte Carlo* sequence or *low discrepancy sequence*). Uniformly distributed random draws converge to an evenly spaced grid as a number of draws increases but the convergence rate is low; in turn, low disrepancy sequences are constructed to be evenly spaced and their convergence rate is much faster; see Maliar and Maliar (2015) for a discussion. We use

Sobol low discrepancy sequence for constructing the grid but there are a variety of other low discrepancy sequences that can be used for this purpose; see Niederreiter (1992). The random and quasi-random grids are shown in Figure B.2.

Using random grids has its advantages and disadvantages. Unlike the EDS and cluster grids used in Maliar and Maliar (2015), the random and quasi-random grids operate on a fixed hypercube and do not benefit from the domain reduction. However, the random and quasi-random grids are much faster and easier to construct than the EDS and cluster grids. By constructing grids on predefined intervals, we did not have to write our own rough solution routine or rely on external software like Dynare to construct a preliminary solution to the model as we would with cluster grid or EDS techniques and we do not need to re-construct the grids iteratively. The MATLAB code accompanying the present paper achieves an almost a 60-time speed up over the original code in Maliar and Maliar (2015), while still producing highly accurate solutions.

Numerical results

We have provided an implementation of this algorithm in the Matlab, Python, and Julia programming languages. In this section we discuss our experience with the three languages and report our results in Table B.3.

We report the solution for two parameterizations that are previously analyzed in Maliar and Maliar (2015). To save on space, we report the solution only for the quasi-random (Sobol) grid. The solution for the random grid is similar (although slightly less accurate). Our two parameterizations differ in the value of the inflation target π_*. In our first experiment, we use $\pi_* = 1.0598$, which corresponds to long term average as estimated in Del Negro et al. (2007), while in the second experiment, we use $\pi_* = 1$. For the second experiment, we report results when we solve the model both when the ZLB is and is not imposed (in the first experiment, ZLB is never active).

The computational cost increases rapidly with a polynomial degree. The number of points in the grid must be at least as large as the number of the polynomial coefficients to identify such coefficients. In particular, for the polynomial degrees 1, 2, 3, 4 and 5, we construct 20,100, 300, 1000 and 2000 grid points respectively to identify 9, 45, 165, 495 and 1287 coefficients (we use about 50 percent more of grid points than the polynomial coefficients to enhance the numerical stability of the algorithm). The cost of second-order polynomial approximations is about 1-3 seconds depending on specific parameterization used. It is fast enough to repeatedly solve the model inside an estimation procedure.

Concerning the accuracy, in our first experiment, the inflation target is relatively high and the probability of reaching a zero lower bound (ZLB) on nominal interest rates is so low that it is never observed in finite simulation (unless we use the initial condition that leads to ZLB). In the absence of binding ZLB, the accuracy of our numerical solutions increases considerably with the degree of polynomial, in particular, the maximum residuals across the model's equations decrease from -1.94 to -5.47 when the polynomial order increases from 1 to 5 respectively. The accuracy of the solutions is surprisingly high given that we assume very large volatility of labor shock of 18%. If we reduce the size of the labor shock to 5%, the residuals in the table decrease by nearly 2 orders of magnitude and the accuracy levels will be comparable to those we obtained for the neoclassical growth model.

In our second experiment, the inflation target is low and the probability of reaching the ZLB increases to about 2%. When we do not impose the ZLB, the accuracy of the solution looks very similar to our first parameter set. However, with the ZLB imposed, the results are different. Relatively small average residuals indicate that the solution is sufficiently accurate in most of the domain and large maximum residuals show that the accuracy declines sharply in the ZLB area. As this example shows, even the global solution methods might still be insufficiently accurate in the presence of active ZLB because smooth polynomials functions are not well suitable for approximating the ZLB kink. Aruoba et al. (2017) and Maliar and Maliar (2015) show that the accuracy can be slightly increased by using piecewise linear and locally adaptive approximations, respectively, but these extensions lie beyond the scope of our analysis.

2.5 Conclusion

We have described and implemented seven algorithms to solve the stochastic neoclassical growth model, and we have described and analyzed a modification of an algorithm that has been previously used to globally solve new Keynesian models. The implementation has provided us a "laboratory" to explore some of the strengths and weaknesses of the three languages we consider. One insight we gained was that for some algorithms, the choice of programming language has little effect on performance, but for others, there are important speedups that can be obtained. In particular, if an algorithm depends heavily on numerical optimization or root-finding then it is likely that Julia will provide significant speedups, but may come at the cost of ensuring that your code provides enough information to the compiler.

We hope we have brought some clarity in the trade-off between these three programming languages which will first, provide information to economists who are thinking about which language is the best fit for them; and second, provide a clear description, examples, and tools which can facilitate other researchers implement similar algorithms for their own models.

Chapter 3

Student Loans

3.1 Introduction

Income driven student loan repayment (IDR) plans have gained broad support in the United States, including support by politicians from both major political parties. The key benefit of IDR is that it provides insurance against making infeasible loan payments during times in which an individual is experiencing unlucky labor market outcomes such as unemployment or underemployment. We focus on potentially unintended consequences of such policies by investigating how this insurance changes individual behavior for those considering, or already enrolled, in college. We do this by building a detailed model of pre, intra, and post college life and we use this model to understand potential trade-offs that arise from IDR by answering questions such as, “Does IDR improve student outcomes?”, “If we see improvements, how much does it cost to achieve these improvements?”, and “Could other policies achieve similar improvements at a lower cost?”

Student Loans in the U.S.

Currently, there are two main categories of student loan repayment plans in the U.S:

1. **Time-based payments**: These plans have a predetermined sequence of payments that an individual is expected to make, regardless of income. These payments typ-

ically begin after a short grace period once a student leaves college. The plan that borrowers are automatically enrolled in is a time based payment plan which requires individuals to make constant monthly payments for 10 years. We will refer to this plan as the amortized repayment (AMR) plan. There are other fixed payment plans where the payment starts low and increases over time, but the majority of borrowers are in the AMR plan.

2. **Income-driven Plans**: These payment plans tie the monthly payment to the amount that an individual earns in each month. The plans each require an individual to pay a pre-specified percentage of their income until their debt is paid off or until they have made a certain number of payments. The plans differ in the percentage of income paid each month and the number of payments until forgiveness. The plan we will focus on[1], known as REPAYE, requires that an individual pay 10% of their disposable income and for no more than 20 years.

Until recently, almost all student loans were held in the AMR plan. However, the government began increasing support for IDR in 2008 and 2009 in response to an increase in the number of individuals who were struggling to make their student loan payments due to the worsening labor market conditions associated with the 2008 recession. In Figure C.1, we see how quickly participation in IDR plans has increased over the last 5 years. In 2014, approximately 10% of borrowers (20% of loan dollars) were enrolled in an IDR plan, but, by the end of 2018, this percentage had increased to 30% of borrowers (50% of loan dollars).

As the participation rate in IDR plans has increased, so has the estimated cost of running them. In 2012, the Department of Education (ED) estimated that it would require a $1.2 billion subsidy to offer IDR to the 2012 loan cohort, but by 2017 that estimate had been revised to over $3 billion dollars due to higher than expected participation and other

[1] We focus on this plan because it is the most recent and all new IDR debt is held in this plan.

circumstances[2]. In a report[3] on the costs of IDR, the Government Accountability Office (GAO) raised a concern about how the inaccuracy of ED's forecasts impacted the ED budget. ED, while sharing this concern, highlighted the difficulty of forecasting these costs in their response to the GAO:

> . . . there are a number of factors that make forecasting future IDR participation inherently difficult. . . entails behavioral effects that are extremely difficult to incorporate and project into the future

This inherent difficulty of forecasting individual behavior motivates our approach of using a structural model that can account for changes in individual behavior.

Results Preview

In our model, when we move from an economy that uses only AMR to an economy that only uses IDR, the cost of running the student loan program increases by 15% per cohort. This increase in cost is accompanied by a 1 percentage point increase in the enrollment rate and almost no change in college completion rates.

The reason that IDR plans have such a small effect on college enrollment is two fold: First, individuals who attend college have precise information about the probability with which they will graduate from college. Since the returns to college in this model are relatively high, those who have a high probability of success already choose to attend college even though they face the possibility of burdensome student loan repayments. Second, the probabilty of experiencing periods of "burdensome student loan repayments" is relatively low. This means that the difference in the value functions associated with repaying your student loan under the AMR and IDR plans is low enough that enrollment behavior does not significantly change. Additionally, those with a low probability of graduating have no reason to enroll in school because, while they would gain access to student loans, they

[2] There were increases of similar magnitude in the expected subsidy for each cohort from 2009 to 2017. . . See Figure C.3

[3] See Government Accountability Office (2016)

would be forced to pay their yearly tuition costs without any significant increase in their earnings potential.

While the effect of IDR on student outcomes is relatively small, there are non-trivial effects on the cost of running the student loans program. Since the probability of enrollment changes very little across individuals in our economy, we are examining a similar population of students — The question becomes, “Why do the same individuals take out more debt under IDR than AMR?” The answer is that we have changed the incentives around debt accumulation. Under the AMR plan, all students expect that they will repay the debt or, if not, they will experience times in which the loan payments make up a large fraction of their income. This discourages students from taking on more debt than they expect to be able to repay. When students are allowed to repay according to an IDR plan, many students still expect to repay their loans, however, they know that if they were to experience unlucky labor market outcomes, then they wouldn’t be forced to endure periods of low consumption and may never be expected to fully repay the loan. This creates a disconnect between the marginal cost and marginal benefit of an additional dollar of debt, This disconnect incentivizes a higher fraction of students to accumulate some debt, from 46% under AMR to 70% under IDR, and also encourages more debt accumulation among those who take out loans, from $11,750 under AMR to $13,700 under IR. Notably, these effects are largest for those who are least likely to graduate from college in our model.

As with any model, we note that these numbers should be taken with a grain of salt. They depend on the particular structure of our model, while the model is not necessarily sensitive to perturbations of the parameters, it does depend on the labor income processes and the model’s timescale in meaningful ways. We view our model as a vehicle for studying these types of higher education policies and these results as an initial attempt at understanding the effects that IDR might have on pre and intra college behavior. Future work to improve certain aspects of the labor market in this model would have high returns for our understanding of IDR.

Related literature

There has been great interest in the effects of IDR on higher education financing. Insightful discussions of these programs, their proposed implementations, and some of these implications can be found in: Chapman and Harding (1993), Chapman (1994), Barr et al. (Forthcoming), and Dynarski (2014).

We reiterate that the core argument in favor of IDR is based on the repayment burden[4] imposed on borrowers by time based plans. According to work done in Chapman and Dearden (2017) and Chapman and Lounkaew (2015), under an AMR plan, the repayment burden for individuals who earn at the 10th percentile in the United States with a student loan of a typical size can be in excess 0.75. This means that, in order for certain individuals not to become delinquent on their student loans, they must put 75% of their income towards paying down their student loans. IDR plans mitigate these type of risk by placing an upper bound on the repayment burden.

This paper follows a literature that uses dynamic structural models to understand aspects of higher education including papers such as, Keane and Wolpin (2001), Arcidiacono (2005), Ionescu (2009), Chatterjee and Ionescu (2012), and Ferreyra et al. (2017). Recently, work done by Findeisen and Sachs (2016), Heijdra et al. (2017), Ji (2017), Luo and Mongey (2019), and Liu (2016) have used these types of dynamic structural models to think about the implications of IDR and other related student loan related policies. We discuss some relevant work below.

In both Ji (2017) and Luo and Mongey (2019), the authors build a model of post-graduation labor markets and consider the effects that student debt has on the wages that graduates accept. In Ji (2017), the author finds that higher student debt leads to graduates accepting a lower wage quickly after graduation so that they can pay off their debt. In Luo and Mongey (2019), the authors find that, once you incorporate job satisfaction, graduates with high debt seek a higher wage job with a lower job satisfaction. In both papers, the authors analyze the effects associated with moving all loans from an AMR system to an IDR

[4] See Section 3.2 for the precise definition of a repayment burden

system and find that the IDR plans relieve some of the pressure and that the relationships they find between debt and labor market outcomes become smaller due to the insurance provided by IDR. Both of these papers focus only on the role of IDR on decisions made by college graduates and do not include endogenous enrollment or debt decisions which are central to the questions asked in this paper.

Heijdra et al. (2017) uses a heterogeneous agent model with college attendance and human capital accumulation. In their model, rather than an IDR they use a tax levied specifically on those who attended college — this differs from the standard IDR in the sense that it is collected indefinitely and not stopped after an individual finishes paying off their debt. They find that, relative to the current student loan system in the U.S., the graduate tax improves ex-ante welfare for all individuals in the economy. There are a few important distinctions in what we do. They allow individuals to choose between no higher education, an associate's degree, or a bachelor's degree, but do not include dropout risk or endogenous debt accumulation. Moreover, we believe it would be difficult to implement a pure graduate tax due to political constraints — as mentioned in Barr et al. (Forthcoming), graduate taxes effectively impose an infinite interest rate since they can never be repaid.

Two other closely related papers that we think are worth mentioning are Chatterjee and Ionescu (2012) and Cai et al. (Forthcoming).

Chatterjee and Ionescu (2012) build a model of college attendance with the purpose of analyzing how the risk of dropping out and the associated negative labor market outcomes affects the decision to enroll in college. While these authors do not directly analyze an IDR, they do consider a related policy which involves forgiving the debt of college dropouts. They find that even though this plan induces some shirking by college students, that alleviating some of the potentially negative outcomes associated with dropping out encourages more individuals to attempt college and raises welfare by more than 2%.

Cai et al. (Forthcoming) asks very similar questions to the questions we ask in this paper in the context of implementing an IDR plan for student loans in China. They have panel data that includes the yearly earnings and education level for individuals in China. They use this data to estimate an income process and then explore what the implied government subsidies and repayment burdens would be for different repayment plans under an assumed level of student loans. The main differences between what they do and what we do in this paper is that (1) we consider potential selection effects by allowing for individuals to differ in their ability and earnings potential and (2) the amount of student loans accumulated in our model is endogenous whereas they focus on what would happen for a typical level of debt.

Finally, the foundation of our structural model builds on Hendricks and Leukhina (2017). In that paper, the authors use a model of college enrollment and attendance to determine how predictable an individual's success in college is given the information that they have upon graduating from high school. They depend on a detailed description of the pre and intra college decisions and the tradeoffs induced by these decisions to answer their question. These tradeoffs are important in the context of our question about the cost of different student loan plans, however, in their model there is no labor market risk and all uncertainty is resolved once an individual leaves college — however, risky labor income is crucial to the question at hand since it is one of the key benefits of the IDR plan and the most risky aspect of the AMR plan. To address this, we extend their model by explicitly modeling the working stage of life using a Bewley-Hugget-Aiyagari model with idiosyncratic labor market risk and incomplete financial markets.

In fact, Hendricks and Leukhina (2017) envision exactly this type of follow up to their work, they say,

> . . . Since our model captures the distribution of risks and returns associated with entering college, it provides a starting point for the study of college related policies, such as income contingent loans or dual enrollment programs. However, the model abstracts from two features that may be important for policy analysis. . . .
>
> . . . The first feature is study effort. . . The second feature is earnings risk during

> the work phase. One motivation of making college loans income contingent is to alleviate the tight budget constraints of young workers who may be borrowing constrained…

One additional benefit of building directly on their work is that pieces of their calibration were done with proprietary data that we do not have access to, and by building on their model, we can inherit some of these hard to determine parameter values.

Paper Outline

The paper is organized as follows:

In Section 3.2, we will work through a hypothetical situation involving a cohort of student loans to define some of the outcomes of interest. This example highlights the outcomes that will be of interest once we are comparing different repayment plans.

We then describe a model similar to Hendricks and Leukhina (2017) in Section 1.2 and in Section 3.3 we describe the assumptions and parameter choices for the baseline model.

We discuss the outcomes of this model and the differences between AMR and IDR in Section 3.4 and then conclude in Section 3.5.

3.2 Student Loan Examples

In this section, we present two hypothetical examples to motivate what comes in later sections:

First, we present an example describing a single student loan borrower. We use this example to highlight how different loan repayment plans might have an effect on the payments made by borrowers and use our example to define the *repayment burden*.

After discussing an example using a single student, we walk through an example involving an entire cohort of individuals. We use this cohort example as a vehicle to define certain metrics which we can use to evaluate the outcomes of loan repayment plans in our model.

Single Individual

Consider a recent college graduate that accumulated \$29,000 of student loans that have a 6.8% interest rate. After graduation, she begins working at a job that pays \$35,000 per year. For simplicity, we will assume that her income grows at 3% annually and ignore taxes. This means that during her first year of employment her monthly income would be \$2,900 per month.

Under the AMR plan, her monthly payment would be approximately \$300. The *repayment burden* is the monthly payment divided by the monthly income. In this case, the repayment burden is $\frac{300}{2900} \approx 0.10$. One compelling feature of the IDR is that it places a strict upper bound on the repayment burden and avoids occurences of the required loan payments being large relative to an individual's income. In our individual's case, the payment required under the IDR plan would be[5] $0.10 \times (2{,}990 - \frac{18{,}000}{12}) \approx 140$ which leads to a repayment burden of about 0.05[6].

In Figure C.2, we see how the repayment burden evolves over time for both loan repayment plans. We chose our numbers so that the student in our example had the average amount of debt for a college graduate and the 25th percentile earnings of a recent college graduate. Of course, it is also important to recognize that the repayment burdens we computed assume that the individual maintains her employment and does not experience bouts of unemployment — under the AMR as her monthly income goes to 0, the repayment burden goes to ∞.

We also can see that in our example the IDR plan spreads payments over a much longer horizon (17 years) than the AMR plan (10 years). One implication of this is that, if an individual fully repays her debt, then the individual will typically pay more[7] under IDR (\$46,500) than they would under AMR (\$37,500). Of course, if debt is forgiven at the end

[5] Disposable income is defined as 1.5 times the poverty level of income. For a single adult in the United States, the poverty level is around \$12,000

[6] Note that the repayment burden is not fixed at 0.10 because the IDR rate is a marginal cost and the repayment burden is an average cost.

[7] If their income is high enough that they repay their loan sooner than 10 years under IDR then they will pay less overall.

of 20 years then it is still possible that the overall payments are less.

