An Introduction to Life-Cycle Models

Robert Kirkby

Victoria University of Wellington

November 18, 2022

Abstract

Introduces life-cycle models and shows how to solve them and build more realistic versions. Covers most of the main concepts in 'one-asset' life-cycle models with consumption-leisure and consumption-savings decisions. Proceed by a series of models, with the intention that you can read the pdf description of a model, then look through the codes that show how to implement it. By the end you can easily solve one-asset finite-horizon problems, including plotting life-cycle profiles and simulating panel data. A few of the models are dedicated to describing key concepts of borrowing constraints, precautionary savings, and incomplete markets.

Keywords: Life-cycle model.

JEL Classification:

1 Introduction

We gradually build up a life-cycle model that can be used to look at income, hours worked, consumption, and assets over the life-cycle. We will start with a deterministic life-cycle model in which people live for J periods and make decisions on how much to work. Our second model will then add a decision about how much to save (assets). Our third model will just use this model to draw a life-cycle profile. We will then step-by-step make additions to the model to understand how these help us create more realistic life-cycle profiles including idiosyncratic shocks. This pdf explains the models one-by-one, and the codes implementing each of them are all on github. We make use of the VFI Toolkit (vfitoolkit.com). The intention is that you can go through the models one-by-one, first reading the pdf explanation of the model and then running the codes and seeing how to implement it.

By the end we will have a life-cycle model in which people make consumption-savings and consumption-leisure choices, which has working age and retirement, which is capable of capturing that earnings are hump-shaped over age (peaking around ages 45-55), the variance of both income and consumption increase with age, incomes grow in line with deterministic economic growth of the economy as a whole, people have some assets left when they die, people face the risk of substantial medical costs when old, and where borrowing constraints and precautionary savings play an important role. And we will be able to use these to plot life-cycles, including mean, variance, and gini coefficients conditional on age, and even on 5 year age-bins. We will also be easily able to simulate panel data sets from the model on which we could run regressions.

These life-cycle models can be used easily requiring very little knowledge of numerical methods; all you need is Matlab and a gpu.[1] They are intended as a simple way to understand and use life-cycle models. Follow-up documents will then build on these to explain how to extend them to general equilibrium OLG (overlapping-generation) models, and how to solve general equilibrium transition paths for OLG models. The codes are an 'introduction' to life-cycle models, not the final word. If you find this interesting and want to start working with models that have two or three assets you will need to learn to write (much faster) code yourself; I recommend looking up the EGM (endogenous grid method) for solving finite-horizon models and starting from there.[2] Obviously from the perspective of the codes there is no need to stick to 'hours worked', 'assets', and 'labor productivity shocks' as the decision variable, endogenous state, and exogenous state, they can be anything you want to call them/interpret them as.

Note that the calibration of the model parameters has not be taken seriously in these example models. If you do any work with life-cycle models it is very important that you take very seriously the question of what the appropriate parameter values are (whether by calibration, simulationed method of moments estimation, or anything else). I have not done so as I wish to focus on the concepts around setting up and solving life-cycle models.

I have put in enough grids points on assets to make the solutions reasonably accurate, but I recommend you use more to get full accuracy in practice, especially if looking at things like the tails of distributions, inequality, subpopulations, or correlations in panel data. I have erred on the low side while still having enough points to illustrate concepts simply so that the codes can be run on less powerful gpus and therefore be more widely accessible.

If you have any questions about the material, or spot a typo in the codes, or would just like to ask a clarifying question, etc., please use the forum: discourse.vfitoolkit.com

[1]The codes make automatic use of the gpu via the VFI Toolkit for Matlab; the gpu must be an NVIDIA gpu. I tested all of these codes on a gpu with 8gb of ram, vast majority of them would work with less gpu memory but no promises :)

[2]The VFI Toolkit just uses pure discretization (discretize all decision variables, next period endogenous state, this period endogenous state, and the exogenous states), this is simple and robust, but not nearly as good a combination of speed and accuracy and more sophisticated methods.

Contents

2 Building up Life-Cycle Models

2.1 Life-Cycle Model 1: Consumption-Leisure

Households live for J periods and solve the life-cycle problem,

$$\sum_{j=1}^{J} \beta^{j-1} \quad \left[\frac{c_j^{1-\sigma}}{1-\sigma} - \psi \frac{h_j^{1+\eta}}{1+\eta} \right]$$
$$c_j = w h_j$$
$$0 \leq h_j \leq 1$$

where j indexes age, c_j is consumption, h_j is hours worked,[3] w is the wage per hour worked. So a household that works h_j hours receives an hourly wage of w, giving them a pre-tax labor income of wh_j. Notice that all the parameters are just a single number.[4]

We can rewrite this recursively as a value function problem,

$$\begin{aligned} V(j) \quad &= \quad \max_{c,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + \beta V(j+1) \\ & \quad\quad c = wh \\ & \quad\quad 0 \leq h \leq 1, aprime \geq 0 \end{aligned}$$

notice that the value function V depends on the age j, so households face different problems and make different decisions (different optimal policies) at different ages.[5] There is one of these problems for each $j = 1, 2, ..., J$; one problem for each period of the agents life. We assume $V(J+1) = 0$.[6] Notice also from the budget constraint that once the household chooses one of c or h, it is implicitly choosing the other. The codes take advantage of this and just has one decision of h.

To solve this we need parameter values for β, σ, ϕ, η and w. In the codes we will create these. We also need to let the codes know what the 'return function' is: essentially this combines the period-utility function ($\frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta}$) with the constraints ($c = wh$ and $0 \leq h \leq 1$).[7] The only other thing the codes need is the 'grid' on h, and it needs to know which parameter is the discount factor —the parameter that discounts the next period

[3] h is actually 'fraction of time worked', but I will call it hours worked throughout simply as it is easier to think of in this way. $0 \leq h \leq 1$ is showing us that this 'fraction of time worked' is from 0 to 1.

[4] In principle we could write the budget constraint as $c_j \leq wh_j$, but it would anyway bind every period as otherwise the agent is essentially throwing income away by not consuming it all (there is nothing else to do with it in this model).

[5] While true of these models in general this is actually not true in this most simple example. In this first simple example the problems for each age are completely seperate as there is nothing connecting them, that is, there is no way (such as savings) to move resources between periods. Put another way, nothing the household does in one period can have any influence on any other period. As a result, for this simple example the household actually faces a seperate problem in each period, and because all the constraints, utility function and parameters are the same it is even the same problem at every age and so the household will make the same decisions at every age.

[6] $V(J+1)$ is effectively the utility received upon dying.

[7] In practice we will enforce $0 \leq h \leq 1$ via the grid on h, so only the budget constraint appears inside the code for the return function.

value function $V(j+1)$— which in our case is β.

We will set this up in the code and then use the command 'ValueFnIter_FHorz_Case1()' to solve to get both the value function and the policy function. The value function tells us the value at each point on the grid (currently the only variable the value function depends on is j). It also gives us the 'policy function', which is the optimal decision the agent makes, in our case this will just be the choice of h (by default the policy is the choice of grid point for h, rather than the value of h, the codes will make this clearer).

2.2 Life-Cycle Model 2: Retirement

Let's make just about the simplest change we can. Currently households live J periods, specifically they live 81 periods intended to represent ages 20 to 100 years old, inclusive. In reality we know many people retire at age 65 and receive a pension. We will add this to our codes. We need to tell the codes how to detect retirement (which we will make automatic at age 65) and how much the pension is. Let $Jr = 46$ be the retirement age (46=65-19, 19 is years before the first model period of age 20 years old). The life-cycle value function of the household is now,

$$
\begin{aligned}
V(j) \quad &= \quad \max_{c,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + \beta V(j+1) \\
&\quad\quad \text{if } j < Jr: \; c = wh \\
&\quad\quad \text{if } j >= Jr: \; c = pension \\
&\quad\quad 0 \leq h \leq 1, aprime \geq 0
\end{aligned}
$$

notice that $j < Jr$ is pre-retirement (working age), and $j >= Jr$ is retirement during which the household can no longer earn money by working, but does receive a pension. Notice that the codes will need to know age j, which in the code we will call $agej$.

In terms of the codes this will mean creating the parameters (Jr, $agej$ and $pension$) and passing them into the return function.

2.3 Life-Cycle Model 3: Assets

We will now add assets to the household problem. This will give households a way to save from one period to the next. We denote assets as a, and next period assets we call $aprime$. Households decisions depend on the (endogenous) state variable a, and one of those decisions is $aprime$. Because decisions now depend on assets, as well as age, we have $V(a,j)$. The life-cycle value function of the household is now,

$$\begin{aligned} V(a,j) \quad &= \quad \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + \beta V(aprime, j+1) \\ & \quad\quad \text{if } j < Jr: \ c + aprime = (1+r)a + wh \\ & \quad\quad \text{if } j >= Jr: \ c + aprime = (1+r)a + pension \\ & \quad\quad 0 \leq h \leq 1, aprime \geq 0 \end{aligned}$$

Notice that current assets a are added to the 'resources' in the budget constraint, as well as interest rate r which is paid on the asset holdings. $aprime$, next period assets subtract from present period assets, and so $aprime$ appears on the left-hand-side of the budget constraint: every dollar saved ($aprime$) is a dollar not spent (c).

In terms of the codes this will mean creating a grid for the assets, and adding them into the return function.

2.4 Life-Cycle Model 4: Life-Cycle Profiles

We make no change to the model. Instead we will look at 'life-cycle profiles', which are graphs of how variables change with age. So our value function is still the same,

$$\begin{aligned} V(a,j) \quad &= \quad \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + \beta V(aprime, j+1) \\ & \quad\quad \text{if } j < Jr: \ c + aprime = (1+r)a + wh \\ & \quad\quad \text{if } j >= Jr: \ c + aprime = (1+r)a + pension \\ & \quad\quad 0 \leq h \leq 1, aprime \geq 0 \end{aligned}$$

What we want to do is to start households off at age j =1 with zero assets, and watch what they do over their lifetime. Notice that what they do at age 1 is determined by the policy function at age 1, and part of what it tells use is aprime, which in age 2 will become current assets a. So wherever the household starts at age 1, the policy at age 1 tells us where the household is at 2, and then the policy at age 2 tells us where the household is at age 3, etc. We can then draw a graph of what happens to the household over the life-cycle (given how they start) and this will be what we call a 'life-cycle profile'. More precisely, the life-cycle profile of a given variable, say hours worked, shows the mean value of that variable conditional on age (so hours worked at age 1, hours worked at age 2, etc.).[8]

To be able to create life-cycle profiles we need to make an assumption about how agents are 'born' at age $j = 1$. We will assume that they start with zero assets. The codes refer to this as the $jequaloneDist$, the distribution of

[8]Later, in more sophisticated models this distinction will matter and the correct interpretation of the life-cycle profile for a variable is as the age-conditional mean value of that variable (there are also life-cycle profiles for age-conditional standard deviations, etc.). Because the present model has no uncertainty (no idiosyncratic shocks) and all agents are identical at birth (at age $j = 1$) it just so happens that in this model all households will exactly follow this life-cycle profile; there is no variance so everyone is exactly at the age-conditional mean.

agents at age 1.

Because of how the codes work, before we can calculate the life-cycle profiles we have to create something called the 'stationary distribution'.[9] We will not explain the concept of the stationary distribution here, but will cover it later on. To compute the stationary distribution we need one other piece of information, namely how many households are of each age, we will just assume there is an equal fraction of each age, so $1/J$ agents of each age; it is almost always assumed that the total population is normalized to have a mass of 1.[10]

We want to draw three life-cycle profiles: fraction of time worked (h), earnings (wh), and assets (a). To do this we first set up three 'FnsToEvaluate'. And then just run the life-cycle profile command. We then plot these. Note that we are creating 'Mean' life cycle profiles.[11]

2.5 Assignment 1: Add a tax on labor income

Now is a good chance to see if you can make some changes yourself. Try and modify the fourth life-cycle model by adding a tax on labor income, τ_l. So the value function you should implement is,

$$
\begin{aligned}
V(a,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + \beta V(aprime, j+1) \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + (1-\tau_l)wh \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& 0 \leq h \leq 1, aprime \geq 0
\end{aligned}
$$

where the only change is the inclusion of $(1-\tau_l)$ multiplying the wh, note that $(1-\tau_l wh)$ is after-tax labor income.

You will need to add the parameter τ_l, which you can give the value $\tau_l = 0.2$ (a 20% tax rate). You will need to modify the value function. You can also try to draw the life-cycle profile of tax paid ($\tau_l wh$).

A code implementing this can be found in the assignments folder: Assignment1_LifeCycleModel.m and the related return fn code.

2.6 Life-Cycle model 5: Earnings are hump-shaped

If we plotted the life-cycle of earnings using real-world data we would see a clear 'hump-shape'. Earning increase when young, peak around age 45-55, and then fall slightly until retirement. We want to capture this fact with our model. To do this we will introduce deterministic labor productivity units as a function of age. The basic idea is

[9] The stationary distribution is not needed for the specific life-cycle profiles we need in this model, but is necessary for other life-cycle profiles in more sophisticated models.

[10] How households are distributed is needed to compute the stationary distribution, but is not relevant to the life-cycle profiles themselves (the life-cycle profile command essentially ignores them). This looks a bit silly in this simple example as we have to set up information we don't need, but it will make sense in more advanced models.

[11] As mentioned life-cycle profiles are more accurately called 'value conditional on age'. We can think of conditional mean, or conditional variance, etc. In the present model because all agents of a given age are identical (because the model is deterministic and all agents start with the same assets, namely zero) only the conditional mean is relevant.

that when young people work one hour, they are less productive/efficient, and so get paid less. This 'hourly labor efficiency' will increase until around 45-55 years old and then decrease.[12]

We will introduce a parameter, κ_j ('kappa'), that is the labor productivity units. Note that this parameter depends on age, so it is a vector of length J, rather than just a single number like all of our previous parameters. The codes will recognise that the parameter is of size J-by-1 and automatically understand that this parameter depends on age and use it appropriately.[13]

The households value function problem becomes,

$$
\begin{aligned}
V(a,j) \quad &= \quad \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + \beta V(aprime, j+1) \\
& \quad \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j h \\
& \quad \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& \quad 0 \leq h \leq 1, aprime \geq 0
\end{aligned}
$$

notice that labor earnings is now $w\kappa_j h$, 'wage time hourly productivity times hours'.

To implement this we need to create the parameter κ_j, modify the return function, and also change the FnsToEvaluate for earnings.

If you compare the hours worked profile from this model with that of life-cycle model 4 you can see how this changing hourly wage ($w\kappa_j$) has impacted households decisions to work more/less at different ages.

The purpose of this particular model is really about showing how you can set parameters to depend on age. You can make any parameter depend on age by making it of size J-by-1, and the codes will automatically recognise that the parameter is of this size and treat it as a parameter that depends on age.

2.7 Life-Cycle model 6: Chance of dying

Currently everyone in the model lives to age 100 (period j=81). In reality of course some people die at age 70, and others at age 93, and we would like to add this possibility of dying to our model.[14]

From the perspective of the household we do this by introducing a 'conditional survival probability', s_j, that is the probability of surviving to age $j+1$ given you are age j (or equivalently, it is $1-d_j$, where d_j is the conditional probability of death, the probability of dying this year given you are of age j. We can get these from real world data (many countries calculate them), and we use those for the US from 2010.[15]

[12]The equivalent in the data to find realistic numbers would be to look at how hourly wages change with age.