Cohort

For our purposes we will define a cohort as all graduating high school students in a particular year. Consider a cohort that consists of 100,000 individuals. Suppose that 50,000 of these individuals choose to enroll in college and 25,000 successfully graduate from college. We define the following measures of student outcomes within a cohort:

- *Enrollment rate*: The enrollment rate is defined as the number of individuals in a cohort who enroll in college divided by total number of individuals in cohort. In our example, the enrollment rate would be 0.50.
- *Completion rate*: The completion rate is defined as the number of individuals in a cohort who graduate from college divided by total number of individuals in cohort. In the example, the completion rate would be 0.25.
- *Graduate rate*: The graduation rate is defined as the number of individuals in a cohort who graduate from college divided by the number of individuals who enroll in college. In our example, the graduate rate would be 0.50.

Suppose that of the 50,000 students who enroll in school, 30,000 of the students take out some amount of student loans and that the total amount of loans accumulated by the cohort totals to $150,000,000. We now define some measures of how much debt a particular cohort takes on while in college.

- *Fraction in debt*: The fraction in debt is measured as the number of students who accumulated debt divided by the number of students who enrolled in college. In our example, the fraction in debt is 0.60.
- *Average debt*: The average debt is measured as the total amount of loans accumulated by a cohort divided by the number of students who took out debt. In our example, the average debt is $5,000.

Finally, suppose that over the course of the loan lifetimes, the cohort makes payments that are worth \$140,000,000 in present discounted value. We then compute the implied *government subsidy*, Δ_{SL}, as the difference in the amount of loans issued and the present discounted value of payments made by the cohort[8]. In our example, the goverment subsidy is \$10,000,000. Furthermore, as mentioned before, the government subsidy can be composed into several relevant components:

$$\Delta_{SL} = \underbrace{\frac{\sum_i (d_i - X_i)}{\sum_i d_i}}_{\text{subsidy rate}} \times \underbrace{\frac{\sum_i d_i}{N_d}}_{\text{average debt}} \times \underbrace{\frac{N_d}{N_e}}_{\text{fraction in debt}} \times \underbrace{\frac{N_e}{N}}_{\text{enrollment rate}} \times N \tag{3.1}$$

where d_i is the amount of student debt held be individual i, X_i is the present discounted value of the payments made by individual i, N_d is the number of students who take out loans while in school, N_e is the number of students enrolled in college, and N is the number of individuals in the cohort.

The only feature of this decomposition that we have not previously seen is the *subsidy rate*. As we see in Equation 3.1, the subsidy rate is defined as the government subsidy divided by the total amount of debt issued in the economy. In our example, this is $\frac{150{,}000{,}000-140{,}000{,}000}{150{,}000{,}000} \approx 0.067$.

We will use this decomposition in Section 3.4 to understand which channels are most important for understanding how the government subsidy changes when we move from AMR to IDR. This understanding will also help us propose alternate policies that achieve some of the risk mitigation benefits without incurring the extra costs implied by our model.

3.3 Calibration

The structural model laid out in Section 1.2 is dependent on many parameters and several distributional assumptions. In an attempt to prevent ourselves from leveraging this fact

[8] An important note to make here is that if all student loans were repaid, then this subsidy would be negative. This is because the interest rate on student loans is typically higher than the interest rate used to compute government subsidies according to the Federal Credit Reform Act, so the present discounted value of the loan payments would exceed the cost of borrowing that money.

too much and to avoid overfitting the data, we use, where possible, parameter values and distributional assumptions that have been previously used in peer-reviewed research anywhere possible. This won't work for all of the model parameters since certain parameters are sensitive to the model specificiation. For example, the discrete choice shocks, π_e and π_c, need to be determined for our particular model because the effect they have on choices is dependent on the model's Bellman equations. For parameters that we can't determine externally, we use a simulated method of moments approach and aim to match certain moments from the data.

We describe the data sources we use and how we computed the relevant moments in Appendix C.1.

Distributional Assumptions

We discretize the types by drawing $N_j = 200$ types, $j := (m^j, k_1^j, q^j, z^j)$, distributed according to the following distribution:

Let $\{\varepsilon_j^m, \varepsilon_j^k, \varepsilon_j^q, \varepsilon_j^z\}_j$ be independent and identically distributed standard normals[9]. Then

$$
\begin{aligned}
k^j &= \max\{0, \mu_k + \sigma_k \frac{\varepsilon_j^k + \alpha_{km}\varepsilon_j^m}{\sqrt{1+\alpha_{km}^2}}\} \\
m^j &= \frac{\varepsilon_j^m + \alpha_{mq}\varepsilon_j^q + \alpha_{mz}\varepsilon_j^z}{\sqrt{1+\alpha_{mq}^2+\alpha_{mz}^2}} \\
q^j &= \mu_q + \sigma_q \frac{\varepsilon_j^q + \alpha_{qz}\varepsilon_j^z}{\sqrt{1+\alpha_{qz}^2}} \\
z^j &= \max\{0, \mu_z + \sigma_z \varepsilon_j^z\}
\end{aligned}
$$

Additionally, we assume that an individual's ability, conditional on their GPA (m) is drawn from

[9] In practice, we draw the εs using Sobol sequences rather than randomly to provide better coverage of the type space

$$a \sim N(\rho m, \sqrt{1-\rho^2})$$

where $\rho := \frac{\alpha_{am}}{\sqrt{1+\alpha_{am}^2}}$ is the correlation between a and m. We do this by choosing $N_a = 9$ discrete levels of a[10] and computing the implied joint probability distribution over the discrete a, j pairs.

We discretize ζ into $N_\zeta = 4$ possible values. A value ζ_i determines two financial states of college dq^i and Ω_v^i. We choose the form of the per-period cost of college as $q(j, \zeta_i) = q_j + dq^i$. Ω_v represents the choice set of hours that an individual can choose to work while in school and is chosen to match various levels of part-time to full-time[11]. ζ follows a Markov chain, so the cost of college and the hours an individual can work while in college potentially fluctuate while they are enrolled.

We assume the age specific components of income for each education level, $\theta^s(t)$, follows a cubic polynomial in age as in Guvenen (2009) and others

$$\theta^s(t) = \theta_0^s + \theta_1^s \text{age} + \theta_2^s \frac{\text{age}^2}{100} + \theta_3^s \frac{\text{age}^3}{1000}$$

Parameters

The model parameters are determined in three stages:

Previous Knowledge

The previous knowledge parameters consist of parameters that are either widely used or come from specific research papers. The model does not seem to be particularly sensitive to small deviations of these parameters.

These parameters are listed in Table C.1.

[10] We choose these values of a such that $a \in \{\bar{a}_1, \ldots, \bar{a}_{N_a}$ with $\bar{a}_1 < \bar{a}_2 < \cdots < \bar{a}_{N_a}$ such that each ability level is drawn with equal probability

[11] For more details on how these values were determined, see Hendricks and Leukhina (2017)

Externally Calibrated

Our externally calibrated parameters are parameters that have some form of a data counterpart that can be used to line the model up with the data. This ensures that the model will match a particular piece of the data, but that it cannot use these parameters to match unrelated pieces of data. We now briefly describe how we determined parameters that fell in this category and the associated values can be found in Table C.2.

We find the age specific component of income by following standard procedures for estimating a dynamic income process. We use PSID data on family income and restrict the sample to working age (24-60) males and remove those who have certain irregularities such as too low of yearly earnings or too many hours worked[12]. We needed to compute the age specific components on our own because the income dynamics literature typically focuses on the residual income so these values were not reported.

We choose to set the number of classes per year to 6 with each class worth 6 hours of credit[13] in the data. This means the maximum number of hours an individual can earn in a given year is 36 which, according to Hendricks and Leukhina (2017), is the 90th percentile of credits earned. Additionally, we set the maximum amount of time that students can stay in school to 6 years since less than 5% of college graduates take more time than this to graduate.

The data used in Hendricks and Leukhina (2017) corresponds to students who were in college in the mid to late 1980s. We follow their lead and set the cumulative student loan limit to -$19,750 (in 2000 dollars) which approximately matches the 1986 limit of $12,500.

A list of the externally calibrated parameters can be found in Table C.2

[12] We follow decisions made by Guvenen (2009) and Hryshko (2012) when selecting a subsample to work with

[13] This in effect means that each course in our model corresponds to 2 courses in the data since most courses in college are worth 3-4 hours of credit

Internally Calibrated

The final group of parameters are parameters that do not have any data based counterpart. We determine values for these parameters by choosing certain features of the data that we think are informative about these parameters and then numerically minimize the difference between these features in the data and in our model.

This includes the parameters δ_c, δ_v, π_e, π_c, U_{HS}, U_{CD}, U_{CG}, α_{km}, μ_k, and σ_k. We choose these jointly to match statistics on enrollment by HS GPA quartiles, completion by HS GPA quartiles, yearly dropout rates, yearly dropout rates by HS GPA quartiles, yearly fractions of college students with student loans, yearly average student loans, and college earnings. The parameter values for these parameters can be found in Table C.3.

Model Fit

The data and model values for the moments we described are displayed in Tables C.4. Overall, we find that our model is able to accurately capture the moments of the data that we are interested in. We discuss some specific examples below:

A mixture of the discrete choice shocks and disutility of being a college dropout/graduate helps us successfully match the enrollment and drop out rates quite accurately. While the overall yearly drop out rate matches quite well, we do slightly worse for the dropout rates by year and HS GPA quartile. In particular, our model predicts that those in the low GPA quartiles drop out a little slower than in the data, but those in the high GPA quartiles drop out a little more quickly than in the data. Additionally, we consider it a success that using the credit accumulation parameters from Hendricks and Leukhina (2017) results in us accurately matching the educational fractions and completion rates.

One feature that we struggled to match was that in our model too few people accumulated debt during their first year of college (11% of students in our model had debt after the first year whereas the data says that 26% of students have debt after their first year).

3.4 Repayment Plan Comparison

Now that we have presented the model and understand which aspects of the data it successfully replicates, we proceed to perform some counterfactuals. We will analyze how the outcomes from Section 3.2 differ according to the student loan repayment plan used in the economy. In particular, we will consider three versions of our economy:

1. The first, *AMR*, is our baseline model and uses the AMR plan previously described for student loan repayment.
2. The second, *LC*, is a model which uses the income driven repayment plan previously described, but the individuals in the model use decision rules from the *AMR* model.
3. The third, *IDR*, is a model which uses the income driven repayment plan for student loan repayment and individuals use their optimal decisions rules.

Comparing the *AMR* and *IDR* versions of our economy will inform us about how the outcomes differ across the two student student loan repayment plans. However, we think it is useful to also include compare these outcomes with the outcomes under *LC* as well because this is how many policy analysts have previously evaluated IDR plans; they use the observed outcomes from those who have already attended and left college and evaluate the outcomes of new repayment policies using the decisions individuals made under previous policies[14]. In contrast, comparing *AMR* and *IDR* accounts for the changes associated with individuals responding to changes in their environment and, by doing this, can potentially capture features that ED is currently missing in their forecasts.

In the subsections that follow, we will focus on comparing *AMR* against *IDR* and will return to *LC* in Section 3.4.

[14] This type of policy analysis is exactly what Lucas (1976) was concerned about when he wrote, “Given that the structure of an econometric model consists of optimal decision rules of economic agents, and that optimal decision rules vary systematically with changes in the structure of series relevant to the decision maker, it follows that any change in policy will systematically alter the structure of econometric models.”

Enrollment, completion, and graduation

We begin by thinking about the implications the different student loan repayment plans have for the student outcomes discussed in Section 3.2. Table C.5 shows how these outcomes differ for the *AMR* and *IDR* economies. The table shows that, in the aggregate, the enrollment increases by 1 percentage point in *IDR* relative to *AMR* and that there are no significant changes in either the completion or graduation rates.

There are two reasons that the change in enrollment rates is small

First, there is relatively little uncertainty about an individual's probability of successfully graduating from college. In Figure C.4, we confirm this by plotting the individual's expected probability of passing ($\int_a p(a)dF(a|j)$) against their actual probability of passing ($p(a)$).[15]. Individuals with a low ability signal find it not to be optimal to attend college no matter what the repayment method is because they are relatively certain about their inability to graduate and, as a college dropout, they would receive the same income process as the high school workers but must also pay tuition and the utility cost for being a college dropout, U_{CD}.

Secondly, enrollment rates are determined by the relative value of being a college student and high school worker[16]. Since the value of being a high school worker is unaffected by the college repayment plans, the change in probability must come through the value of being a college student. The value of being a college student is affected only indirectly through the value of being a college dropout/graduate worker with a given repayment plan. In our model, IDR does not provide high value insurance because the estimated income process places low probability on individuals experiencing "low" incomes. We demonstrate this in Figure C.5 by plotting the difference in the value functions associated with being a college dropout in an economy under IDR and AMR across various income levels and we then overlay the probability distribution of these incomes for the individual most

[15] The high precision of the ability signals is a result of Hendricks and Leukhina (2017)

[16] If the value of being a college student is given by V^S and the value of being a high school worker is V^{HS} then the discrete choice shock specification in our model leads the probability of enrolling to be $\frac{\exp(V^S)}{\exp(V^{HS})+\exp(V^S)}$

likely to experience low income. We continue to reiterate the caveat that there are aspects of labor income that do not appear in our model due to its yearly frequency. At a monthly frequency one could include unemployment and other features that would increase the value of this insurance by increasing the probability that individuals experience periods of low labor income, so we view these results as a lower bound on the value of this insurance.

The fact that there is almost no change in the completion or graduation rates follows closely from the small change in the enrollment rates coupled with the small changes associated with the value of being a college dropout/graduate. Since the enrollment rate changes very little, no more than 2% for any type in our model, a very similar group of students choose to attend college and, since the value of being a college dropout/graduate has changed little, the probability that they dropout in any given period also changes little.

Debt decisions

We next examine how the different repayment plans affect decisions for debt accumulation for college students. We see in Table C.6 that the fraction of college students who exit with debt increases from 46% to 70% and the average debt held by these students goes from $11,750 to $13,740. This outcome is not particularly surprising — lowering the risk associated with having more debt, incentivizes students to accumulate debt while in school. However, understanding the composition of these changes will improve our understanding of what drives the change in the government subsidy of the student loan program.

In Figure C.6 we display the change in the fraction of students with debt and average debt levels by the individual's HS GPA quartile. We see that there are increases across the board in both variables, however, the largest effects come from a change in behavior by those with the lowest GPA levels. The bottom quartile sees a 50 percentage point increase in the fraction of students who take out loans and a $4,000 increase in the average debt levels while the top quartile only sees an increase of 11 percentage points and $1,500, respectively.

To understand why this happens, let's now consider the marginal benefit and marginal cost of an additional dollar of student loans. The marginal benefit of an additional dollar of debt is some mix of the extra consumption and less hours worked during the period the debt is accumulated. The marginal cost of an additional dollar of debt is the present discounted value of the corresponding payments that would be made. In Figure C.7, we plot the ratio of an additional dollar of debt divided by the present discounted value of the payments made. Loans made under AMR are almost always fully recovered so the ratio is 1, but under IDR there is positive probability that an individual will not be required to fully repay their loan.

Suppose for a moment that the individual was given perfect knowledge of their income realizations during their last year of college and that they knew that, given their current debt level, some amount of their loan would be forgiven. This knowledge implies $\frac{\partial V_t^{C?}}{\partial d_t} = 0$ which, in turn, means that the individual loses nothing from accumulating additional debt (but benefits from additional consumption during school). Obviously, this example is more dramatic than what would typically occur in our model but we can show that there exists some bound, $\tilde{D}$, such that as $\lim_{d \to \tilde{D}}$ we see $\frac{\partial V_t^{C?}(k_t, d_t, S_t, w_t, \varepsilon_t^y)}{\partial d_t} \to 0$. Since both income and GPA are positively correlated with ability level, the individuals who are least likely to repay their loans are those from the lowest HS GPA quartiles. This results in students from the lowest HS GPA quartiles having the largest increase in incentives to accumulate additional debt which is what we observe in our model.

Cost of IDR

Finally, we turn to thinking about the relative cost of these two policies to the government. We write the percent change in the government subsidy associated with moving from *AMR* to *IDR* as:

$$\% \text{ Change in } \Delta_{SL} = 1 - \left(\frac{\Delta_{SL}^{IDR}}{\Delta_{SL}^{AMR}} \right)$$

In our model, the government subsidy for the *IDR* economy is 15% larger than in the *AMR* economy.

We can decompose this into the four components described in Section 3.2 — the enrollment rate ($\tilde{N}_e$), the fraction of students with debt ($\tilde{N}_d$), the average debt among students with debt ($\bar{d}$), and the subsidy rate (δ_{SL}) — to write the percent change as the product of the ratios of each component:

$$\%\text{ Change in } \Delta_{SL} = 1 - \left(\frac{\tilde{N}_e^{IDR}}{\tilde{N}_e^{AMR}}\right) \times \left(\frac{\tilde{N}_d^{IDR}}{\tilde{N}_d^{AMR}}\right) \times \left(\frac{\bar{d}^{IDR}}{\bar{d}^{AMR}}\right) \times \left(\frac{\delta_{SL}^{IDR}}{\delta_{SL}^{AMR}}\right)$$

In Section 3.4 and Section 3.4, we've already discussed how each of these components changes except for the subsidy rate. In Table C.7, we report the government subsidy rates and a per person cost[17] for each of the different economies.

One unexpected feature of our model that is worth discussing is that the student loan program has a negative subsidy (positive returns) for the government. While there are certain components of the U.S. student loan portfolio that have a negative subsidy rate[18], this is not broadly true in the U.S. and should be seen as a shortcoming of our model. The negative subsidy rate is likely tied to two components of our model: First, the government can perfectly enforce its student loan program and there is no option for default in the *AMR* economy. Second, everyone in our model participates in labor market for an extended amount of time and makes their payments. This outcome reinforces to us the fact that expanding the model to capture additional features of the loan repayment system or labor market could be useful in evaluating these proposals, but this significantly increases the computational difficulty of the model. Rather than get distracted with these improvements,

[17] We choose to use a per person cost rather than report the full subsidy because the amount of subsidy is dependent on the number of individuals are in the simulated cohort — If one wanted to estimate how large this subsidy would be for a cohort of high school graduates, this number could be multiplied by the cohort size

[18] For example, the estimated subsidy on Federal Direct Loans issued in 2017 that are being repaid using the AMR plan is approximately -15 billion dollars, see Government Accountability Office (2016). If one divided this -15 billion by the total number of Federal Direct Loans issued that were being repaid using IDR then we would get a negative subsidy rate.

we focus on the behavioral changes that occur and highlight possible channels that ED could use to improve their own forecasts.

In Section 3.4 we highlighted the heterogeneity in the changes of debt accumulation by HS GPA quartile. In TableC.8, we show how the subsidy rates change by the student loan repayment plan for different HS GPA quartiles. The key takeaway is that the majority of the increase in the required government subsidy comes from those in low HS GPA quartiles — The largest of which comes from an almost 200% increase in the required subsidy rate to students from the lowest GPA quartile.

The final exercise we do is ask, "How much could the subsidy rate decrease, holding the other changes that occur under *IDR*, constant, without increasing the government subsidy beyond its *AMR* value?" In order to not increase the government subsidy, we need:

$$\left(\frac{\tilde{N}_e^{IDR}}{\tilde{N}_e^{AMR}}\right) \times \left(\frac{\tilde{N}_d^{IDR}}{\tilde{N}_d^{AMR}}\right) \times \left(\frac{\bar{d}^{IDR}}{\bar{d}^{AMR}}\right) \times \left(\frac{\delta_{SL}^{IDR}}{\delta_{SL}^{AMR}}\right) = 1$$

This would require a government subsidy rate of -0.07 under *IDR* which is significantly lower than the -0.13 from *AMR*. The reason that a subsidy rate of -0.07 can maintain the same goverment subsidy is that we have increased the volume of student loans in the economy by changing what fraction of students accumulate debt and how much debt they accumulate. In fact, one way that this subsidy rate can be achieved under the *IDR* plan is by simply offering IDR to only the top three GPA quartiles.

Lucas Critique

We now return briefly to the *LC* model. Recall that in this model we use IDR to repay the loans made in the economy, but the individuals in our model make decisions as if they were in the *AMR* economy. Since the decision rules by the agents have not changed, there are no changes to the enrollment/dropout decisions and no changes to the amount of debt accumulated. In this economy, the government subsidy still increases, due to limit on the repayment burden and eventual forgiveness, but it only increases by 8% rather than

15%. We believe that this helps reiterate ED's point that in order to accurately evaluate the cost of the IDR plans, there must be an understanding of what implications they have on individual behavior.

3.5 Conclusion

As the US continues to expand their IDR programs, it is important that ED is able to provide an accurate forecast of the required government subsidy. In their response to Government Accountability Office (2016), they argue that, in order to do this accurately, one must account for changes in behavior by potential borrowers. We do this by building a structural model of pre, intra, and post college decisions and use our model to evaluate some of the changes that might occur if the student loan program was switched from a strictly AMR system to a strictly IDR system. We find that IDR has small effects on student outcomes, but that it raises the cost to the government of running the student loan program. We use our model to pinpoint what drives these increased costs and find that the increase is mostly driven by additional debt accumulated by those who are unlikely to be able to repay their debts.

Appendix A

Supplementary Material for Chapter 1

A.1 Computation

*S*caling. One challenge with computing solutions to this model is that the productivity process (1.8) has a unit root. We deal with that by scaling: We divide everything by $\bar{z}_t$ and label it with a tilde: $\tilde{c}_{1t} = c_{1t}/\bar{z}_t$, $\tilde{c}_{2t} = c_{2t}/\bar{z}_t$, $\widetilde{U}_t = U_t/\bar{z}_t$, and so on. We define the relevant state by $\tilde{s}_t = (\widehat{z}_t, v_t)$ and the growth rate in the scaling variable $\bar{z}_t$ by $g_{t+1} = \bar{z}_{t+1}/\bar{z}_t$.

Since the functions are all hd1, we can divide the whole Bellman equation by $\bar{z}_t$:

$$\begin{aligned}
\widetilde{J}(\widetilde{U}_t, \tilde{s}_t) &= \max_{\{\tilde{c}_{1t}, \widetilde{U}_{t+1}\}} V\{\tilde{c}_{1t}, \mu_t[g_{t+1}\widetilde{J}(\widetilde{U}_{t+1}, \tilde{s}_{t+1})]\} \\
\text{s.t.} \quad & V\{\tilde{c}_{2t}, \mu_t(g_{t+1}\widetilde{U}_{t+1})\} \geq \widetilde{U}_t \\
& \text{plus resource constraints and shocks.}
\end{aligned}$$

The scaled resource constraints become

$$\begin{aligned}
\tilde{a}_{1t} + \tilde{a}_{2t} &= \tilde{y}_{1t} = \widehat{z}_t \\
\tilde{b}_{1t} + \tilde{b}_{2t} &= \tilde{y}_{2t} = 1/\widehat{z}_t \\
\tilde{c}_{1t} &= h(\tilde{a}_{1t}, \tilde{b}_{1t}) \\
\tilde{c}_{1t} &= h(\tilde{a}_{1t}, \tilde{b}_{1t}).
\end{aligned}$$

The laws of motion for the shocks are (1.11) for $\widehat{z}_t$ and (1.9) for v_t (no scaling required). $\bar{z}_t$ drops out, leaving us with a lower-dimensional state.

Given a solution to the scaled problem, we can multiply by $\bar{z}_t$ to produce a solution to the original problem. The notation is horrendous. The point is simply that we can convert our problem to one with stable shocks.

*A*lgorithm. Another challenge in computing a solution is that at each date t, we need to choose utility promises $\widetilde{U}_{t+1}$ for every state the following period. We adapt the algorithm of Collin-Dufresne, Johannes, and Lochstoer (2015) to a multi-good setting. Their algorithm has three essential features. First, they make a clever choice of state variable. Second, they use backward recursion, starting at a terminal date and computing the value function recursively at earlier dates. Third, they use the first-order conditions to solve for the optimal policies, rather than simply choosing the best from a finite set of possibilities.

Consider the state variable. We characterize the problem using promised utility $\widetilde{U}_t$ as the state. Collin-Dufresne, Johannes, and Lochstoer replace promised utility with the consumption share. Given the consumption share, the conditions of the solution give us promised utility as a byproduct. We use the (additive) Pareto weight the same way, with consumption shares and promised utility as byproducts. The choice of variable does not change the solution, but in our experience it can have a large effect on the performance of the algorithm.

Now consider backward recursion. We approximate an infinite-horizon dynamic programming problem with a long finite-horizon problem. At the terminal date T, we set utility equal to current consumption. This corresponds to a steady state in which consumptions are constant at these values at all future dates. If we know the Pareto weight λ^*_T in all states at T, we can compute allocations of the two intermediate goods and consumptions of the two agents in the same states. The consumptions then give us value functions for each state at T that we can use the previous period.

In any previous period t, we need to choose both current consumptions and continuation values for the Pareto weight in every succeeding state at $t+1$. We choose the Pareto weights λ^*_{t+1} and current consumptions c_{jt} by solving the planner's first-order conditions. We speed this up by pre-solving the allocation problem on a fine grid for the state variables. Given a solution for the Pareto weight, the time aggregators then give us current utility and the value function.

We repeat this process until the value function and decision rules converge. We use a sup norm criterion over both value functions and decision rules.

We use a discrete grid for the Pareto weight and other state variables. Our numerical implementation has 451 points for the Pareto weight, 13 points each for the exogenous state variables $\widehat{z}_t$ and v_t, and 5 quadrature nodes for each of the three future innovations — a total of 9.5 million points. We then use Hermite quadrature to compute the expectations in the certainty equivalent functions. If the number of points seems high, we have found that it's essential to have a fine grid over the Pareto weight to describe its dynamics adequately. We could probably work with a less fine grid, but this gives us some confidence that the solutions are accurate.

We do all of these calculations in Julia, a programming language that combines the convenience of dynamic vector-based languages like Matlab with the speed of compiled languages like C or Fortran.

A.2 Tables and Figures

Table A.1: Benchmark parameter values.

Parameter	Value	Comment
Preferences		
ρ	-1	IES $= 1/(1-\rho) = 1/2$
α	-9	RA $= 1-\alpha = 10$
β	0.98	
Armington aggregator		
σ	0	Cobb-Douglas
ω	0.1	chosen to hit import share of 0.1
Productivity growth		
$\log g$	0.004	Tallarini (2000, Table 4)
$v^{1/2}$	0.015	Tallarini (2000, Table 4), rounded off
φ_v	0.95	Backus, Ferriere, and Zin (2015, Table 1)
τ	0.74×10^{-5}	makes v three standard deviations from zero
γ	0.1	persistence of productivity difference

Figure A.1: Consumption frontiers. Lines represent the frontier quantities of consumption given unit quantities of the intermediate goods. The dashed black line has $\omega = 1/2$, making the two final goods the same. For the others, we choose an import share of 0.1 and use (1.13) to adjust ω as we vary σ. The elasticities of substitution noted in the figure are $1/(1-\sigma)$.

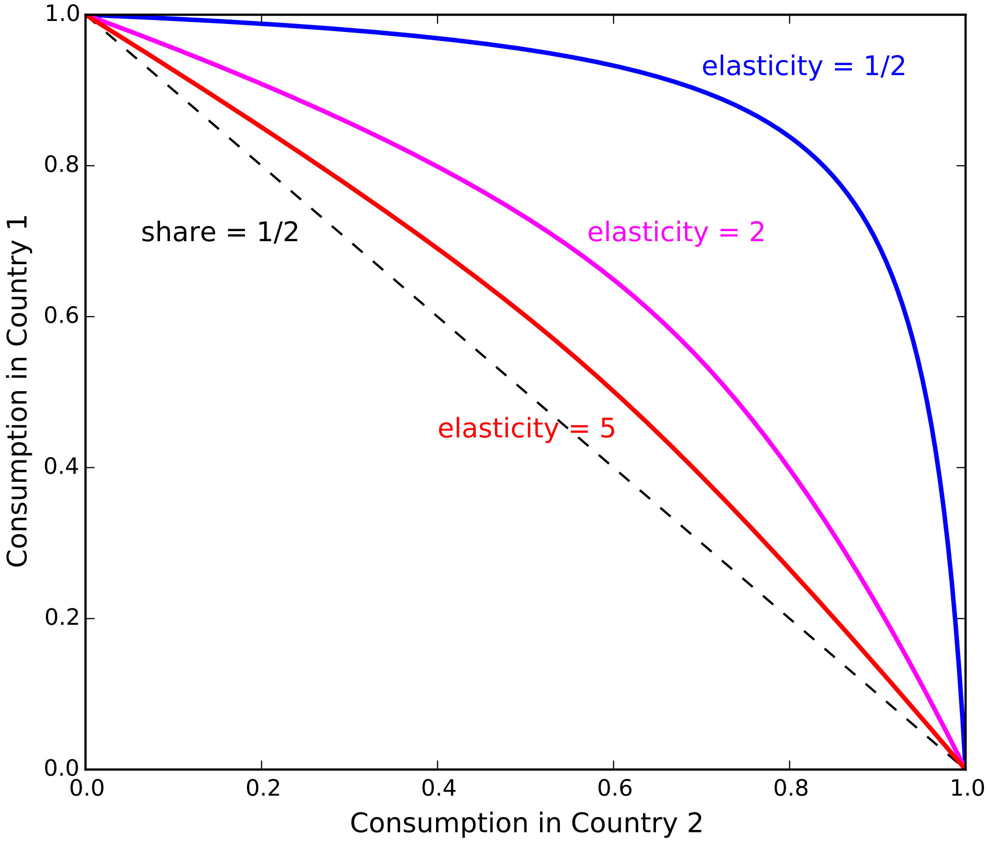


Figure A.2: Pareto and consumption frontiers. The outer line is the consumption frontier with benchmark parameter values. The inner line is the Pareto frontier: the utility J of agent 1 given promised utility U to agent 2. In each case, the other state variables are $z_{1t} = z_{2t} = \widehat{z}_t = 1$ and $v_t = v$.

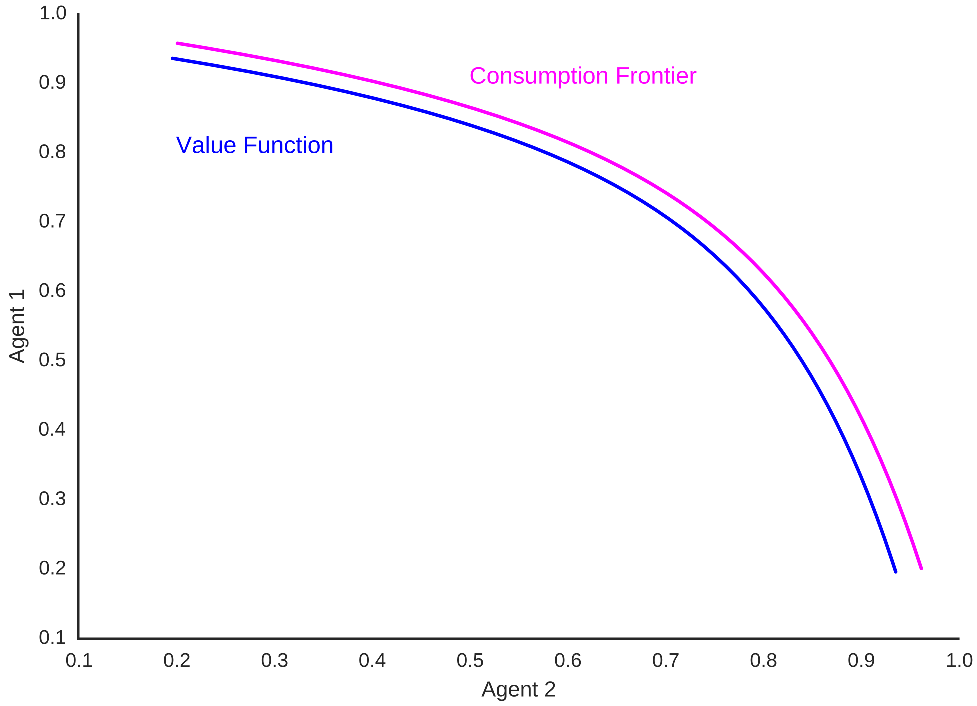


Figure A.3: Dynamics of the additive and recursive Pareto weight. The two lines represent simulations of models with additive ($\alpha = \rho = -1$) and recursive ($\alpha = -9$, $\rho = -1$) preferences. The simulations use the same paths for exogenous state variables. In each case, we plot $\log \lambda_t^*$ against time.

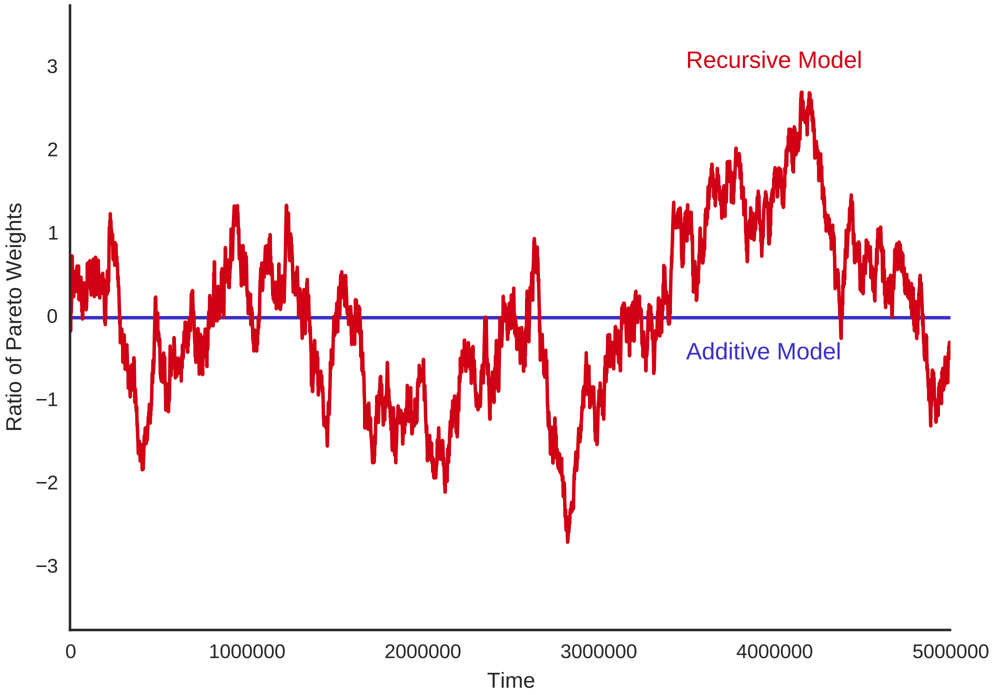


Figure A.4: Risk aversion and expected changes in the Pareto weight. The lines represent the expected change in $\log \lambda_t^*$, or $E_t[\log \lambda_{t+1}^*] - \log \lambda_t$, with three values of risk aversion $1-\alpha$: 2 (additive), 10, and 50.

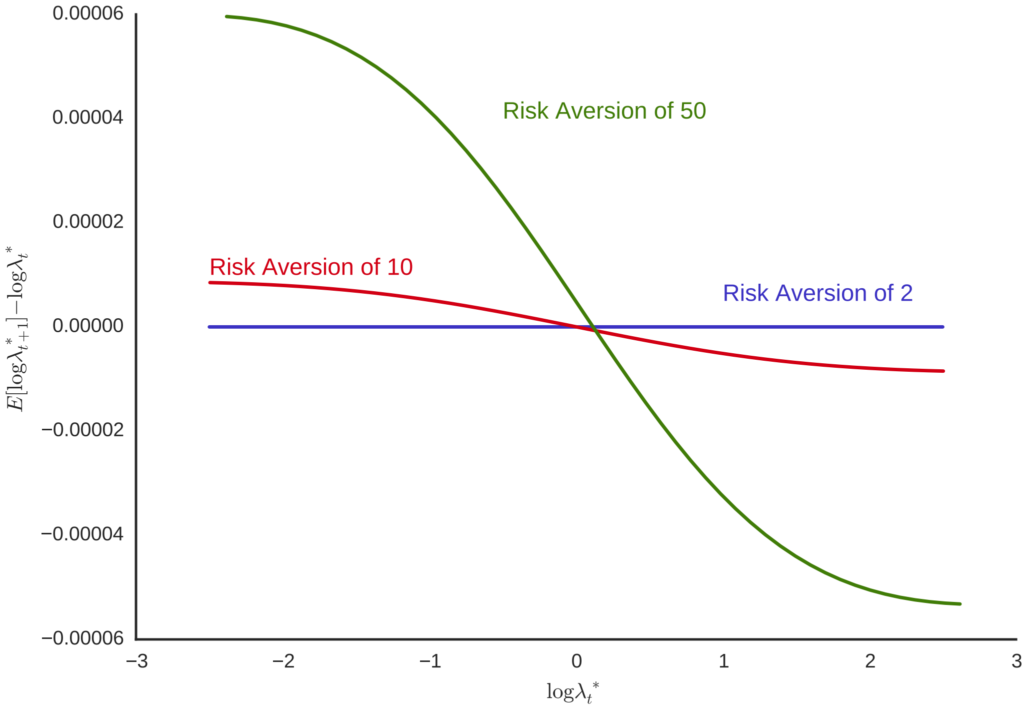


Figure A.5: Armington substitutability and expected changes in the Pareto weight. The lines represent the expected change in $\log \lambda_t^*$, or $E_t[\log \lambda_{t+1}^*] - \log \lambda_t$, with three values of the substitutability parameter σ in the Armington aggregator. The elasticities $1/(1-\sigma)$ are 2/3, 1, and 2.