[13]The codes do not care if it is J-by-1 or 1-by-J, both work fine.

[14]Partly because uncertainty about how long they will live means people save more assets for old age in a manner that is more realistic (relative to just dying earlier, not relative to dying later which is how it will look in our codes relative to the previous life-cycle model), partly because if we want to later turn this model into a model of the economy as a whole, rather than just of a household, then we want a more realistic age distribution (you have probable seen 'demographic pyramid's).

[15]We are calling our agents 'households', but the data is for individuals. We will ignore this distinction here and deal with it in later models where we can think about individuals as distinct from households within the model.

To put the conditional survival probability into the model we assume that households only care about next period if they are alive next period. So they only care about next period with probability s_j. We do this by including s_j as an additional discount factor (that depends on age).

The households problem thus becomes,

$$
\begin{aligned}
V(a,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + s_j \beta V(aprime, j+1) \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j h \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& 0 \leq h \leq 1, aprime \geq 0
\end{aligned}
$$

notice that s_j appears alongside β as an additional discount factor.

We need to add the parameter s_j to the codes (it will be J-by-1 as it depend on age). We also need to tell the codes that s_j should be used as an additional discount factor. Note that there is no change to the return function.

If you compare the asset profiles you will see that people have changed the amount of assets they keep for retirement because there is a chance they will die before getting to consume them.

2.8 Life-Cycle model 7: Warm-glow of bequests

Households in our model currently aim to die with zero assets; this was possible to acheive exactly until we introduced the conditional survival probabilities, you can see it in the life-cycle profiles of assets. In reality many people leave assets behind when they die, and not just people leaving bequests/inheritances to their children, even people without children leave assets, often to charities. One way to deal with this would be to model children explicitly, such 'dynastic OLG' models exists, but are more complex. We will instead model a 'warm glow of bequests': households get utility from leaving assets behind when they die. As a result households with aim to die with non-zero assets, which is realistic.

The warm glow of bequests from leaving behind assets a is, $warmglow(a) = warmglow1 \frac{(a-warmglow2)^{1-warmglow3}}{1-warmglow3}$; where $warmglow2$ is the 'bliss point', which can be thought of as the ideal/target amount of assets to leave behind when dying, and $warmglow1$ determines the importance of the warm glow of bequests (relative to, e.g., consumption or leisure). Notice how the functional form is almost exactly the same as the utility of consumption, and $warmglow3$ acts like the curvature parameter. By using the same functional form it is much easier to choose an appropriate calibration of the parameters for the warm-glow of bequests.

We want agents to only receive the warm glow of bequests when they die. The easiest way to do this would be to only given the warm glow of bequests with assets left at the end of the last period ($j = J$). In this case we would just add a term $\mathbb{I}_{j=J} warmglow(aprime)$ to the return function.[16] Alternatively we could give the warm

[16] $\mathbb{I}_{j=J}$ is the 'indicator function' for $j = J$: it takes a value of 1 when $j = J$ and a value of zero otherwise.

glow of bequests whenever agents die —note that the conditional survival probabilities mean they could die at 'any' time— in which case we could add a term $(1-s_j)warmglow(aprime)$ (to all ages), with $(1-sj)$ being the probability of dying and therefore receiving the warm glow of bequest from the assets left at the end of the period, $aprime$. We will in fact use $\beta(1-s_j)warmglow(aprime)$ as households don't die until the end of the period, but notice that this is really just changing the precise interpretation/value of $warmglow$, nothing meaningful changes when introducing β here. Because agents of all ages have a risk of dying the warm glow would impact asset choices rather directly at all ages, so we will just restrict it to when agents are 75 years or older (retirement age plus 10), which we do by using $\beta(1-s_j)\mathbb{I}_{(j>=Jr+10)}warmglow(aprime)$

So our households problem is now

$$
\begin{aligned}
V(a,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
& \qquad\qquad + s_j\beta V(aprime, j+1) \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j h \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& 0 \leq h \leq 1, aprime \geq 0
\end{aligned}
$$

notice that the $(1-s_j)\beta warmglow(aprime)$ term is going into the return function.

Implementing this requires us to create the warm glow parameters and then pass these (and β and s_j) into the return function which needs to be modified to include the warm glow of bequests. The warm glow is only given when age j is at years into retirement, this is just done for convenience as otherwise it distorts assets decisions at young ages too much because the risk of dying is non-zero at yound ages; it is close to zero, but not zero.

We will set $warmglow2 = 3$, so the target assets when dying are 3. You can see the impact this has on assets when old by looking at the life-cycle profile of assets.

2.9 Life-Cycle model 8: Idiosyncratic shocks and heterogeneity

Until now the only way households could differ is if they started with different assets at 'birth' ($j = 1$). Everything about the model is deterministic, and people in the same state always make the same decisions. We will add 'idiosyncratic shocks', which essentially means that each household will be hit with a shock that affects just them (rather than the economy as a whole; since we do not yet model the economy as a whole the distinction is not yet important, but will be later). We will start with the simplest thing we can, an unemployment shock. The unemployment shock will take two possible values, employed and unemployed. If a household currently has the employed value then they can choose to work just as before. If a household currently has the unemployed value then they are unable to earning income by working (their labor supply, h, must equal zero).[17]

[17]Note that for the retired households these unemployment shocks are irrelevant; we will still model them as that is easier to set up, but we will see in a later model how to change the shocks depending on age (in this case get rid of them when

We will assume that the value of the unemployment shocks next period depend on the value of the unemployment shock today; that the unemployment shocks are a (first-order) markov process.[18] Let z be the unemployment shock. Let $z = 1$ be employment state and $z = 0$ be the unemployment state. So we could think of the possible values of z as $[1, 0]$. We can define the 'markov transition function' as the matrix

$$\pi_z = \begin{bmatrix} p_{ee} \; p_{eu} \\ p_{ue} \; p_{uu} \end{bmatrix}$$

where p_{ee} is the probability of employment state next period given employment state this period; that is, the probability of *transitioning* from employment this period to employment next period. p_{eu} is the probability of transitioning from employment this period to unemployment next period. Notice then that the first row is the probabilities of the possible outcomes next period, and so the first row must add to one (because tomorrow happens with probability one).[19] The bottom row is the transition probabilities from unemployment this period to employment next period and unemployment next period, respectively.

Because the unemployment shocks are markov the expectations about tomorrow depend on the state today, and as a result the value function will depend on the value of the unemployment shocks.

The household value fn problem is,

$$\begin{aligned} V(a, z, j) \quad = \quad & \max_{c, aprime, h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\ & \qquad + s_j \beta E[V(aprime, zprime, j+1)|z] \\ & \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j z h \\ & \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\ & 0 \leq h \leq 1, aprime \geq 0 \\ & zprime = \pi_z(z), \text{ is a two-state markov} \end{aligned}$$

Notice that z is now a state, and also notice that next period value function is now calculated in expectation, the $E[]$ in $E[V(aprime, zprime, j+1)|z]$, because the future is uncertain (it depends on the idiosyncratic shock).

In the codes we will need to let the code know that there is an exogenous state, z, which takes two values, and give the grid and the transition probabilities for z. We also need to modify the return function. Almost everything else is unchanged, the codes recognise the exogenous shock and treat it appropriately when solving the value function, stationary distribution, and life-cycle profiles. We need to make a minor change to $jequaloneDist$ as we need to specify whether 'newborns' start in the employed or unemployed state; we will make it 0.7 employed

age is of retirement age).

[18]A (first-order) Markov process is one in which $Pr(y_{t+1}|y_t, y_{t-1}, y_{t-2}, ...) = Pr(y_{t+1}|y_t)$, that is, to predict the value of the markov process next period, y_{t+1}, the only useful information is y_t (if we didn't know the value of y_t other things would be useful, but as long as we know y_t that is all that matters).

[19]This is true of any markov transition matrix: for each row, the sum of the elements in that row much sum to one.

and 0.3 unemployed (remember the total mass of newborns should be 1).

Notice that while both a and z are states, the household can choose a, which is therefore an *endogenous* state. While z cannot be chosen by the household, so it is an *exogenous* state. For exogenous states we have to say how they change over time, which is the role of π_z.

Because adding an idiosyncratic shock z changes the 'size' of the value function and policy function the codes repeat our earlier exercises of plotting these.

2.10 Life-Cycle model 9: Idiosyncratic shocks again, AR(1)

We introduced a shock z that takes two possible values: employed or unemployed. Now let's change z to being about labor productivity. We already have κ_j as deterministic labor productivity depending on age. We will make z an AR(1) process on labor productivity units.[20] The code allows for exogenous shock variables to be any discrete markov process, so we have to convert the AR(1) process z into a markov process. There are many ways to do this, we will use Farmer-Toda; if you are interested in the alternatives like Rouwenhorst, Tauchen and Tauchen-Hussey, see Life-Cycle model A1.

Our household value function is essentially unchanged,

$$\begin{aligned}
V(a,z,j) \quad &= \quad \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
&\qquad +s_j\beta E[V(aprime,zprime,j+1)|z] \\
&\text{if } j<Jr:\ c+aprime=(1+r)a+w\kappa_j zh \\
&\text{if } j>=Jr:\ c+aprime=(1+r)a+pension \\
&0\leq h\leq 1, aprime\geq 0 \\
&log(zprime)=\rho_z log(z)+\epsilon,\ \epsilon\sim N(0,\sigma^2_{\epsilon,z})
\end{aligned}$$

with the only difference being the definition of z (and implicitly π_z). We could rewrite $log(zprime)=\rho_z log(z)+\epsilon$ as $log(zprime)=\pi_{log(z)}(log(z))$, which is how we can see that an AR(1) is just a specific kind of markov process.

We need a way to discretize $log(z)$, approximating it as a discrete markov process with a grid and transition matrix. When we come to code this we will use the Farmer-Toda method to discretize $log(z)$ and $\pi_{log(z)}(log(z))$. We can then set $z=exp(log(z))$ to get a grid on z, and notice that because we now (after Farmer-Toda method) have $log(z)$ as discrete if follows that $\pi_z=\pi_{log(z)}$; that is we need to take the exponential of the grid, but can just use the transition matrix.[21]

So in the code we just need to create the parameters ρ_z and $\sigma_{\epsilon,z}$. Then use the Farmer-Toda method to create

[20] AR(1) means 'autoregressive process of order 1'. An AR(1) process y_t follows $y_t=(1-\rho)c+\rho+\epsilon_t$, $\epsilon_t\sim N(0,\sigma^2_\epsilon)$. It is a standard model in time-series econometrics. Note that by making the constant $(1-\rho)c$ it follows that $E[y_t]=c$, which gives a nice easy way to interpret c and ρ.

[21] Why the log? Because the AR(1) process can be negative, but by taking the exponential of the AR(1) we guarantee that z is positive (which helps interpret it, what would a negative productivity even mean?).

the grid and transition matrix based on this. Then take exponential of the grid (we will then also normalize the grid to mean of 1, this makes choosing parameter values easier, although it will be irrelevant in our case).[22] We need to decide how many states to use when discretizing the AR(1) process: the more states the more accurate the discretization, but more states will make the code slower. There is no 'correct' number of states, and you should probably check your results for their sensitivity to how you discretize shocks. We will use $n\_z = 21$, which is substantially more than the standard in the literature.

The Farmer-Toda method that we use here is recommended for discretizing AR(1) processes, which the sole exception of those AR(1) processes with very high persistence, $\rho_z \geq 0.99$, for which the Rouwenhorst method is recommended. The Rouwenhorst method, along with other methods common in the literature like the Tauchen method and the Tauchen-Hussey method can be easily implemented, see Life-Cycle Model A1, but I recommend against using them as they are not as accurate.

A very brief explanation of how Farmer-Toda works: in a first step, a grid for z is set using either evenly-spaced points. The second step is to choose the transition probabilities. For a given grid point today the corresponding row of the transition matrix is a probability distribution for tomorrows state, so Farmer-Toda choose the transition probabilities of this row to target the moments of this probability distribution (by default the first two moments, but code allows up to four). If we have, e.g., five grid points, then this is two restrictions, which would not be enough (there would be a continuum of possible solutions), so Farmer-Toda add the target of maximizing the entropy (relative likelihood) which makes the solution unique. Note that this is done row-by-row for the transition matrix.[23]

The main weakness of Farmer-Toda is that it takes about 1 second to run, while the alternatives take more like 1/1000th of a second. For our purposes this is irrelevant but may matter if you want to do, e.g., simulated likelihood estimation or simulated method of moments estimation of the life-cycle model. (Don't worry if you don't understand what these are, they are much more advanced than our current focus.)

[22] We discretize the AR(1) and then take the exponential of the grid, and then normalize the grid to have a mean of one. Notice that we could alternatively just discretize the AR(1) and then use $exp(z)$ in the return function (in the budget constraint). This would run fine, but it would not have the mean of $exp(z)$ being one. This would be massively complicate things like trying to calibrate/estimate κ_j to the data.

[23] The idea of matching the moments and then using maximum entropy is actually from Tanaka and Toda (2013) who apply it to i.i.d random variables, Farmer and Toda (2017) is about generalizing to a markov process by doing this row-by-row for the transition matrix.

2.11 Life-Cycle model 10: Exogenous labor supply

We solve a version of Life-Cycle model 9 in which the labor supply is exogenous, that is there is no choice of hours worked,

$$
\begin{aligned}
V(a,z,j) \quad = \quad & \max_{c,aprime} \frac{c^{1-\sigma}}{1-\sigma} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\
& \qquad\qquad + s_j \beta E[V(aprime, zprime, j+1)|z] \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j z \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& aprime \geq 0 \\
& log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z})
\end{aligned}
$$

notice that h has disappeared from the 'decision variables', and so has the disutility of hours worked. z is now a 'stochastic endowment' or 'exogenous labor income' or 'exogenous earnings' (more precisely $w\kappa_j z$ is the endowment/labor income endowment); just different ways to describe it. This problem is often called the '*income flucations*' problem in the literature.

Implementing this first sets $n_d = 0$ and is then essentially just setting the 'decision variable grid' to be empty; $d\_grid = []$, and deleting h from all the formulaes (as well as removing the parameters that related to hours worked in either the utility function or the budget constraint). You could do the same by setting $d\_grid = 0$.

Note that keeping w is slightly odd here, but it makes it easier to compare results to the endogenous labor supply models. In any case $w = 1$ so that is irrelevant.

2.12 Life-Cycle model 11: Idiosyncratic shocks again, persistent and transitory

We introduced an idiosyncratic shock to labor productivity units. We modelled that shock as an AR(1) process. But empirical work suggests we can do better by modelling changes in labor producitivity units as a combination of two shocks, one persistent and one transitory. The persistent shocks we will model as an AR(1). The transitory shock we will model as i.i.d. There are two ways you could do this using VFI Toolkit. The first is simply to treat them as two markov shocks, and this is done in Life-Cycle model 11A. Alternatively, alongside the standard exogenous state 'z' variables that are understood to be first-order Markov processes, VFI Toolkit also has exogenous state 'e' variables that it understands to be i.i.d., and this is done in Life-Cycle model 11B. Setting up as an 'e' variable takes an extra line or two of code but has the advantage that the code will then be faster; it is the recommended way to treat variables that are i.i.d.

2.12.1 Life-Cycle model 11A: transitory i.i.d. as second 'z' variable

The persistent shocks, z_1, we will model as an AR(1). The transitory shock, z_2 we will model as i.i.d.