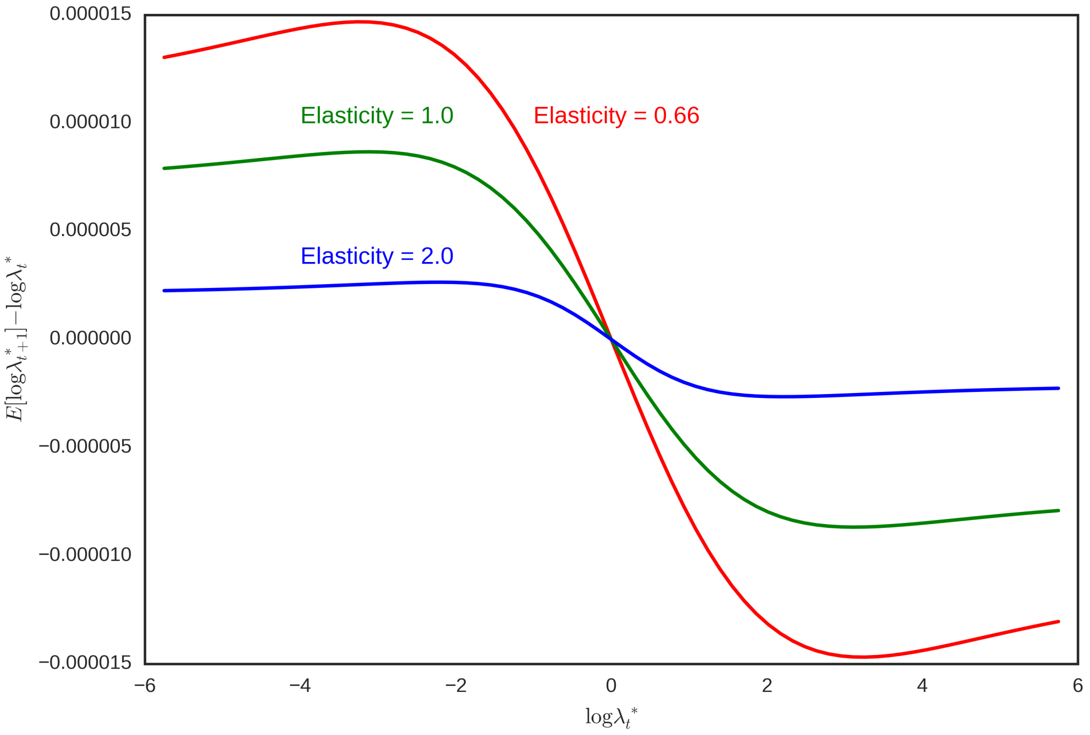


Figure A.6: Intertemporal substitution and expected changes in the Pareto weight. The lines represent the expected change in $\log \lambda_t^*$, or $E_t[\log \lambda_{t+1}^*] - \log \lambda_t$, with three values of the substitutability parameter ρ in the time aggregator: -1, -0.01, and 1/3. They correspond to intertemporal elasticities of substitution of 1/2, 0.99, and 3/2.

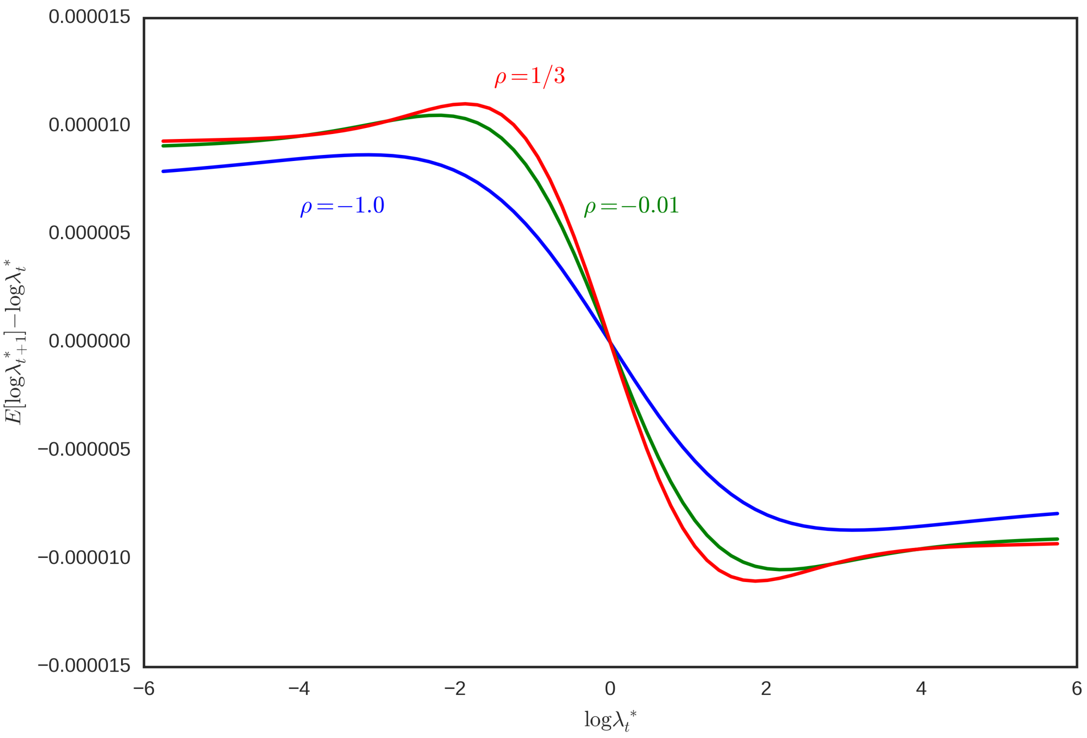


Figure A.7: Consumption and the real exchange rate. The dots represent simulations of models with additive ($\alpha = \rho = -1$) and recursive ($\alpha = -9$) preferences. In each case, we plot $\log e_t = \log(p_{2t}/p_{1t})$ against $\log(c_{2t}/c_{1t})$ for a simulation of the model.

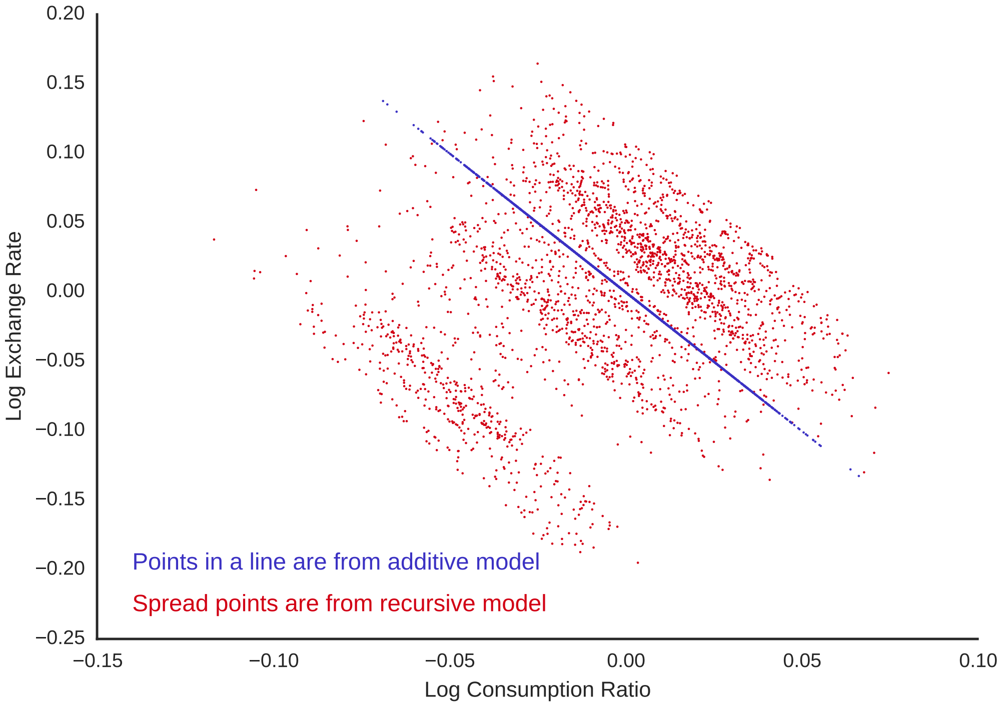


Figure A.8: Dynamics of the real exchange rate. The lines represent autocorrelation functions for the real exchange rate ($\log e_t$) in models with additive ($\alpha = \rho = -1$) and recursive ($\alpha = -9$) preferences.

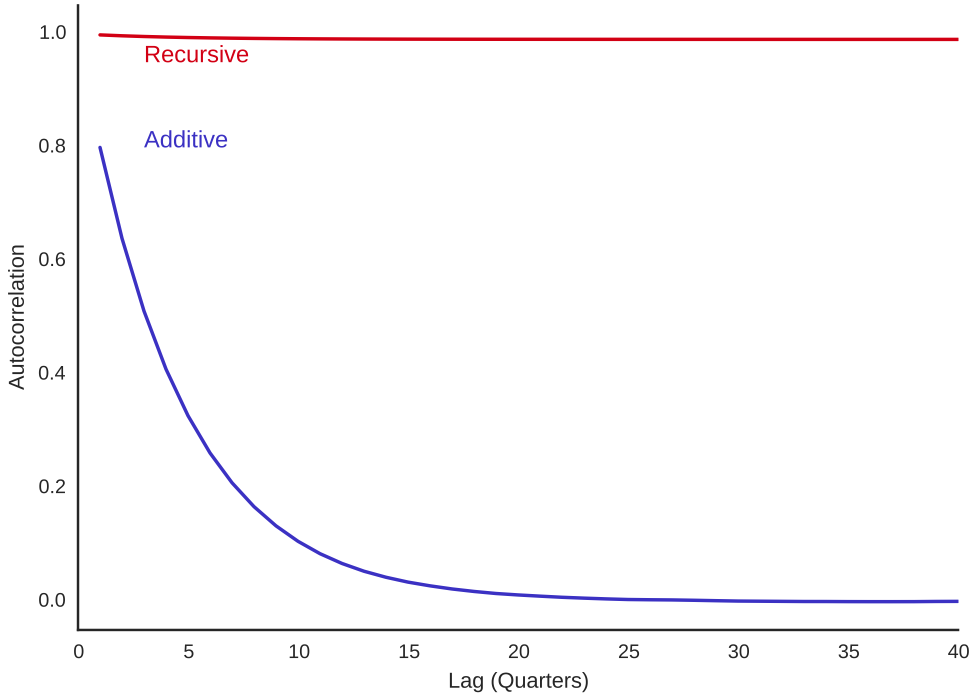


Figure A.9: Responses of variables to an impulse in relative productivity $\log \widehat{z}_t = (1/2)(\log z_{1t} - \log z_{2t})$ in country 2. The impulse takes place at date $t = 1$. Responses are reported as percent deviations from mean values.

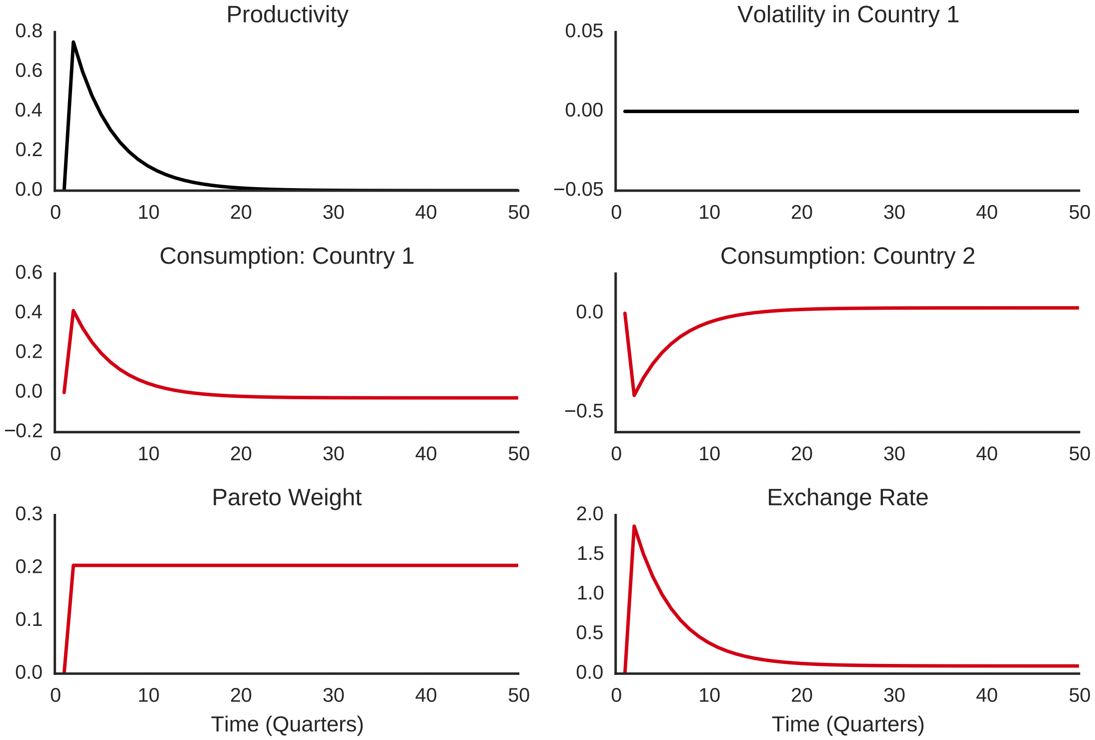
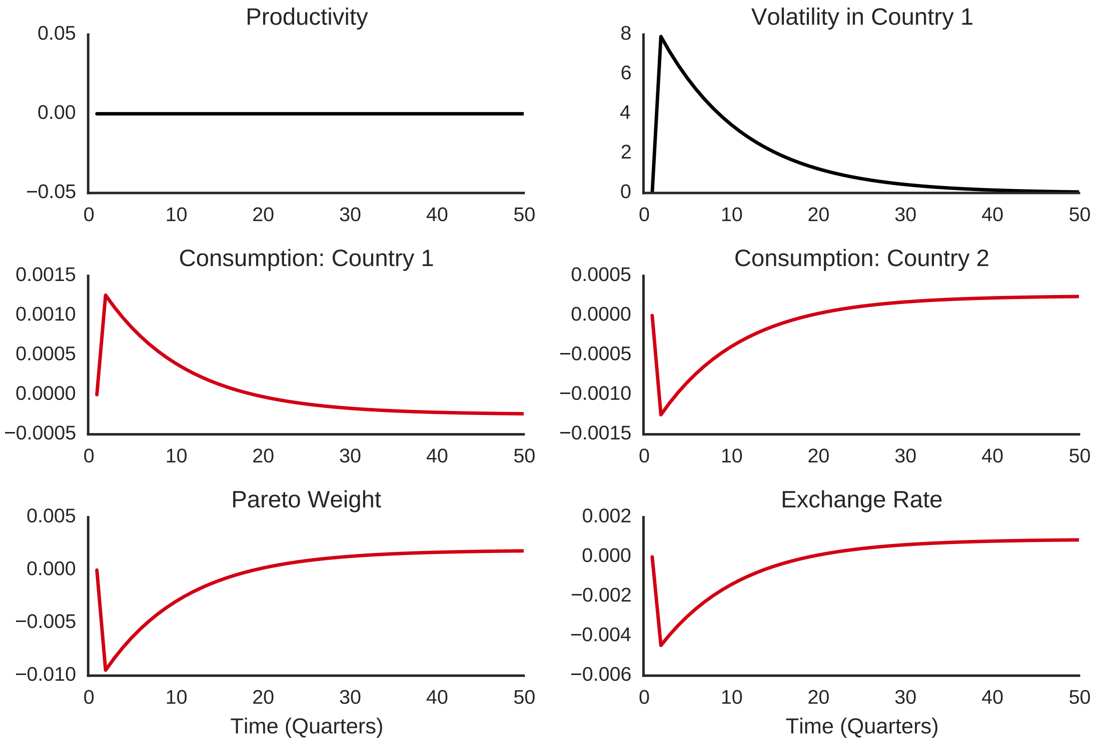


Figure A.10: Responses of variables to an impulse in volatility v_t. The impulse takes place at date $t = 1$. Responses are reported as percent deviations from mean values.



Appendix B

Supplementary Material for Chapter 2

B.1 New Keynesian Model

In A1-A7, we provide a description of Bellman and Euler equation algorithms, which we use to solve the neoclassical stochastic growth model with inelastic labor supply described in Section 2 including i) envelope condition method iterating on value function (ECM-VF); ii) conventional value function interation (VFI); iii) endogenous grid method (EGM); iv) policy function iteration via envelope condition (ECM-PI); v) conventional policy function iteration via FOC (PI); vi) envelope condition method iterating on derivative of value function (ECM-DVF); vii) conventional Euler equation method (EE).

A1: Envelope condition method (ECM).

Algorithm 1. ECM.

```
Initialization.
    a. Choose an approximating function $\widehat{V}(\cdot;b) \approx V$.
    b. Choose integration nodes, $\varepsilon_j$, and weights, $\omega_j$, $j = 1,...,J$.
    c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
    d. Make an initial guess on $b^{(1)}$.
Iterative cycle. Computation of a solution.
    At iteration $i$, perform the following steps:
Step 1. Computation of values of $V$ on the grid.
    For $m = 1,...,M$,
    a. Use $b^{(i)}$ to compute $\widehat{V}_1\left(k_m, z_m; b^{(i)}\right)$.
    b. Compute the corresponding values of $c_m$ using
    $c_m = u'^{-1}\left[\frac{\widehat{V}_1\left(k_m, z_m; b^{(i)}\right)}{1-\delta+z_m f'(k_m)}\right]$.
    c. Find $k'_m$ using
    $k'_m = (1-\delta)k_m + z_m f(k_m) - c_m$.
    d. Find value function on the grid
    $\widehat{v}_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}\left(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)}\right)$.
Step 2. Computation of $b$ that fits the value function on the grid.
    Run a regression to find $\widehat{b}$ :
    $\widehat{b} = \arg\min_{b} \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.
Step 3. Convergence check and fixed-point iteration.
    a. Check for convergence for $i \geq 2$: end Step 2 if
    $\frac{1}{M}\sum_{m=1}^{M}\left|\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}\right| < 10^{-9}$.
    b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi)b^{(i)} + \xi\widehat{b}$.
```

A2: Value function iteration (VFI).

```
Algorithm 2. VFI.
Initialization.
    a. Choose an approximating function $\widehat{V}(\cdot;b) \approx V$.
    b. Choose integration nodes, $\varepsilon_j$, and weights, $\omega_j$, $j = 1,...,J$.
    c. Construct a grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
    d. Make an initial guess on $b^{(1)}$.
Iterative cycle. Computation of a solution.
    At iteration $i$, perform the following steps:
Step 1. Computation of values of $V$ on the grid.
    For $m = 1,...,M$,
    a. Solve for $k'_m$ satisfying
    $u'\left((1-\delta)k_m + z_m f(k_m) - k'_m\right) = \beta \sum_{j=1}^{J} \omega_j \widehat{V}_1\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.
    b. Find $c_m$ satisfying
    $c_m = (1-\delta)k_m + z_m f(k_m) - k'_m$.
    c. Find value function on the grid
    $\widehat{v}_m \equiv u(c_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.
Step 2. Computation of $b$ that fits value function on the grid.
    Run a regression to find $\widehat{b}$ :
    $\widehat{b} = \arg\min_{b} \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.
Step 3. Convergence check and fixed-point iteration.
    a. Check for convergence for $i \geq 2$: end Step 2 if
    $\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}} \right| < 10^{-9}$.
    b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.
```

A3: Endogenous grid method (EGM).

Algorithm 3. EGM.

Initialization.

a. Choose an approximating function $\widehat{V}(\cdot;b) \approx V$.

b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1,...,J$.

c. Construct grid $\Gamma = \{k'_m, z_m\}_{m=1}^{M}$.

d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V on the grid.

For $m = 1,...,M$,

a. Compute $\widehat{W}\left(k'_m, z_m; b^{(i)}\right) \equiv \sum_{j=1}^{J} \omega_j \widehat{V}\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$ and $\widehat{W}_1\left(k'_m, z_m; b^{(i)}\right) \equiv \sum_{j=1}^{J} \omega_j \widehat{V}_1\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.

b. Find $c_m = u'^{-1}\left[\beta \widehat{W}_1\left(k'_m, z_m; b^{(i)}\right)\right]$.

c. Use a solver to find k_m satisfying

$(1-\delta)k_m + z_m f\left(k_m\right) = c_m + k'_m$.

d. Find value function on the grid

$\widehat{v}_m \equiv u(c_m) + \beta \widehat{W}\left(k'_m, z_m; b^{(i)}\right)$.

Step 2. Computation of b that fits value function on the grid.

Run a regression to find $\widehat{b}$:

$\widehat{b} = \arg\min_{b} \sum_{m=1}^{M} \|\widehat{v}_m - \widehat{V}(k_m, z_m; b)\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if

$\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k_m)^{(i)} - (k_m)^{(i-1)}}{(k_m)^{(i-1)}} \right| < 10^{-9}$.

b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.

A4: Policy function iteration using envelope condition (PI-ECM)

```
Algorithm 4. PI-ECM.

Initialization.
    a. Choose an approximating function $\widehat{V}(\cdot;b) \approx V$ and $\widehat{K}(\cdot;v) \approx K$.
    b. Choose integration nodes, $\varepsilon_j$, and weights, $\omega_j$, $j = 1,...,J$.
    c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
    d. Make an initial guess on $v^{(1)}$ and $b^{(1)}$.

Iterative cycle. Computation of a solution.
    At iteration $i$, perform the following steps:
Step 1. Computation of values of $V$ on the grid.
    For $m = 1,...,M,$
    a. Use $v^{(i)}$ to compute $k'_m = \widehat{K}\left(k_m, z_m; v^{(i)}\right)$.
    b. Compute the corresponding values of $k'_m$ using
    $c_m = (1-\delta)k_m + z_m f(k_m) - k'_m$.
    c. Iterate on $b^{(i)}$ to find an approximate solution to
    $\widehat{V}\left(k_m, z_m; b^{(i)}\right) = u(c_m) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}\left(k'_m, z_m^{\rho} \exp(\varepsilon_j); b^{(i)}\right)$.
    d. Find policy function on the grid
    $\widehat{k}'_m = (1-\delta)k_m + z_m f(k_m) - u'^{-1}\left[\frac{\widehat{V}_1\left(k_m, z_m; b^{(i)}\right)}{1-\delta+z_m f'(k_m)}\right]$.
Step 2. Computation of $v$ that fits the value function on the grid.
    Run a regression to find $\widehat{v}$ :
    $\widehat{v} = \arg\min_{v} \sum_{m=1}^{M} \|\widehat{k}'_m - \widehat{K}(k_m, z_m; v)\|$.
Step 3. Convergence check and fixed-point iteration.
    a. Check for convergence for $i \geq 2$: end Step 2 if
    $\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}} \right| < 10^{-9}$.
    b. Use damping with $\xi = 1$ to compute $v^{(i+1)} = (1-\xi)v^{(i)} + \xi \widehat{v}$.
```

A5: Policy function iteration using first-order condition (PI-FOC)

```
**Algorithm 5. PI-FOC.**

*Initialization.*
    a. Choose an approximating function $\widehat{V}(\cdot;b) \approx V$ and $\widehat{K}(\cdot;v) \approx K$.
    b. Choose integration nodes, $\varepsilon_j$, and weights, $\omega_j$, $j = 1,...,J$.
    c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
    d. Make an initial guess on $v^{(1)}$ and $b^{(1)}$.

*Iterative cycle. Computation of a solution.*
    At iteration $i$, perform the following steps:
*Step 1. Computation of values of $V$ on the grid.*
    For $m = 1,...,M$,
    a. Use $v^{(i)}$ to compute $k'_m = \widehat{K}\left(k_m, z_m; v^{(i)}\right)$.
    b. Compute the corresponding values of $k'_m$ using
    $c_m = (1-\delta)k_m + z_m f\left(k_m\right) - k'_m$.
    c. Iterate on $b^{(i)}$ to find an approximate solution to
    $\widehat{V}\left(k_m, z_m; b^{(i)}\right) = u\left(c_m\right) + \beta \sum_{j=1}^{J} \omega_j \widehat{V}\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.
    d. Find policy function on the grid
$u'\left((1-\delta)k_m + z_m f\left(k_m\right) - \widehat{k}'_m\right) = \beta \sum_{j=1}^{J} \omega_j \widehat{V_1}\left(\widehat{k}'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.

*Step 2. Computation of $v$ that fits the value function on the grid.*
    Run a regression to find $\widehat{v}$ :
    $\widehat{v} = \arg\min_{v} \sum_{m=1}^{M} \|\widehat{k}'_m - \widehat{K}(k_m, z_m; v)\|$.

*Step 3. Convergence check and fixed-point iteration.*
    a. Check for convergence for $i \geq 2$: end Step 2 if
    $\frac{1}{M} \sum_{m=1}^{M} \left| \frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}} \right| < 10^{-9}$.
    b. Use damping with $\xi = 1$ to compute $v^{(i+1)} = (1-\xi)v^{(i)} + \xi\widehat{v}$.
```

A6: Euler equation algorithm parameterizing Q.

Algorithm 6. Euler equation algorithm parameterizing V_1.

Initialization.

a. Choose an approximating function $\widehat{V}_1(\cdot;b) \approx V_1$.

b. Choose integration nodes, ε_j, and weights, ω_j, $j = 1,...,J$.

c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.

d. Make an initial guess on $b^{(1)}$.

Iterative cycle. Computation of a solution.

At iteration i, perform the following steps:

Step 1. Computation of values of V_1 on the grid.

For $m = 1,...,M$,

a. Use $b^{(i)}$ to compute $\widehat{V}_1\left(k_m, z_m; b^{(i)}\right)$.

b. Find k'_m using

$k'_m = (1-\delta)\, k_m + z_m f\,(k_m) - u'^{-1}\left(\frac{\widehat{V}_1\left(k_m, z_m; b^{(i)}\right)}{1-\delta+z_m f\,(k_m)}\right)$

c. Find the values of v_m on the grid

$\widehat{v}_{1,m} \equiv \beta\left[1-\delta+zf'\,(k_m)\right] \sum_{j=1}^{J} \omega_j \widehat{V}_1\left(k'_m, z_m^{\rho} \exp\left(\varepsilon_j\right); b^{(i)}\right)$.

Step 2. Computation of b that fits the V_1 function on the grid.

Run a regression to find $\widehat{b}$:

$\widehat{b} = \arg\min_{b} \sum_{m=1}^{M} \left\|\widehat{v}_{1,m} - \widehat{V}_1(k_m, z_m; b)\right\|$.

Step 3. Convergence check and fixed-point iteration.

a. Check for convergence for $i \geq 2$: end Step 2 if

$$\frac{1}{M} \sum_{m=1}^{M} \left|\frac{(k'_m)^{(i)} - (k'_m)^{(i-1)}}{(k'_m)^{(i-1)}}\right| < 10^{-9}.$$

b. Use damping with $\xi = 1$ to compute $b^{(i+1)} = (1-\xi) b^{(i)} + \xi \widehat{b}$.

A7. Euler equation algorithm parameterizing K

Algorithm 7. Euler equation algorithm parameterizing K.

```
Initialization.
    a. Choose approximating functions $\widehat{K}(\cdot;v) \approx K$.
    b. Choose integration nodes, $\varepsilon_j$, and weights, $\omega_j$, $j = 1,...,J$.
    c. Construct grid $\Gamma = \{k_m, z_m\}_{m=1}^{M}$.
    d. Make an initial guess on $v^{(1)}$.
Step 1. Computation of values of $\widehat{k}'$ on the grid.
    For $m = 1,...,M$,
    a. Use $v^{(i)}$ to compute $\widehat{K}\left(k_m, z_m; v^{(i)}\right)$ and
    $k''_{m,j} = \widehat{K}\left(\widehat{K}\left(k_m, z_m; v^{(i)}\right), z_m^{\rho}\exp\left(\epsilon_j\right); v^{(i)}\right)$, $j = 1,...,J$.
    b. Find $c'_{m,j}$ using
    $c'_{m,j} = (1-\delta)\widehat{K}\left(k_m, z_m; v^{(i)}\right) + z_m f\left(\widehat{K}\left(k_m, z_m; v^{(i)}\right)\right) - k''_{m,j}$.
    c. Find the values of $c_m$ on the grid
    $u'\left(c_m\right) = \beta \sum_{j=1}^{J} \omega_j u'\left(c'_{m,j}\right)\left[1-\delta + z_m^{\rho}\exp\left(\epsilon_j\right) f'\left(\widehat{K}\left(k_m, z_m; v^{(i)}\right)\right)\right]$.
    d. Find the values of $k'_m$ on the grid
    $\widehat{k}'_m = (1-\delta)k_m + z_m f\left(k_m\right) - c_m$.
Step 2. Computation of $v$ that fits the capital function on the grid.
    a. Run a regression to find $\widehat{v}$:
    $\widehat{v} = \arg\min_{v} \sum_{m=1}^{M} \|\widehat{k}'_m - \widehat{K}\left(k_m, z_m; v^{(i)}\right)\|$.
    b. Use damping with $\xi = 1$ to compute $v^{(i+1)} = (1-\xi)v^{(i)} + \xi\widehat{v}$.
```

Appendix B: An Euler equation algorithm for solving a new Keynesian model

We consider a new Keynesian model studied in Maliar and Maliar (2015). This is a stylized new Keynesian model with Calvo-type price frictions and a Taylor (1993) rule.

The model. The economy is populated by households, final-good firms, intermediate-good firms, monetary authority and government.

Households. The representative household solves

$$\max_{\{C_t,L_t,B_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp\left(\eta_{u,t}\right) \left[\frac{C_t^{1-\gamma}-1}{1-\gamma} - \exp\left(\eta_{L,t}\right) \frac{L_t^{1+\vartheta}-1}{1+\vartheta} \right] \tag{B.1}$$

$$\text{s.t. } P_t C_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} + T_t = B_{t-1} + W_t L_t + \Pi_t, \tag{B.2}$$

where $(B_0, \eta_{u,0}, \eta_{L,0}, \eta_{B,0})$ is given; C_t, L_t, and B_t are consumption, labor and nominal bond holdings, respectively; P_t, W_t and R_t are the commodity price, nominal wage and (gross) nominal interest rate, respectively; $\eta_{u,t}$ and $\eta_{L,t}$ are exogenous preference shocks to the overall momentary utility and disutility of labor, respectively; $\eta_{B,t}$ is an exogenous premium in the return to bonds; T_t is lump-sum taxes; Π_t is the profit of intermediate-good firms; $\beta \in (0,1)$ is the discount factor; $\gamma > 0$ and $\vartheta > 0$ are the utility-function parameters. The processes for shocks are

$$\eta_{u,t+1} = \rho_u \eta_{u,t} + \epsilon_{u,t+1}, \qquad \epsilon_{u,t+1} \sim \mathcal{N}\left(0, \sigma_u^2\right), \tag{B.3}$$

$$\eta_{L,t+1} = \rho_L \eta_{L,t} + \epsilon_{L,t+1}, \qquad \epsilon_{L,t+1} \sim \mathcal{N}\left(0, \sigma_L^2\right), \tag{B.4}$$

$$\eta_{B,t+1} = \rho_B \eta_{B,t} + \epsilon_{B,t+1}, \qquad \epsilon_{B,t+1} \sim \mathcal{N}\left(0, \sigma_B^2\right), \tag{B.5}$$

where ρ_u, ρ_L, ρ_B are the autocorrelation coefficients, and σ_u, σ_L, σ_B are the standard deviations of disturbances.

Final-good firms. Perfectly competitive final-good firms produce final goods using intermediate goods. A final-good firm buys $Y_t(i)$ of an intermediate good $i \in [0,1]$ at price $P_t(i)$ and sells Y_t of the final good at price P_t in a perfectly competitive market. The profit-maximization problem is

$$\max_{Y_t(i)} \quad P_t Y_t - \int_0^1 P_t(i) Y_t(i)\, di \tag{B.6}$$

$$\text{s.t. } Y_t = \left(\int_0^1 Y_t(i)^{\frac{\varepsilon-1}{\varepsilon}} di \right)^{\frac{\varepsilon}{\varepsilon-1}}, \tag{B.7}$$

where (B.7) is a Dixit-Stiglitz aggregator function with $\varepsilon \geq 1$.

Intermediate-good firms. Monopolistic intermediate-good firms produce intermediate goods using labor and are subject to sticky prices. The firm i produces the intermediate good i. To choose labor in each period t, the firm i minimizes the nominal total cost, TC (net of government subsidy v),

$$\min_{L_t(i)} \quad \text{TC}(Y_t(i)) = (1-v) W_t L_t(i) \tag{B.8}$$

$$\text{s.t. } Y_t(i) = \exp(\eta_{a,t}) L_t(i), \tag{B.9}$$

$$\eta_{a,t+1} = \rho_a \eta_{a,t} + \epsilon_{a,t+1}, \qquad \epsilon_{a,t+1} \sim \mathcal{N}\left(0, \sigma_a^2\right), \tag{B.10}$$

where $L_t(i)$ is the labor input; $\exp(\eta_{a,t})$ is the productivity level; ρ_a is the autocorrelation coefficient; and σ_a is the standard deviation of the disturbance. The firms are subject to Calvo-type price setting: a fraction $1-\theta$ of the firms sets prices optimally, $P_t(i) = \widetilde{P}_t$, for $i \in [0,1]$, and the fraction θ is not allowed to change the price and maintains the same price as in the previous period, $P_t(i) = P_{t-1}(i)$, for $i \in [0,1]$. A reoptimizing firm $i \in [0,1]$ maximizes the current value of the profit over the time when $\widetilde{P}_t$ remains effective,

$$\max_{\widetilde{P}_t} \quad \sum_{j=0}^{\infty} \beta^j \theta^j E_t \left\{ \Lambda_{t+j} \left[\widetilde{P}_t Y_{t+j}(i) - P_{t+j} \text{mc}_{t+j} Y_{t+j}(i) \right] \right\} \tag{B.11}$$

$$\text{s.t. } Y_t(i) = Y_t \left(\frac{P_t(i)}{P_t} \right)^{-\varepsilon}, \tag{B.12}$$

where (B.12) is the demand for an intermediate good i following from (B.6), (B.7); Λ_{t+j} is the Lagrange multiplier on the household's budget constraint (B.2); mc_{t+j} is the real marginal cost of output at time $t+j$ (which is identical across the firms).

Government. Government finances a stochastic stream of public consumption by levying lump-sum taxes and by issuing nominal debt. The government budget constraint is

$$T_t + \frac{B_t}{\exp\left(\eta_{B,t}\right) R_t} = P_t \frac{\overline{G} Y_t}{\exp\left(\eta_{G,t}\right)} + B_{t-1} + vW_t L_t, \tag{B.13}$$

where $\frac{\overline{G}Y_t}{\exp(\eta_{G,t})} = G_t$ is government spending, vW_tL_t is the subsidy to the intermediate-good firms, and $\eta_{G,t}$ is a government-spending shock,

$$\eta_{G,t+1} = \rho_G \eta_{G,t} + \epsilon_{G,t+1}, \qquad \epsilon_{G,t+1} \sim \mathcal{N}\left(0, \sigma_G^2\right), \tag{B.14}$$

where ρ_R is the autocorrelation coefficient, and σ_R is the standard deviation of disturbance.