Our household value function now has $z_1 z_2$ in place of z to determine idiosyncratic producitivity units,

$$\begin{aligned} V(a, z_1, z_2, j) \quad = \quad & \max_{c, aprime, h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\ & \quad + s_j \beta E[V(aprime, z_1 prime, z_2 prime, j+1)|z_1] \\ & \text{if } j < Jr: \ c + aprime = (1+r)a + w\kappa_j z_1 z_2 h \\ & \text{if } j >= Jr: \ c + aprime = (1+r)a + pension \\ & 0 \leq h \leq 1, aprime \geq 0 \\ & log(z_1 prime) = \rho_{z1} log(z_1) + \epsilon, \ \epsilon \sim N(0, \sigma^2_{\epsilon, z1}) \\ & log(z_2) \sim N(0, \sigma^2_{z2}) \end{aligned}$$

Notice that z_1 is AR(1) and z_2 is i.i.d. normal (in logs).

Note that to get an exogenous shock (log) z_2 which is i.i.d. $N(0, \sigma^2_{z2})$, we can simply use the same method as we did for the AR(1) process and just set $\rho = 0$.[24]

How do we let the codes know we have two exogenous shocks? First we have to set n_z to be a vector with the number of grid points of the first and second exogenous shocks, so if we use 21 points for the AR(1) shock and 5 for the iid shock we will set $n\_z = [21, 5]$. Second we set $z\_grid = [z1\_grid; z2\_grid]$; (we 'stack' the column grids of the two shocks). Third, we set the joint transition matrix $pi\_z = kron(pi\_z2, pi\_z1)$;, by using the kronecker product, notice that when doing this you must put them in reverse order. We also need to modify the inputs to the ReturnFn and FnsToEvaluate so that they include both $z1$ and $z2$.

2.12.2 Life-Cycle model 11B: transitory i.i.d. as an 'e' variable

The persistent shocks, z, we will model as an AR(1). The transitory shock, e we will model as i.i.d. Because VFI Toolkit will know that z is markov and e is i.i.d. it can take advantage of the simplicity of the i.i.d. shock to run faster. We make no change at all to the household problem from Life-Cycle model 11A, which is just rewritten here with z and e instead of z_1 and z_2. Note that the codes use the identical ReturnFn as what matters in the code is just the ordering of inputs, the names are solely internal to the script (this is a basic programming concept you are likely familiar with).

[24] If we wanted any other i.i.d. distribution we just create the transition matrix for z_2, π_{z2}, so that it has all rows with that same distribution. Recall that a row of the transition matrix represents the probabilities of going from the state that the row represents to each of the other states, so if all the rows are the same it is saying that the current state does not matter, which means it will be i.i.d. So, e.g., we could make z_2 a uniformly distributed variable by making the transition matrix a matrix of ones, divided by the number of states of z_2 (really we just want to divide each row, but dividing the whole matrix is an easy way of implementing this).

Our household value function now has ze in place of z to determine idiosyncratic producitivity units,

$$\begin{aligned}
V(a,z,e,j) \quad &= \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
&\quad +s_j\beta E[V(aprime,zprime,eprime,j+1)|z] \\
&\text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j zeh \\
&\text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
&0 \leq h \leq 1, aprime \geq 0 \\
&log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0,\sigma^2_{\epsilon,z}) \\
&log(e) \sim N(0,\sigma^2_e)
\end{aligned}$$

Notice that z is AR(1) and e is i.i.d. normal (in logs).

Note that to get an exogenous shock (log) z_2 which is i.i.d. $N(0,\sigma^2_{z2})$, we can simply use the same method as we did for the AR(1) process and just set $\rho = 0$, because it is explicitly i.i.d. we then keep only the first row (or any row) of the transition matrix created to store as a column vector in *pi_e*.

How do we let the codes know we have two exogenous shocks? First we just set up z exactly as we would if it were the only such shock (because it is the only markov shock). Then we set up e in a similar fashion, except that it has to go in *vfoptions* and *simoptions* (because it is not part of the baseline setup). So we set *vfoptions.n_e*, *vfoptions.e_grid* and *vfoptions.pi_e* to be the number of points, the grid, and the probability weights, respectively. We need to put the exact same things into *simoptions.n_e*, *simoptions.e_grid* and *simoptions.pi_e*. The important difference when setting up an i.i.d. 'e' variable compared to a markov 'z' variable is that instead of having a transition matrix 'pi_z' we instead have probability weights 'pi_e'; instead of a matrix we have a column vector.

2.13 Alternative Exogenous Shock Processes (Appendix A)

There are many other things we might want to do with exogenous shocks, both in terms of different kinds of shock process —VAR(1), AR(1) with non-gaussian shocks, AR(1) with parameters that depend on age—- as well as different methods for discretizing shocks. A wide variety of these are demonstrated in Life-Cycle Models Appendix A (models A1 to A9).

In Life-Cycle Model 11 we had two shocks which are independent. It is possible to have them interact. Two simple cases to illustrate. First, it is possible to use a more sophisticated version of the Tauchen method to discretize a VAR (vector autoregression) see Life-Cycle Model A7. Second, we could have one shock being recession/boom, and the other employment/unemployment, and we want unemployment to be more likely during recessions, so the second shock transitions depend on the state of the first shock. This is just a matter of setting up the joint transition matrix appropriately, see Life-Cycle Model A8.

There are two main types of idiosyncratic shocks used in the literature that we have not yet addressed, permanent (unit-root/random walk) and fixed effects. Permanent (bounded) unit-root shocks are easy to create as a markov process but require a lot of grid points in the state and so for that reason are difficult to model, Life-Cycle Model A3 shows how.

If we have multiple shocks that are correlated we might what to use a 'jointly-determined-grid' instead of a kronecker (cross-product) grid. Life-Cycle Model A9 shows how to do so.

We will discuss later, starting from Life-Cycle Model 24, the best way to use the code to model a fixed effect. You could do it by simply setting it up as another exogenous shock which has the identity matrix as the transition matrix, but the code will be slower and use more memory than the alternative we discuss later using 'permanent types'.[25]

We discuss later in Life-Cycle Models 20 and 21 how to allow the idiosyncratic shocks to be age-dependent. The specific case of an AR(1) process with age-dependent parameters is covered Life-Cycle Model A5.

How do you use empirical data to get the parameter values to use for your model with these kinds of shocks? The econometrics of how to estimate the combination of fixed effects, persistent AR(1) shocks, and iid shocks, alongside a deterministic function of age, for the labor producitivity shocks from empirical data, see Kaplan (2012). For the AR(1) with age-dependent parameters (plus a transitiory iid shock with age-dependent variance), you can find how to esimate it in Karahan and Ozkan (2013).

Other than the Farmer-Toda methods we have seen there are other ways to discretize AR(1) models, and Life-Cycle Model A1 looks at these. There are also ways to discretize shocks specifically for life-cycle models where the shock process changes with age Fella, Gallipoli, and Pan (2019), seen in Life-Cycle model A5. The full list of the 'appendix models' is:

Life-Cycle Model A1: AR(1) persistent shocks, alternative quadrature methods.

Life-Cycle Model A2: AR(1) persistent shocks with gaussian-mixture innovations.

Life-Cycle Model A3: Permanent shocks with endogenous labor.

Life-Cycle Model A4: Second-order Markov Processes (implementing an AR(2) persistent shock).

Life-Cycle Model A5: Age-dependent shocks.

Life-Cycle Model A6: Age-dependent shocks: persistent and transitory.

Life-Cycle Model A7: VAR(1) persistent shocks.

Life-Cycle Model A8: Shocks that depend on each other: 'recession' and 'unemployment'

Life-Cycle Model A9: Correlated shocks with jointly-determined grids.

A1-A6 are variations on Life-Cycle Model 9. They largely just change z (the permanent shocks with exogenous labor obviously requires more substantial changes). A7-A9 use two exogenous shocks that interact, and so are variations on Life-Cycle Model 11 (specifically 11A).

[25] A brief explanation of the two approaches.

2.14 Assignment 2: Alternative Utility Functions

Because the utility function is coded inside the return function changing it is easy. We will now present a few different utility functions. It is left as an assignment for you to try and implement the 'non-seperable CES utility function' by modifying LifeCylceModel9.m code, and look at how they change the life-cycle profiles. A 'solution' is provided as Assignment2_LifeCycleModel.m.

We can write a generic utility function of consumption and leisure as $u(c,l)$, where leisure can be defined as $l = 1 - h$, is the fraction of time not worked. The advantage of using leisure rather than hours worked in the utility function is that utility is increasing in leisure, which makes theoretical properties easier to derive; we will no longer need to subtract the disutiliy of hours worked as we did until now.

Using this generic utility function we can write the value function problem as,

$$\begin{aligned} V(a,z,j) \quad &= \quad \max_{c,aprime,l} u(c,l) + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\ & \qquad\qquad + s_j \beta E[V(aprime, zprime, j+1)|z] \\ & \qquad \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j z(1-l) \\ & \qquad \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\ & \qquad 0 \leq l \leq 1, aprime \geq 0 \\ & \qquad log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z}) \end{aligned}$$

notice that $h = 1 - l$ has been substituted into the budget constraint, and the restriction that $0 \leq h \leq 1$ has been replaced with $0 \leq l \leq 1$; the obvious modifications around replacing $h = 1 - l$ in the FnsToEvaluate also need to be made.

In the Life-Cycle models we have seen until now we used the 'seperable CES' utility function,

$$u(c,l) = \frac{c^{1-\sigma_c}}{1-\sigma_c} + \psi \frac{l^{1-\sigma_l}}{1-\sigma_l}$$

note of course that this is not the exact form of the utility function we used until now, but it has the same properitie, just that we are now writing it in terms of l instead of h (you could substitute $1-h$ in place of l here and the properties of the utility function would be the same).[26]

The CES refers to 'constant elasticity of substitution', and the 'seperable' refers to how the (marginal) utility of consumption is independent of leisure (and vice-versa the marginal utility of leisure is independent of consumption).

[26] The exact numbers of utils would differ, but as is well know utility is only defined up to a linear transformation; that is, the utility functions $u()$ and $a + bu()$, where a,b are constants, both represent the same preferences. This can be proven in very general terms, see any advanced Microeconomic Theory textbook. If you are unconvinced, write out a basic problem, say one-period lagrangian problem choosing between consumption of two goods, and solve it with both these utility fns, you will see how a disappears when you take derivatives, and then b cancels out because it is in all the marginal utility terms.

One important alternative is 'non-seperable CES' utility, which is given by,

$$u(c,l) = \frac{(\sigma_1 c^{\sigma_2} + (1-\sigma_1) l^{1-\sigma_2})^{1-\sigma_3}}{1-\sigma_3}$$

which also has constant-elasticity-of-substitution, but now the marginal utility of consumption depends on leisure, and vice-versa (take the derivatives and you will see they depend on both). σ_1 can be understood as a 'share' parameter, in this case it is the share of consumption (versus leisure) in utility. $1/(1-\sigma_2)$ is the elasticity of substitution between c and l. σ_3 is determining the decreasing marginal utility, and will therefore determine things like risk-aversion. Note that you can obviously set $h = 1 - l$ and solve the model in terms of hours worked, the 'assignment solution' does this: modify life-cycle model 9 to have non-seperable CES utility function.

An important subcase of the 'non-seperable CES' utility function, is given in the limit when $\sigma_2 \to 0$, and we get,

$$u(c,l) = \frac{(c_1^{\sigma} l^{1-\sigma_1})^{1-\sigma_3}}{1-\sigma_3}$$

This is particularly important in models that contain economic growth; see Life-Cycle model 22 for an explanation of why it matters, and of how to include deterministic growth in a model.[27]

An alternative non-seperable utility function is GHH preferences.[28] They are easier to write in terms of hours worked,

$$u(c,l) = \frac{\left(c - \psi \frac{h^{1+\eta}}{1+\eta}\right)^{1-\sigma}}{1-\sigma}$$

what matters to labor supply is just the wage. As the marginal rate of substitution is independent of consumption and only depends on the real wage, there is no wealth effect on the labour supply. They are not consistent with a balanced growth path (that would result from deterministic growth). A version that extends this to allow a balanced growth path are Jaimovich-Rebelo preferences.

2.15 Life-Cycle model 12: Epstein-Zin utility function

Epstein-Zin utility avoids an 'restriction' implicit in the use of CES utility functions, namely CES utility has a single parameter that determines both risk aversion and the intertemporal elasticity of substitution. Epstein-Zin preferences have seperate parameters determining risk aversion and the intertemporal elasticity of substitution.

Implementing Epstein-Zin preferences is slightly more complicated that just changing the utility function, it also involves a reformulation of the value function problem itself. Starting from Life-Cycle model 9, we make just

[27] You could call this the Cobb-Douglas utility function. It is the same functional form as the Cobb-Douglas production function, just that the parameters need some rewriting to make it obvious.

[28] Greenwood-Hercowitz-Huffman preferences.

the change to Epstein-Zin preferences and get the value function problem,

$$V(a,z,j) = \max_{c,aprime,h} \left[(1-s_j\beta)u^{1-1/\psi} + s_j\beta(E[V(aprime,zprime,j+1)^{1-\gamma}|z])^{\frac{1-1/\psi}{1-\gamma}}\right]^{\frac{1}{1-1/\psi}} \tag{1}$$

$$\text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j zh \tag{2}$$

$$\text{if } j >= Jr: \; c + aprime = (1+r)a + pension \tag{3}$$

$$0 \leq h \leq 1, aprime \geq 0 \tag{4}$$

$$log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z}) \tag{5}$$

where $u = c^{1-\chi}(1-h)^{\chi}$. If you wanted to do a version with exogenous labor the you use $u = c$.

Implementing this in the codes is as simple as creating the additional parameters, and setting,
$vfoptions.exoticpreferences =' EpsteinZin'$
You also need to add the two Epstein-Zin parameters, γ and ψ as the last two entries in the DiscountFactorParamNames. Note that implementing this also means the return function should return u (not $u^{1-1/\psi}$).

For an explanation of Epstein-Zin utility and how it is implemented in the codes see this link. The link has a pdf explaning Epstein-Zin preferences, both conceptually and how to implement them, and links to codes implementing them in both finite and infinite horizon value function problems.

2.16 Life-Cycle model 13: Simulating Panel Data

We used the FnsToEvaluate to create the life-cycle profiles. We can also use the same FnsToEvaluate to simulate panel data (now that we have idiosyncratic shocks). Note that a single household can be simulated to create a time series, and by starting numerous household simulations we get panel data.

We will use the model of Life-Cycle model 9, so we do not repeat that here. And we will simulate the same FnsToEvaluate as we used in that model.

We can choose the number of time periods for the panel data simulations using *simoptions.simperiods* (it cannot be larger than number of model periods in a life-cycle model), and can choose the number of individuals time series simulations using *simoptions.numbersims* (1000 by default). Any simulation needs to start from somewhere. The codes use 'InitialDist' as a distribution from which the starting state of agents is drawn (*seedpoint* is what VFI Toolkit calls it internally), we will just set InitialDist to be equal to the stationary distribution in the current example, but in principle any distribution could be used.

2.17 Life-Cycle model 14: More Life-Cycle Profiles

Now that our models have idiosyncratic shocks so we can look at life-cycle profiles other than the mean. For example we will look at the variance conditional on age. We will also see how to group ages together in 'age bins', for example we can look at mean income for 5-year age-groupings, or at the labor earnings Gini coefficient for all 'working age' individuals. This will all take just a few lines of code.