Monetary authority. The monetary authority follows a Taylor rule with a zero lower bound (ZLB) on the nominal interest rate:

$$R_t = \max\left\{ R_* \left(\frac{R_{t-1}}{R_*}\right)^{\mu} \left[\left(\frac{\pi_t}{\pi_*}\right)^{\phi_\pi} \left(\frac{Y_t}{Y_{N,t}}\right)^{\phi_y}\right]^{1-\mu} \exp\left(\eta_{R,t}\right), 1 \right\}, \tag{B.15}$$

where R_* is the long-run value of the gross nominal interest rate; π_* is the target inflation; $Y_{N,t}$ is the natural level of output; and $\eta_{R,t}$ is a monetary shock,

$$\eta_{R,t+1} = \rho_R \eta_{R,t} + \epsilon_{R,t+1}, \qquad \epsilon_{R,t+1} \sim \mathcal{N}\left(0, \sigma_R^2\right), \tag{B.16}$$

where ρ_R is the autocorrelation coefficient, and σ_R is the standard deviation of disturbance.

Natural level of output. The natural level of output $Y_{N,t}$ is the level of output in an otherwise identical economy but without distortions. It is a solution to the following planner's problem

$$\max_{\{C_t, L_t\}_{t=0,\ldots,\infty}} E_0 \sum_{t=0}^{\infty} \beta^t \exp(\eta_{u,t}) \left[\frac{C_t^{1-\gamma} - 1}{1-\gamma} - \exp(\eta_{L,t}) \frac{L_t^{1+\vartheta} - 1}{1+\vartheta} \right] \tag{B.17}$$

$$\text{s.t. } C_t = \exp(\eta_{a,t}) L_t - G_t, \tag{B.18}$$

where $G_t = \frac{\overline{G} Y_t}{\exp(\eta_{G,t})}$ is given, and $\eta_{u,t+1}$, $\eta_{L,t+1}$, $\eta_{a,t+1}$, and $\eta_{G,t}$ follow the processes (B.3), (B.4), (B.10), and (B.14), respectively.

Parameterization and implementation details Most of the parameters are calibrated using the estimates of Del Negro et al. (2007, Table 1, column ”DSGE posterior”); namely, we assume $\gamma = 1$ and $\vartheta = 2.09$ in the utility function (B.1); $\phi_y = 0.07$, $\phi_\pi = 2.21$, and $\mu = 0.82$ in the Taylor rule (B.15); $\varepsilon = 4.45$ in the production function of the final-good firm (B.7); $\theta = 0.83$ (the fraction of the intermediate-good firms affected by price stickiness); $\overline{G} = 0.23$ in the government budget constraint (B.13); and $\rho_u = 0.92$, $\rho_G = 0.95$, $\rho_L = 0.25$, $\sigma_u = 0.54\%$, $\sigma_G = 0.38\%$, $\sigma_L = 18.21\%$ (the latter is a lower estimate of Del Negro et al., 2007, Table 1, column ”DSGE posterior”), and $\sigma_L = 40.54\%$ (an average estimate of Del Negro et al., 2007) in the processes for shocks (B.3), (B.14) and (B.4). From Smets and Wouters (2007), we take the values of $\rho_a = 0.95$, $\rho_B = 0.22$, $\rho_R = 0.15$, $\sigma_a = 0.45\%$, $\sigma_B = 0.23\%$ and $\sigma_R = 0.28\%$ in the processes for shocks (B.10), (B.5) and (B.16). We set the discount factor at $\beta = 0.99$. To parameterize the Taylor rule (B.15), we use the steady-state interest rate $R_* = \frac{\pi_*}{\beta}$, and we consider two alternative values of the target inflation, $\pi_* = 1$ (a zero net inflation target) and $\pi_* = 1.0598$ (this estimate comes from Del Negro et al., 2007).

To approximate the equilibrium policy rules, we use a family of ordinary polynomials. To compute conditional expectations in Euler equations (2.8), (2.9) and (2.12), we use a monomial integration rule (either a formula with $2N$ nodes or the one with $2N^2 + 1$ nodes); see Judd, Maliar and Maliar (Quantitative Economics, 2011) for a detailed description of the monomial integration formulas. To compute a solution, we use two alternative grids: one is a random grid composed of uniformely distributed points (i.e., we make independent random draws for all variables in the grid within the given range), and the other is a quasi-Monte Carlo grid (namely, a Sobol grid that fills in a given multidimensional hypercube). In Step 2b, the damping parameter is set at $\xi = 0.1$, and the convergence parameter is set at $\varpi = 10^{-7}$. The solution algorithm is described in Appendix.

Computational method In Appendix B, we describe an Euler equation algorithm, which we use to solve the new Keynesian model described in Section 3.

An algorithm iterating on the Euler equation

Step 1. Initialization.

```
a. Choose $\left(\Delta_{-1}, R_{-1}, \eta_{u,0}, \eta_{L,0}, \eta_{B,0}, \eta_{a,0}, \eta_{R,0}, \eta_{G,0}\right)$ and $T$.
b. Draw $\{\epsilon_{u,t+1}, \epsilon_{L,t+1}, \epsilon_{B,t+1}, \epsilon_{a,t+1}, \epsilon_{R,t+1}, \epsilon_{G,t+1}\}_{t=0,...,T-1}$.
   Compute and fix $\left\{\eta_{u,t+1}, \eta_{L,t+1}, \eta_{B,t+1}, \eta_{a,t+1}, \eta_{R,t+1}, \eta_{G,t+1}\right\}_{t=0,...,T-1}$.
c. Choose approximating functions $S \approx \widehat{S}\left(\cdot; b^S\right)$, $F \approx \widehat{F}\left(\cdot; b^F\right)$, $\mathrm{MU} \approx \widehat{\mathrm{MU}}\left(\cdot; b^{\mathrm{MU}}\right)$.
d. Make initial guesses on $b^S$, $b^F$, $b^{\mathrm{MU}}$.
e. Choose integration nodes, $\left\{\epsilon_{u,j}, \epsilon_{L,j}, \epsilon_{B,j}, \epsilon_{a,j}, \epsilon_{R,j}, \epsilon_{G,j}\right\}_{j=1,...,J}$ and weights, $\left\{\omega_j\right\}_{j=1,...,J}$.
f. Constuct a grid $\Gamma = \left\{\Delta_m, R_m, \eta_{u,m}, \eta_{L,m}, \eta_{B,m}, \eta_{a,m}, \eta_{R,m}, \eta_{G,m}\right\}_{m=1,...,M} \equiv \{x_m\}_{m=1,...,M}$
```

Step 2. Computation of a solution for S, F, MU.

```
a. At iteration $i$, for $m = 1, ..., M$, compute
   – $S_m = \widehat{S}\left(x_m; b^S\right)$, $F_m = \widehat{F}\left(x_m; b^F\right)$, $C_m = \left[\widehat{\mathrm{MU}}\left(x_m; b^{\mathrm{MU}}\right)\right]^{-1/\gamma}$;
   – $\pi_m$ from $\frac{S_m}{F_m} = \left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$ and $\Delta'_m = \left[(1-\theta)\left[\frac{1-\theta\pi_m^{\varepsilon-1}}{1-\theta}\right]^{\frac{\varepsilon}{\varepsilon-1}} + \theta\frac{\pi_m^{\varepsilon}}{\Delta_m}\right]^{-1}$;
   – $Y_m = \left(1 - \frac{\overline{G}}{\exp(\eta_{G,m})}\right)^{-1} C_m$, and $L_m = Y_m\left[\exp\left(\eta_{a,m}\right)\Delta'_m\right]^{-1}$;
   – $Y_{N,m} = \left[\frac{\exp(\eta_{a,m})^{1+\vartheta}}{[\exp(\eta_{G,m})]^{-\gamma}\exp(\eta_{L,m})}\right]^{\frac{1}{\vartheta+\gamma}}$;
   – $R'_m = \max\{1, \Phi_m\}$, $\Phi_m = R_*\left(\frac{R_m}{R_*}\right)^{\mu}\left[\left(\frac{\pi_m}{\pi_*}\right)^{\phi_\pi}\left(\frac{Y_m}{Y_{N,m}}\right)^{\phi_y}\right]^{1-\mu}\exp\left(\eta_{R,m}\right)$;
   – $x'_{m,j} = \left(\Delta'_m, R'_m, \eta'_{u,m,j}, \eta'_{L,m,j}, \eta'_{B,m,j}, \eta'_{a,m,j}, \eta'_{R,m,j}, \eta'_{G,m,j}\right)$ for all $j$;
   – $S'_{m,j} = \widehat{S}\left(x'_{m,j}; b^S\right)$, $F'_{m,j} = \widehat{F}\left(x'_{m,j}; b^F\right)$, $C'_{m,j} = \left[\widehat{\mathrm{MU}}\left(x'_{m,j}; b^{\mathrm{MU}}\right)\right]^{-1/\gamma}$;
   – $\pi'_{m,j}$ from $\frac{S'_{m,j}}{F'_{m,j}} = \left[\frac{1-\theta\left(\pi'_{m,j}\right)^{\varepsilon-1}}{1-\theta}\right]^{\frac{1}{1-\varepsilon}}$;
   – $\widehat{S}_m = \frac{\exp\left(\eta_{u,m}+\eta_{L,m}\right)}{\exp(\eta_{a,m})}L_m^{\vartheta}Y_m + \beta\theta\sum_{j=1}^{J}\omega_j \cdot \left\{\left(\pi'_{m,j}\right)^{\varepsilon}S'_{m,j}\right\}$,
   – $\widehat{F}_m = \exp\left(\eta_{u,m}\right)C_m^{-\gamma}Y_m + \beta\theta\sum_{j=1}^{J}\omega_j \cdot \left\{\left(\pi'_{m,j}\right)^{\varepsilon-1}F'_{m,j}\right\}$,
   – $\widehat{C}_m^{-\gamma} = \frac{\beta\exp(\eta_{B,m})R'_m}{\exp(\eta_{u,m})}\sum_{j=1}^{J}\omega_j \cdot \left[\frac{\left(C'_{m,j}\right)^{-\gamma}\exp\left(\eta'_{u,m,j}\right)}{\pi'_{m,j}}\right]$.
b. Find $b^S$, $b^F$, $b^{\mathrm{MU}}$ that solve the system in Step 2a.
   – Get: $\widehat{b}^S \equiv \underset{b^S}{\mathrm{argmin}}\sum_{m=1}^{M}\left\|\widehat{S}_m - \widehat{S}\left(x_m; b^S\right)\right\|$. Similarly, get $\widehat{b}^F$ and $\widehat{b}^{\mathrm{MU}}$.
   – Use damping to compute $b^{(i+1)} = (1-\xi)\, b^{(i)} + \xi\widehat{b}$, where $b \equiv \left(\widehat{b}^S, \widehat{b}^F, \widehat{b}^{\mathrm{MU}}\right)$.
   – Check for convergence: end Step 2 if
```

$$\frac{1}{M}\max\left\{\sum_{m=1}^{M}\left|\frac{(S_m)^{(i+1)}-(S_m)^{(i)}}{(S_m)^{(i)}}\right|, \sum_{m=1}^{M}\left|\frac{(F_m)^{(i+1)}-(F_m)^{(i)}}{(F_m)^{(i)}}\right|, \sum_{m=1}^{M}\left|\frac{(\mathrm{MU}_m)^{(i+1)}-(\mathrm{MU}_m)^{(i)}}{(\mathrm{MU}_m)^{(i)}}\right|\right\} < \varpi.$$

We compute residuals on a stochastic simulation of $10,200$ observations (we eliminate the first 200 observations). In the test, we use a monomial rule $M2$ with $2 \cdot 6^2 + 1$ nodes which is more accurate than the other monomial rule $M1$.

B.2 Tables and Figures

Table B.1: Accuracy and speed of value iterative methods

Degree	L_1	L_∞	Julia CPU (s)	Matlab CPU (s)	Python CPU (s)
Conventional VFI (VFI)					
2nd	-3.83	-2.76	2.06	74.02	11.20
3rd	-4.97	-3.32	1.79	49.87	6.83
4th	-6.06	-4.03	1.88	43.08	6.06
5th	-7.00	-4.70	3.28	32.48	4.61
Envelope condition method (ECM)					
2nd	-3.83	-2.76	0.56	0.38	0.34
3rd	-4.97	-3.32	0.35	0.21	0.24
4th	-6.06	-4.03	0.57	0.22	0.27
5th	-7.00	-4.70	0.63	0.11	0.15
Endogenous grid method (EGM)					
2nd	-3.81	-2.76	1.03	59.89	4.11
3rd	-4.95	-3.34	0.76	46.19	2.82
4th	-6.06	-4.05	0.91	34.89	2.51
5th	-7.04	-4.73	0.99	27.05	1.89

[a] L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality conditions and test points (in $\log 10$) units on a stochastic simulation of 10,000 observations. CPU is the time necessary for computing a solution (in seconds)

Table B.2: Accuracy and speed of policy iterative methods

Degree	L_1	L_∞	Julia CPU (s)	Matlab CPU (s)	Python CPU (s)
Conventional PI					
2nd	-3.81	-2.76	0.37	47.13	5.66
3rd	-4.95	-3.34	1.58	39.30	5.07
4th	-6.06	-4.05	2.06	41.83	5.75
5th	-7.04	-4.73	3.28	32.54	4.44
Envelope condition method iterrating on policy functions (ECM-PI)					
2nd	-3.81	-2.76	0.32	0.38	0.42
3rd	-4.95	-3.34	0.37	0.15	0.34
4th	-6.06	-4.05	0.55	0.24	0.41
5th	-7.04	-4.73	0.64	0.11	0.27
Envelope condition method iterating on derivative of value function (ECM-DVF)					
2nd	-3.81	-2.76	0.74	0.68	0.61
3rd	-4.95	-3.34	1.12	0.66	0.65
4th	-6.06	-4.05	1.62	0.69	0.75
5th	-7.04	-4.73	4.05	0.73	0.81
Euler equation method					
2nd	-3.81	-2.76	0.40	0.46	0.38
3rd	-4.95	-3.34	0.37	0.28	0.27
4th	-6.06	-4.05	0.53	0.24	0.25
5th	-7.04	-4.73	0.77	0.11	0.12

[a] L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality conditions and test points (in $\log 10$) units on a stochastic simulation of 10,000 observations. CPU is the time necessary for computing a solution (in seconds)

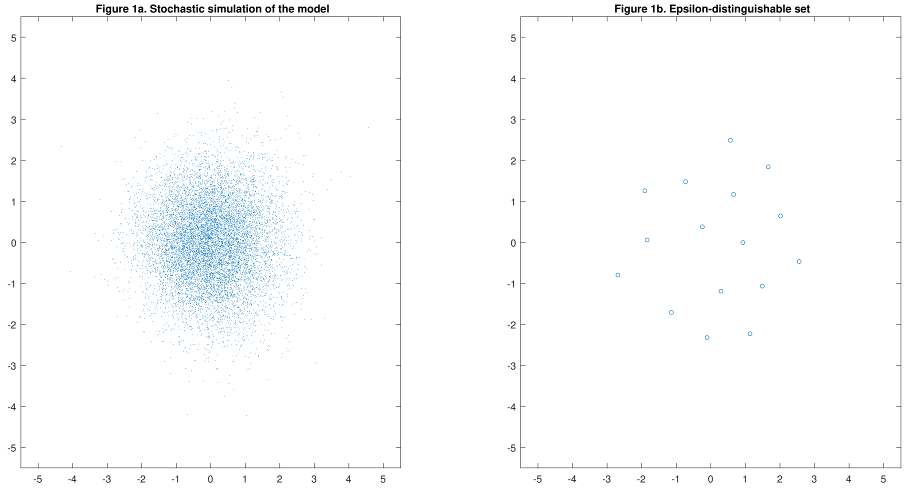


Figure B.1: Stochastic-simulation grid and epsilon-distinguishable set grid.

Table B.3: Accuracy and speed of a projection method for solving new Keynesian model

Degree	L_1	L_∞	Julia CPU (s)	Python CPU (s)	Matlab CPU (s)
			Inflation target $\pi_* = 1.0598$		
1st	-3.41	-1.94	0.74	0.96	0.79
2nd	-4.71	-3.13	3.28	2.24	2.65
3rd	-6.07	-4.25	14.30	9.18	7.49
4th	-6.73	-4.65	54.84	94.26	55.57
5th	-7.00	-5.47	902.95	1879.04	1120.17
			Inflation target $\pi_* = 1$		
1st	-3.12	-1.73	0.39	0.50	0.47
2nd	-4.40	-2.77	1.45	0.98	1.09
3rd	-5.71	-3.54	6.06	3.81	3.08
4th	-6.82	-4.89	23.62	38.44	22.45
5th	-7.01	-5.12	375.57	798.46	461.46
			Inflation target $\pi_* = 1$ with ZLB		
1st	-3.12	-1.73	0.41	0.45	0.50
2nd	-4.40	-2.16	1.45	0.87	1.06
3rd	-5.60	-2.15	5.86	3.70	2.95
4th	-6.17	-2.15	23.87	39.73	22.98
5th	-6.20	-2.15	347.89	822.97	455.39

[a] L_1 and L_∞ are, respectively, the average and maximum of absolute residuals across optimality conditions and test points (in $\log 10$) units on a stochastic simulation of 10,000 observations. CPU is the time necessary for computing a solution (in seconds)

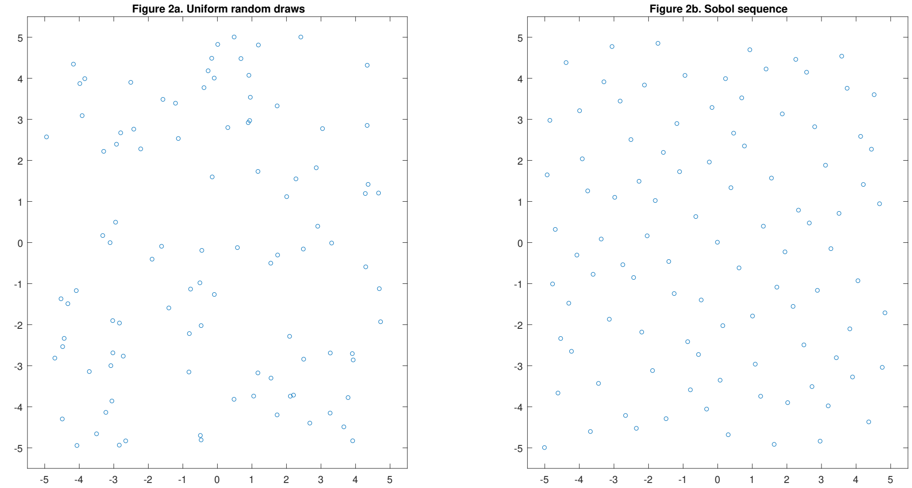


Figure B.2: Uniformly-spaced random grid and quasi-Monte Carlo grid.