The model is exactly that of Life-Cycle model 9, so we do not repeat that here.

The variance, gini, and much more conditional on age were already being automatically computed, just that we did not look at them.

To use 'age bins' we just need to specify them. By default it is assumed that every period is it's own bin, this would correspond to setting $options.agegroupings = 1 : 1 : N_j$. To compute them in 5 year age bins use $options.agegroupings = 1 : 5 : N_j$ (note, if N_j is not divisible by 5 the last bin will contain less than 5). For working age as a single bin use $options.agegroupings = [1, Params.Jr]$. In general, $options.agegroupings$ is a vector each element of which defines the period at which a new bin starts. So to give a more complex example $options.agegroupings = [1, 7, 10, 11, 15]$ would create four bins, the first would be periods 1 to 6, the second is periods 7 to 9, third is 10, fourth is 11 to 14, fourth is 15 on (until N_j, inclusive). We then just add $options$ as the final input to the Life Cycle Profiles command.

3 Illustrating some important concepts

3.1 Life-Cycle model 15: Consumption and Borrowing Constraints 1

We will use our models to look at how borrowing constraints impact consumption when young. Households would like to smooth consumption over their lifetime (based on life-cycle permanent income hypothesis). But earnings have a hump-shape. So households want to save some income from the 'hump' to consume in retirement, which is easy enough. They also want to bring some income from the 'hump' forward to when they are young, which requires borrowing. But they are unable to borrow due to borrowing constraints when young. So consumption is lower when young, and more importantly the marginal utility of consumption when young is higher, and marginal propensities to consume are also higher when young.

We will use a simple life-cycle model with exogenous labor, no idiosyncratic shocks, and we will make the life-cycle profile of labor producitivity units very step to really make the borrowing constraint bind strongly when young.

$$
\begin{aligned}
V(a,j) &= \max_{c,aprime} \frac{c^{1-\sigma}}{1-\sigma} + \beta V(aprime, j+1) \\
&\text{if } j < Jr: \ c + aprime = (1+r)a + w\kappa_j \\
&\text{if } j >= Jr: \ c + aprime = (1+r)a + pension \\
&0 \leq h \leq 1 \\
&aprime \geq borrowingconstraint
\end{aligned}
$$

notice that labor earnings is now $w\kappa_j$, which is exogenous so you could equally call it 'endowment' or exogenous income.

The code sets a parameter called 'borrowconstraint' that you can change to see how the borrowing constraint matters. There are life-cycle profiles of consumption and marginal utility consumption and you can see how the borrowing constraint causes a 'jump' in consumption because people cannot borrow and consume as much as they would like. You can also see that the borrowing constrained have a very high marginal utility of consumption. As a result there are large welfare gains for the household from anything that loosens the borrowing constraint (whether it is being able to borrow more, lump-sum transfers, etc.). The borrowing constrained household would also consume any extra income and so has a high marginal propensity to consume (how high depends on how long borrowing constraint is expected to continue binding).

Note that all of our models until now have had a borrowing constraint, namely $aprime \geq 0$. This was enforced via the grid on assets which had a minimum value of zero, meaning that choosing less than zero assets was simply not possible. To implement the present model we have to set a lower minimum point on the grid on assets, and we could either use that implicitly, or do as the codes have here and add a parameter called 'borrowconstraint' that imposes a tighter borrowing contraint via the return function (note that the parameter must be greater than or equal to the minimum point on the asset grid else it is not relevant). It is worth mentioning that it is 'not possible' to have a model without a borrowing constraint, in the sense that in the absence of a borrowing constraint optimial behaviour would be to borrow an infinite amount, consume it all, and die in infinite debt; mathematically this is of course possible, I say not possible in the sense you would never actually want to use that model.

3.2 Life-Cycle model 16: Consumption and Borrowing Constraints 2

We will look at how stochastic shocks mean borrowing contraints can sometimes bind when shocks are 'bad', and what this means for consumption. We use Life-Cycle Model 9, which had an AR(1) idiosyncratic shock process, except we will use just 5 states to discretize it so as to make it easier to see 'bad' shocks.[29]

In stochastic models borrowing constraints often bind after a series of 'bad' shocks. They don't bind most of the time, nor for most households, so looking at life-cycle profiles is not a good way to see them. Two options are possible: (i) we could simulate households that get a series of bad shocks and see them run up against the borrowing constraints, or (ii) we could run regressions. We will take the first approach here.

For the panel data, we first simulate a panel data set and then look for candidate households for whom the borrowing constraints are likely to bind, that is, households who have a run of bad shocks. To make it clearer we first look for households that have median or better shocks in the first 15 periods, so that they mostly get away from the borrowing constraints, and then narrow our search to those who have 7 or more bad shocks in periods 16-25. This defines our candidate households. We plot the first 30 periods for ten of these candidate households (the ten are arbitrarily selected from among all candidate households) and for these we plot the consumption,

[29] Just 5 points will provide a poor approximation to the AR(1) process, we use just five points purely to make it easier to illustrate.

the marginal utility of consumption, and next period assets. You can see in Figure 3 how the households during periods 16-25 run into the borrowing constraint as a result of the bad shocks, and as a result their consumption is low and their marginal utility of consumption is high.

Note that the reason households do not like running into borrowing constraints —because it forces them to cut consumption more than they would otherwise like— is exactly the same in this model as in the previous Life-Cycle Model 15, what differs is just the reason that the borrowing constraints bind.

To emphasise that the borrowing constraints bind for a different reason to what happened in Life-Cycle Model 15 we flatten the earnings profile to largely eliminate what drove them in that model.

3.3 Life-Cycle model 17: Precautionary Savings (with exogenous earnings)

We have already seen how the binding of borrowing constraints can force households to reduce consumption, and in particular how this leads to a large increase in the marginal utility of consumption. Households want to avoid this situation of binding borrowing constraints, and so they engage in 'precautionary savings': savings that are 'higher' than they would otherwise be so as to avoid the probability of the borrowing constraints binding.[30]

We use the exact same model from Life-Cycle Model 10, and then solve it a second time, which we call the 'no shock' version, in which $z = 1$ (rather than an AR(1) process in logs). We will use just 3 states to discretize it so as to make it easier to see how the savings policies differ.[31]

How to see precautionary savings? We take three approaches: (i) savings policy functions, (ii) life-cycle profiles, (iii) aggregate assets.

We start with the savings policies. Figure 1 shows how households in the model with the median shock value and low asset holdings[32] choose to save more, for precautionary reasons, than households in the model without shocks. This is true even though the median shock is actually slightly lower earnings than the model without shocks (0.9976 vs 1). The top panel of Figure 2 shows the period before retirement in which there are no future shocks and so no precautionary savings motive,[33] you can see that now the households with the median shock actually save less than those with no shocks (because of the 0.9976 vs 1 earnings mentioned before). The other panels of Figure 2 show that in retirement, because shocks are now irrelvant the households all make the exact same decisions (multiple lines are plotted, but you can only see one as they are all on top of each other).

[30]Note that precautionary savings is a phenomenon of stochastic models: precautionary savings are savings to avoid the chance of a borrowing constraint binding that the household. In a deterministic model households know for a fact that borrowing constraints will or will not bind. In a stochastic model a household can do precautionary savings, get good shock outcomes that means the household never end up in a situation where the borrowing constraint would bind, and now have ended up saving more than they would would like ex-post. In a deterministic model it is not possible to choose savings optimally ex-ante that you end up regretting as overly cautious ex-post.

[31]Just 3 points will provide a poor approximation to the AR(1) process, we use just three points purely to make it easier to illustrate. We do make one other change from Life-Cycle Model 10, namely we make the initial age j=1 distribution of agents have zero assets (as before) and the stationary distribution of shocks (previously just the median), this is so the life-cycle profile of mean earnings is exactly equal in the models with and without shocks; the difference is very small.