Appendix C

Supplementary Material for Chapter 3

C.1 Data

Due to the sensitive nature of information on an individual's education data, one difficulty of any empirical work on higher education financing is the scarcity of publically available micro data. In conclusion to a discussion of how an economist might design a student loans program, Dynarski (2014) claims, "Designing such a program requires detailed data on individual earnings and borrowing, which are currently unavailable to researchers within and outside the government. If loan policy is to be firmly grounded in research, this gap in the data must be closed." While not as accurate as administrative data, we leverage several data sets on education and earnings outcomes in this paper.

High School and Beyond: The High School and Beyond survey is a longitudinal survey that follows individuals who were either sophmores or seniors in 1980 through 1992[1]. We did not have access to the restricted access micro data and associated transcript data, but Hendricks and Leukhina (2017), who did have access, published many relevant moments from this data set. We take advantage of this and use their empirical estimates for various parameters in our model.

[1] The individuals who were high school sophmores in 1980 were interviewed in 1980, 1982, 1984, 1986, and 1992 while the individuals who were high school seniors were only interviewed in 1980, 1982, 1984, and 1986

Panel Study of Income Dynamics: The Panel Study of Income Dynamics (PSID) is the longest running longitudinal survey in the world. As in Carroll and Samwick (1997), Guvenen (2009), and Hryshko (2012) the PSID will be used to estimate income dynamics — We successfully replicate estimated income processes from these papers and will use them as inputs to our model.

C.2 Tables and Figures

Tables

Parameter Name	**Value**	**Source**
β	0.98	Standard
r	0.04	
T	42	Retirement at 60
τ, λ	0.118, 1.07	Heathcote et al. (2017)
$\sigma^{HS,w}, \sigma^{CD,w}, \sigma^{CG,w}$	$\sqrt{0.011}, \sqrt{0.011}, \sqrt{0.0099}$	Guvenen (2009), Carroll and Samwick (1997)
$\sigma^{HS,y}, \sigma^{CD,y}, \sigma^{CG,y}$	$\sqrt{0.052}, \sqrt{0.052}, \sqrt{0.047}$	
$\phi_{HS}, \phi_{CD}, \phi_{CG}$	0.15, 0.15, 0.197	Hendricks and Leukhina (2017)
μ, w_{coll}	0.010, 3.55	
$\gamma_{\text{min}}, \gamma_1, \gamma_2$	0.47, 4.58, 2.10	
α_{am}, $\alpha_{m,GPA}$	2.87, 1.2	
α_{kq}, α_{mq}, α_{mz}, α_{qz}	0.0, -0.04, 0.46, -0.12	
μ_q, σ_q	0.53, 0.35	
μ_z, σ_z	0.32, 0.55	

Table C.1: Previous Knowledge Parameters

Parameter Name	**Value**	**Source**
$\bar{\theta}^{HS}$, $\bar{\theta}^{CG}$	8.73, 5.06	PSID (2019)
θ_1^{HS}, θ_2^{HS}, θ_3^{HS}	0.066, -0.088, 0.002	
θ_1^{CG}, θ_2^{CG}, θ_3^{CG}	0.324, -0.624, 0.039	
$\bar{T}_C$	6	College features
n_c	6	
n_{grad}	21	
$\underline{\mathrm{d}}$, $\underline{\mathrm{D}}$	-0.5, -1.975	

Table C.2: Externally Calibrated Parameters

Parameter Name	**Value**	**Target**
δ_c, δ_v	0.62, 1.15	Debt accumulation
U_{HS}, U_{CD}, U_{CG}	0.0, -2.97, -6.81	Enrollment
α_{km}, μ_k, σ_k	0.55, 2.57, 3.94	Fraction with debt
π_e, π_c	0.30, 1.58	Enrollment and dropout

Table C.3: Internally Calibrated Parameters

Moment Descriptions			**Data**	**Model**
Name	**Year**	**GPA Quartile**		
Time to degree	—	—	4.45	4.73
Education Fraction (HS)	—	—	0.52	0.53
Education Fraction (CD)	—	—	0.23	0.21
Education Fraction (CG)	—	—	0.25	0.26
Enrollment	—	Quartile 1	0.22	0.18
	—	Quartile 2	0.35	0.35
	—	Quartile 3	0.55	0.55
	—	Quartile 4	0.81	0.81
Completion	—	Quartile 1	0.02	0.03
	—	Quartile 2	0.08	0.13
	—	Quartile 3	0.28	0.30
	—	Quartile 4	0.60	0.60
College Earnings	—	Quartile 1	0.65	0.85
	—	Quartile 2	0.58	0.70
	—	Quartile 3	0.54	0.59
	—	Quartile 4	0.51	0.43
Average debt	Year 1	—	0.35	0.33
	Year 2	—	0.59	0.50
	Year 3	—	0.78	0.75
	Year 4	—	0.95	1.06
Fraction with debt	Year 1	—	0.26	0.11
	Year 2	—	0.34	0.27
	Year 3	—	0.41	0.42
	Year 4	—	0.48	0.52
Drop out	Year 1	—	0.16	0.15
	Year 2	—	0.14	0.10
	Year 3	—	0.08	0.07
	Year 4	—	0.05	0.05
	Year 5	—	0.02	0.03

Table C.4: Data and model moments

	AMR	**LC**	**IDR**
Enrollment rate	0.48	0.48	0.49
Completion rate	0.27	0.27	0.27
Graduation rate	0.56	0.56	0.56

Table C.5: Student outcomes under different versions of our economy

	AMR	**LC**	**IDR**
Fraction with debt	0.46	0.46	0.70
Average debt	$11,750	$11,750	$13,740

Table C.6: Student debt decisions under plans

	AMR	**LC**	**IDR**
Subsidy rate	-0.13	-0.12	-0.06
Government subsidy (per person)	-$0.085	-$0.077	-$0.072

Table C.7: Government subsidy rates and per person subsidy

	AMR	**IDR**
Quartile 1	-0.13	0.11
Quartile 2	-0.13	-0.01
Quartile 3	-0.13	-0.07
Quartile 4	-0.13	-0.11

Table C.8: Government subsidy rates by HS GPA quartile

Figures

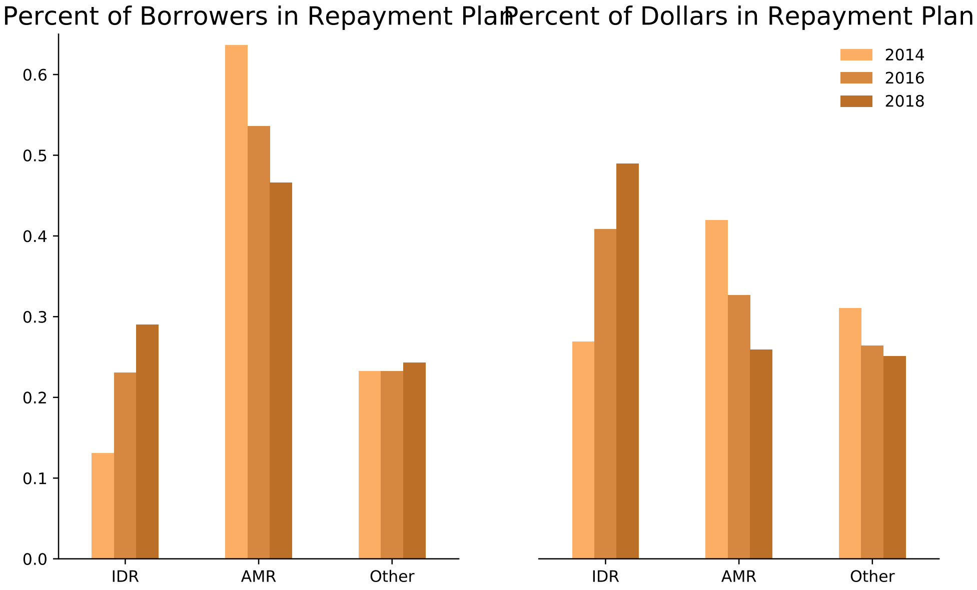


Figure C.1: This figure shows how the enrollment in IDR has changed over last 4 years

Source: U.S. Department of Education, Federal Student Aid Data Center, Federal Student Loan Portfolio.

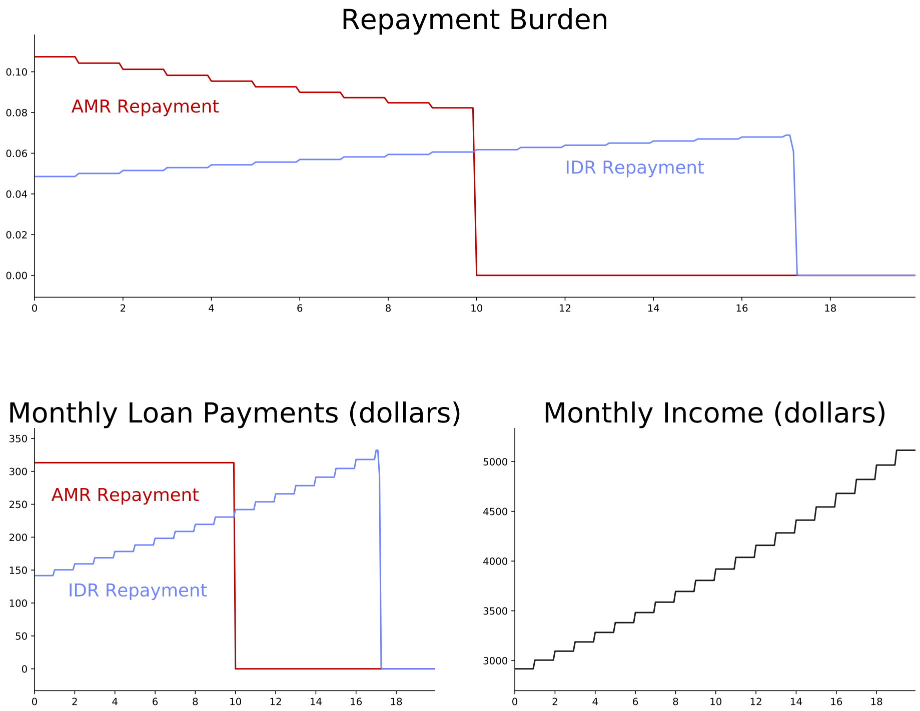


Figure C.2: This figure shows how the repayment burden and repayment amount differ for the AMR and IDR student loan repayment plans. It also plots the history of income for the example individual.

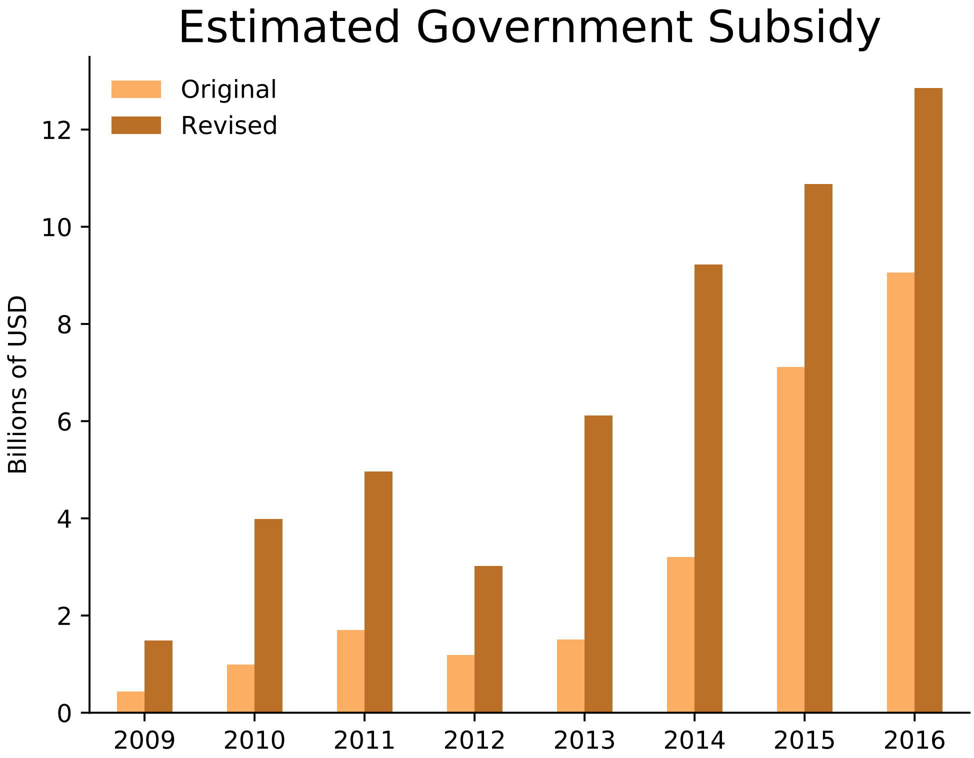


Figure C.3: This figure shows the original subsidy estimate alongside the 2017 updated estimate. Source: GAO analysis of the U.S. Department of Educations' 2011-2017 budget estimates

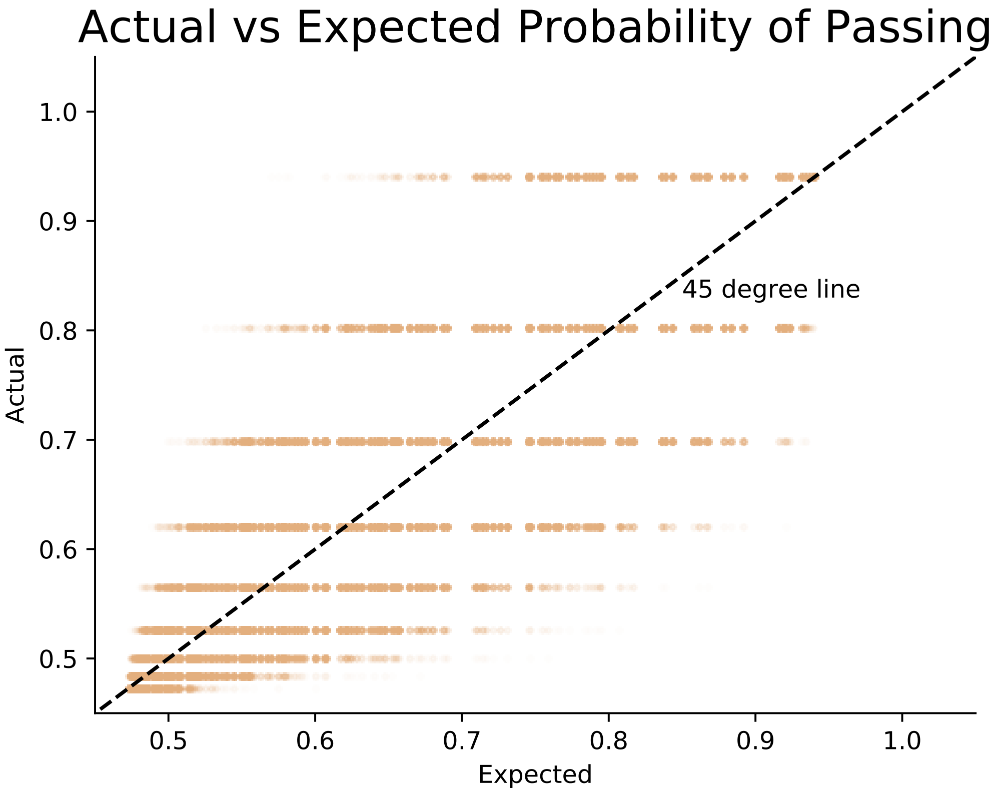


Figure C.4: This figure shows the difference between the value function under the IDR and AMR plans for a college dropout 3 periods after leaving college. Behind the line is the probability distribution over income levels is plotted.

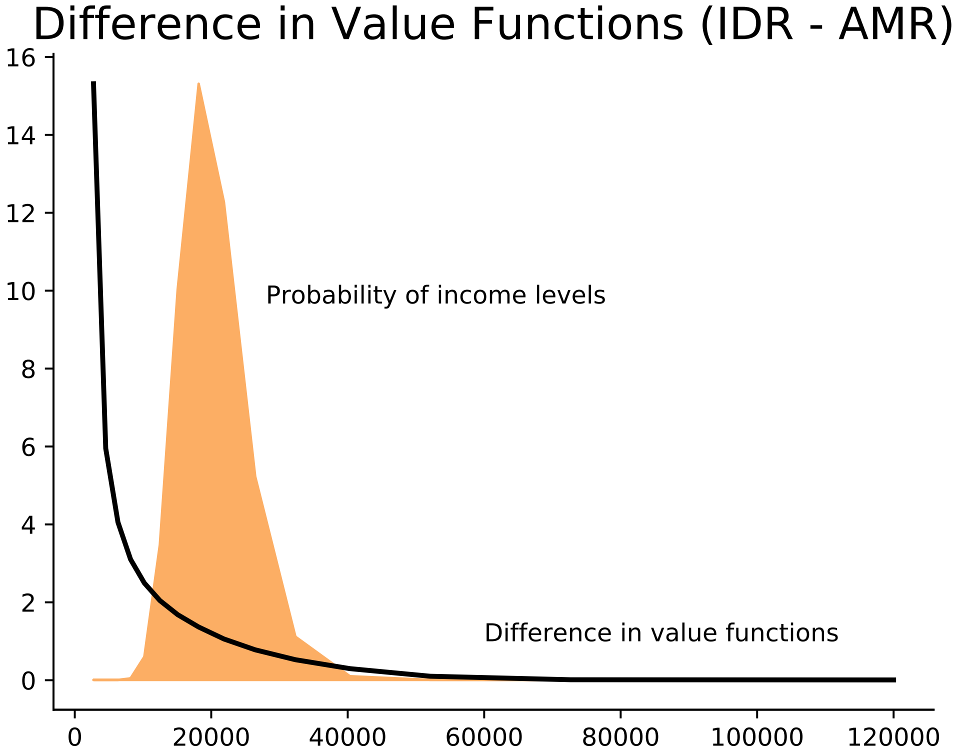


Figure C.5: This figure shows the difference between the value function under the IDR and AMR plans for a college dropout 3 periods after leaving college. Behind the line is the probability distribution over income levels is plotted.

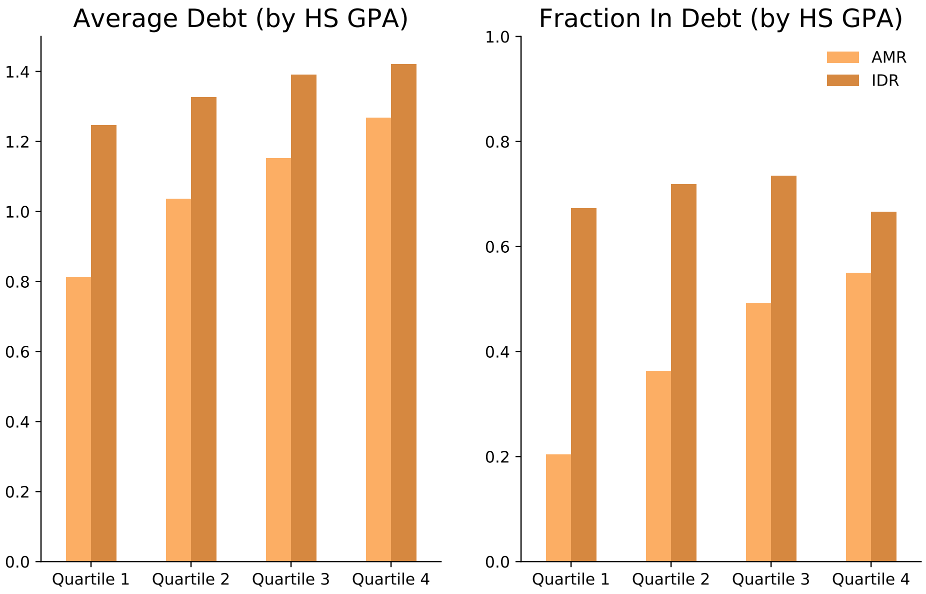


Figure C.6: This figure shows the differences in average debt among those who have positive student loans and the fraction of college students who borrow for the AMR and IDR plans

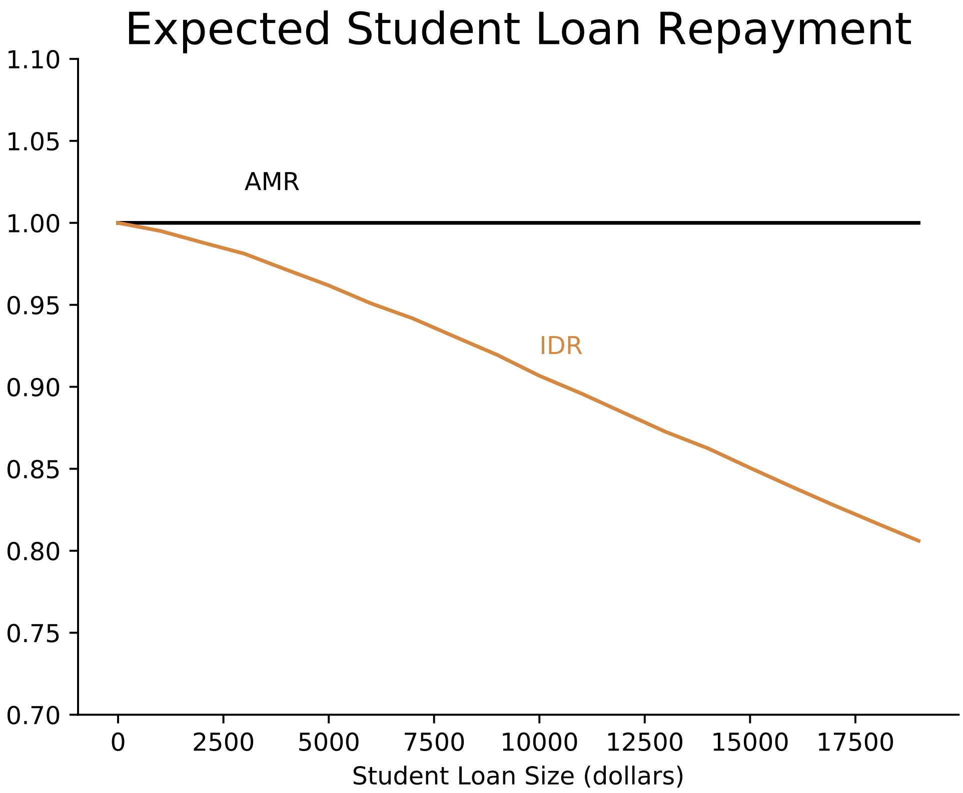


Figure C.7: This figure shows the fraction of a student loan that is expected to be repaid for AMR and IDR across different loan sizes.

Bibliography

ACHDOU, Y., J. HAN, J.-M. LASRY, P.-L. LIONS, AND B. MOLL (2017): "Income and Wealth Distribution in Macroeconomics: A Continuous-Time Approach," Working Paper 23732, National Bureau of Economic Research. 33

ANDERSON, E. W. (2005): "The dynamics of risk-sensitive allocations," *J. Econ. Theory*, 125, 93–150. 3, 15

ARCIDIACONO, P. (2005): "Affirmative Action in Higher Education: How Do Admission and Financial Aid Rules Affect Future Earnings?" *Econometrica*, 73, 1477–1524. 49

ARELLANO, C. (2008): "Default Risk and Income Fluctuations in Emerging Economies," *American Economic Review*, 98. 19

ARELLANO, C., L. MALIAR, S. MALIAR, AND V. TSYRENNIKOV (2016): "Envelope condition method with an application to default risk models," *Journal of Economic Dynamics and Control*, 69, 436–459. 21, 29

ARUOBA, S. B., P. CUBA-BORDA, AND F. SCHORFHEIDE (2017): "Macroeconomic dynamics near the ZLB: A tale of two countries," Tech. rep., National Bureau of Economic Research. 43

ARUOBA, S. B. AND J. FERNÁNDEZ-VILLAVERDE (2015): "A comparison of programming languages in economics," *Journal of Economic Dynamics and Control*, 58, 265–273. 21, 31

BACKUS, D., A. FERRIERE, AND S. ZIN (2015): "Risk and ambiguity in models of business cycles," *J. Monet. Econ.*, 69, 42–63. 11

BACKUS, D. K., P. J. KEHOE, AND F. E. KYDLAND (1994): "Dynamics of the Trade Balance and the Terms of Trade: The J-Curve?" *Am. Econ. Rev.*, 84, 84–103. 4

BACKUS, D. K. AND G. W. SMITH (1993): "Consumption and real exchange rates in dynamic economies with non-traded goods," *J. Int. Econ.*, 35, 297–316. 13

Bansal, R. and A. Yaron (2004): *Risks for the Long Run: A Potential Resolution of Asset Pricing Puzzles on JSTOR*, Journal of Finance. 10

Barillas, F. and J. Fernández-Villaverde (2007): "A generalization of the endogenous grid method," *Journal of Economic Dynamics and Control*, 31, 2698–2712. 32

Barr, N., B. Chapman, L. Dearden, and S. Dynarski (Forthcoming): "Getting student financing right in the US: lessons from Australia and England," *Economics of Education Review*. 49, 50

Baxter, M. and M. J. Crucini (1995): "Business Cycles and the Asset Structure of Foreign Trade," *Int. Econ. Rev.*, 36, 821–854. 2

Borovička, J. (2016): "Survival and Long-Run Dynamics with Heterogeneous Beliefs under Recursive Preferences," . 3, 15

Cai, Y., B. Chapman, and Q. Wang (Forthcoming): "Repayment burdens of mortgage-style student loans in China and steps toward income-contingent loans," *Economics of Education Review*. 50

Carroll, C. D. (2006): "The method of endogenous gridpoints for solving dynamic stochastic optimization problems," *Economics letters*, 91, 312–320. 21, 31, 32

Carroll, C. D. and A. A. Samwick (1997): "The Nature of Precautionary Wealth," *Journal of Monetary Economics*, 40. 99

Chapman, B. (1994): "Income Contingent College Loans," *Journal of Economic Perspectives*. 49

Chapman, B. and L. Dearden (2017): "Conceptual and Empirical Issues for Alternative Student Loan Designs: The Significance of Loan Repayment Burdens for the United States," *The ANNALS of the American Academy of Political and Social Science*, 671, 249–268. 49

Chapman, B. and A. Harding (1993): "Australian Student Loans," *Australian Economic Review*, 1. 49

Chapman, B. and K. Lounkaew (2015): "An analysis of Stafford loan repayment burdens," *Economics of Education Review*, 45. 49

Chari, V. V., P. J. Kehoe, and E. R. McGrattan (2002): "Can Sticky Price Models Generate Volatile and Persistent Real Exchange Rates?" *Rev. Econ. Stud.*, 69, 533–563. 13

——— (2007): "Business Cycle Accounting," *Econometrica*, 75, 781–836. 2

Chatterjee, S. and F. Ionescu (2012): "Insuring Student Loans Against the Risk of College Failure," *Quantitative Economics*. 49, 50

Colacito, R. and M. Croce (2011): "Recursive allocations and wealth distribution with multiple goods : existence , survivorship , and dynamics," . 3, 10, 15

Colacito, R. and M. M. Croce (2013): "International Asset Pricing with Recursive Preferences," *J. Finance*, 68, 2651–2686. 3, 10, 11, 13, 15, 16

Colacito, R., M. M. M. Croce, S. Ho, and P. Howard (2014): "BKK the EZ Way International Long-Run Growth News and Capital Flows," *Working Paper*. 3

Collin-dufresne, P., M. Johannes, and L. A. Lochstoer (2015): "A Robust Numerical Method for Solving Risk-Sharing Problems with Recursive Preferences," 1–35. 3

Corsetti, G., L. Dedola, and S. Leduc (2008): "International Risk Sharing and the Transmission of Productivity Shocks," *Rev. Econ. Stud.*, 75, 443–473. 2, 13

Crucini, M. J. and M. Shintani (2008): "Persistence in law of one price deviations: Evidence from micro-data," *J. Monet. Econ.*, 55, 629–644. 16

Del Negro, M., F. Schorfheide, F. Smets, and R. Wouters (2007): "On the fit of new Keynesian models," *Journal of Business & Economic Statistics*, 25, 123–143. 39, 42

Dynarski, S. (2014): "An Economist's Perspective on Student Loans in the United States," *Economic Studies at Brookings: Working Paper Series*. 49, 98

Epstein, L. G. and S. E. Zin (1989): "Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework," *Econometrica*, 57, 937–969. 4

Ferreyra, M. M., C. Garriga, and R. Manuelli (2017): "General Equilibrium Effects of Higher Education Funding: The Case of Colombia and Brazil," . 49

Findeisen, S. and D. Sachs (2016): "Education and optimal dynamic taxation: The role of income-contingent student loans," *Journal of Public Economics*, 138, 1–21. 49

Government Accountability Office (2016): "Education Needs to Improve Its Income-Driven Repayment Plan Budget Estimates," . 47, 64, 66

Guerrieri, L. and M. Iacoviello (2015): "OccBin: A toolkit for solving dynamic models with occasionally binding constraints easily," *Journal of Monetary Economics*, 70, 22–38. 19

GUVENEN, F. (2009): "An Empirical Investigation of Labor Income Processes," *Review of Economic Dynamics*, 12. 57, 58, 99

HASANHODZIC, J. AND L. J. KOTLIKOFF (2013): "Generational risk-is it a big deal?: Simulating an 80-period olg model with aggregate shocks," Tech. Rep. 19179, National Bureau of Economic Research. 19

HEATHCOTE, J. AND F. PERRI (2002): "Financial autarky and international business cycles," *J. Monet. Econ.*, 49, 601–627. 2, 10

HEATHCOTE, J., K. STORESLETTEN, AND G. VIOLANTE (2017): "Optimal Tax Progressivity: An Analytical Framework," *Quarterly Journal of Economics*, 132. 99

HEIJDRA, B. J., F. KINDERMANN, AND L. S. REIJNDERS (2017): "Life in shackles? The quantitative implications of reforming the educational financing system," *Review of Economic Dynamics*, 25, 37 – 57. 49, 50

HENDRICKS, L. AND O. LEUKHINA (2017): "How risky is college investment?" *Review of Economic Dynamics*. 51, 52, 57, 58, 59, 61, 98, 99

HRYSHKO, D. (2012): "Labor income profiles are not heterogeneous: Evidence from income growth rates," *Quantitative Economics*. 58, 99

IONESCU, F. (2009): "The Federal Student Loan Program: Quantitative Implications for College Enrollment and Default Rates," *Review of Economic Dynamics*. 49

JI, Y. (2017): "Job Search under Debt: Aggregate Implications of Student Loans," *Working Paper*. 49

JUDD, K. L. (1998): *Numerical methods in economics*, MIT press. 36

JUDD, K. L., L. MALIAR, AND S. MALIAR (2011): "Numerically stable and accurate stochastic simulation approaches for solving dynamic economic models," *Quantitative Economics*, 2, 173–210. 39

——— (2017): "Lower Bounds on Approximation Errors to Numerical Solutions of Dynamic Economic Models," *Econometrica*, 85, 991–1012. 19

JURADO, K., S. C. LUDVIGSON, AND S. NG (2015): "Measuring Uncertainty," *Am. Econ. Rev.*, 105, 1177–1216. 11

KEANE, M. P. AND K. I. WOLPIN (2001): "The Effect of Parental Transfers and Borrowing Constraints on Educational Attainment," *International Economic Review*, 42, 1051–1103. 49

KEHOE, P. J. AND F. PERRI (2002): “International business cycles with endogenous incomplete markets,” *Econometrica*, 907–928. 2

KOLLMANN, R. (1995): “Consumption, real exchange rates and the structure of international asset markets,” *J. Int. Money Finance*, 14, 191–211. 13

——— (2015): “Exchange Rates Dynamics with Long-Run Risk and Recursive Preferences,” *Open Econ. Rev.*, 26, 175–196. 3, 10, 11, 15, 16

KOLLMANN, R., J. KIM, AND S. H. KIM (2011): “Solving the multi-country real business cycle model using a perturbation method,” *Journal of Economic Dynamics and Control*, 35, 203–206. 19

KOSE, M. A. AND K.-M. YI (2006): “Can the standard international business cycle model explain the relation between trade and comovement?” *J. Int. Econ.*, 68, 267–295. 2

KREPS, D. M. AND E. L. PORTEUS (1978): “Temporal Resolution of Uncertainty and Dynamic Choice Theory,” *Econometrica*, 46, 185–200. 4

KYDLAND, F. E. AND E. C. PRESCOTT (1982): “Time to Build and Aggregate Fluctuations,” *Econometrica*, 50, 1345–1370. 10

LASÉEN, S. AND L. E. O. SVENSSON (2011): “Anticipated Alternative Instrument-Rate Paths in Policy Simulations,” CEPR Discussion Papers 8176, C.E.P.R. Discussion Papers. 19

LEPETYUK, V., L. MALIAR, AND S. MALIAR (2017): “Should Central Banks Worry About Nonlinearities of their Large-Scale Macroeconomic Models?” Staff Working Papers 17-21, Bank of Canada. 20

LIU, J. (2016): “The Effects of Tuition and Student Loan Policies on College Outcomes and Lifetime Earnings,” *Publicly Accessible Penn Dissertations*, 1855. 49

LUCAS, R. (1976): “Econometric policy evaluation: A critique,” *Carnegie-Rochester Conference Series on Public Policy*, 1, 19–46. 60

LUO, M. AND S. MONGEY (2019): “Student Debt and Job Choice: Wages vs. Job Satisfaction,” *NBER Working Paper*. 49

MALIAR, L. AND S. MALIAR (2013): “Envelope Condition Method versus Endogenous Grid Method for Solving Dynamic Programming Problems,” *Economic Letters*, 120, 262–266. 31

——— (2014): "Numerical methods for large scale dynamic economic models," *Handbook of computational economics*, 3, 325–477. 20, 40

——— (2015): "Merging simulation and projection approaches to solve high-dimensional problems with an application to a new Keynesian model," *Quantitative Economics*, 6, 1–47. 21, 22, 37, 39, 40, 41, 42, 43

Niederreiter, H. (1992): *Random number generation and quasi-Monte Carlo methods*, SIAM. 42

Petrosky-Nadeau, N. and L. Zhang (Forthcoming): "Solving the DMP Model Accurately," *Quantitative Economics*. 19

PSID (2019): "Panel Study of Income Dynamics," *Produced and distributed by the Survey Research Center, Institute for Social Research, University of Michigan, Ann Arbor, MI*. 100

Rabanal, P., J. F. Rubio-Ramírez, and V. Tuesta (2011): "Cointegrated TFP processes and international business cycles," *J. Monet. Econ.*, 58, 156–171. 11

Santos, M. S. (1999): "Chapter 5 Numerical solution of dynamic economic models," *Handbook of Macroeconomics*, 1, 311–386. 36

Santos, M. S. and J. Rust (2004): "Convergence properties of policy iteration," *SIAM Journal on Control and Optimization*, 42, 2094–2115. 34

Smets, F. and R. Wouters (2007): "Shocks and Frictions in US Business Cycles: A Bayesian DSGE Approach," *American Economic Review*, 97, 586–606. 39

Tallarini, T. D. (2000): "Risk-sensitive real business cycles," *J. Monet. Econ.*, 45, 507–532. 11

Tretvoll, H. (2011): "Resolving the Consumption-Real Exchange Rate Anomaly with Recursive Preferences," . 3, 11, 13, 15, 16

——— (2015): "International Correlations and the Composition of Trade," . 3, 10, 15

——— (2018): "Real exchange rate variability in a two-country business cycle model," *Rev. Econ. Dyn.*, 27, 123–145. 3, 10, 15

Weil, P. (1989): "The equity premium puzzle and the risk-free rate puzzle," *J. Monet. Econ.*, 24, 401–421. 4