[32]Note that the graph zooms in on low asset holdings, as seen by the x-axis.

[33]Shocks are only relevant during working age as they are to labor producitivity units, and retirees never work.

We look at life-cycle profiles in Figure 3. Those on the left are with shocks, on the right are without shocks; except the bottom row that shows those with shocks minus those without shocks. Looking at the bottom-left panel we confirm that there is no difference in mean earnings for households in the models with and without shocks (is just the left panel of top row minus the right panel of top row); this is just to confirm that any savings differerences are not coming from differences in mean earnings. In the bottom-right panel we see the precautionary savings: households with shocks have higher assets than households without shocks. We can also see how the precautionary savings interact with the life-cycle consumption-smoothing motives we discussed in Life-Cycle Model 15. When households are young the life-cycle consumption-smoothing motives dominate and no households save any assets (see panels in second row); the precautionary savings motives are there, but they are overwhelmed. Later in working age the life-cycle consumption-smoothing motives largely disappear (in sense of keeping assets at zero) and we can see the precautionary savings in the bottom-right panel with households that face earnings risk holding more assets tha households that do not face any shocks.

Our final look at precautionary savings comes from looking at aggregate assets —the total assets held by all households added up across the stationary distribution— where we can see that assets are higher for the households with shocks than for the households without shocks. We don't use these 'aggregates' much in life-cycle models as they make more sense when thinking about the economy as a whole, but this idea that households will with precautionary savings save up more assets is very important in general equilibrium models of incomplete markets, an idea we will discuss later. The aggregate assets are printed to the screen when you run the codes.

3.4 Life-Cycle model 18: Precautionary Savings with Endogenous labor

Precautionary savings are one way to avoid the binding of borrowing constraints. An alternative is to adjust labor supply; working more to avoid hitting borrowing constraints. So models with endogenous labor supply have lower amounts of precautionary savings. We will use Life-Cycle Model 9, but discretizing AR(1) process on z using just three shocks values to make things clearer.

Note that while endogenous labor will reduce precautionary savings this is not the only effect it will have on savings, because it will also interact with life-cycle motives for savings; for example, households can choose to work more when they reach higher earnings levels during middle age and do all their saving for retirement then, which would do some combination of reducing assets when young (although most households don't save when young and are already up against their borrowing constraints) and increasing the asset levels in middle-age that they carry over into retirement. As a result whether endogenizing labor supply results in higher or lower levels of assets overall depends on various factors; especially, the utility of leisure and the form of the utility function.

Too see all of this we will solve the model four times. First we solve the model with endogenous labor and exogenous shocks, second we solve the model with endogenous labor and no shocks, third we solve the model with exogenous labor and exogenous shocks, and fourth we solve the model with exogenous labor and no shocks.

Comparing the model with endogenous labor and shocks to the model with endogenous labor without shocks we can see that there are precautionary savings as a result of shocks, seen, e.g., in the higher aggregate capital/income ratio in the model with shocks (see the numbers printed to screen). We can also see how agents near the borrowing constraint (low assets) chose to work more hours as a way to avoid the budget constraint. In period $j = 1$ households are trying to escape the borrowing constraint mostly for life-cycle reasons, so there is little difference between the models with and without shocks in the top panel of Figure 4. By constrast by age $j = 20$ the borrowing constraints are mostly about bad shocks, and so we can see how the model with shocks alway results in more labor supply at low assets levels compared to the model without shocks (life-cycle motives also play a role). You can see how this use of precautionary labor supply is often enough to actually reduce savings relative to the model without shocks; top panels of Figures 1, 2 and 3. This is partly because precautionary labor reduces demand for precautionary savings, and partly because of how it changes life-cycle motives.

If we now compare the model with endogenous labor and exogenous shocks to the model with exogenous labor and exogenous shocks we can see that in our current calibration endogenizing labor leads to an increase in savings; e.g., by comparing capital/income ratios. Note that this varies over the life-cycle, you can see how endogenous labor supply decreases savings at ages $j = 1$ (bottom panel of Figure 1) and $j = 20$ (bottom panel of Figure 2), but for life-cycle motives actually increases savings at age $j = 40$ (bottom panel of Figure 4). This is because the reduced precautionary savings are more than offset by the increased saving for life-cycle motives at young ages, but then the life-cycle motives go towards higher savings, adding to the precautionary savings motive, at middle age. This provides a good example of how there are often various effects of any model changes working in different directions and it is rarely clear beforehand which will dominate.

To ease the comparison of results between the model with endogenous labor supply and that with exogenous labor supply we have normalized earnings in the later so that both have the same aggregate (mean) earnings. The reason is as follows: note that for exogenous labor earnings is $w\kappa_j z$, while with endogenous labor earnings is $w\kappa_j zh$, and $0 \leq h \leq 1$, so earnings will be lower in the model with exogenous savings.[34] We can do this be simply adding a parameter to earnings in the exogenous labor model, so that earnings are now $\phi_{normalizemeanearnings} w\kappa_j z$. By setting $\phi_{normalizemeanearnings}$ equal to the mean earnings in the model with endogenous labor divided by the mean earnings in the model with exogenous labor we will give both the models the same mean earnings.[35]

This exercise of endogenous labor supply reducing precautionary savings is much cleaner in an infinite-horizon (and general equilibrium) model (Pijoan-Mas, 2006), where there are no life-cycle considerations.

[34] We could try and deal with this by looking at aggregate capital/income ratios, but this is not a full fix as there would still be income effects that may be playing a substantial role. By normalizing mean earnings we substantially reduce these other channels.

[35] Because we are doing this to the model with exogenous labor supply we don't need to worry about how people react in determining mean earnings. To find this $\phi_{normalizemeanearnings}$ we would first just solve both models with $\phi_{normalizemeanearnings} = 1$ and then calculate the appropriate $\phi_{normalizemeanearnings}$.

3.5 Life-Cycle model 19: Incomplete Markets

We saw how borrowing constraints and idiosyncratic shocks together lead to precautionary savings. There is an important third aspect to this that was 'hidden' in the background: incomplete markets. Loosely speaking, complete markets is the presence of perfect insurance which all households buy and therefore the idiosyncratic shocks they later receive are irrelavant as they are perfectly insured against them. With complete markets, because households perfectly insure themselves, there is no meaningful inequality; no meaningful distribution of households.

If we wanted to create a life-cycle model in which there was an exogenous shock that takes two values we would need two assets (one of which pays out in the case the shock takes the first value, the second asset pays out in the case the shock takes the second value), and so we won't attempt to do that here.

Instead we will solve a two period model and show how briefly the idea of how complete and incomplete markets relate. We start with a deterministic two-period model, then add shocks in the second period. We end with a brief discussion of how Representative Agents are related to complete markets models. Note that all the models being solved elsewhere in this document (and more generally, in essentially all of the heterogenous agent model literature) are incomplete markets models.

Let's start with a simple two-period deterministic model. The household solve the following maximization problem,

$$\max_{\{c_1,c_2,a\}} u(c_1) + \beta u(c_2)$$
$$\text{s.t. } c_1 + a = y_1$$
$$c_2 = (1+r)a + y_2$$

We can solve this using the Lagrangian method. First write the Lagrangian,

$$\mathcal{L}(c_1, c_2, a, \lambda_1, \lambda_2) = u(c_1) + \beta u(c_2) + \lambda_1(y_1 - c_1 - a) + \lambda_2((1+r)a + y_2 - c_2)$$

The first-order conditions of this are

$$\frac{\partial \mathcal{L}}{\partial c_1} = 0; \quad u'(c_1) + \lambda_1(-1) = 0$$
$$\frac{\partial \mathcal{L}}{\partial c_2} = 0; \quad \beta u'(c_2) + \lambda_2(-1) = 0$$
$$\frac{\partial \mathcal{L}}{\partial a} = 0; \quad \lambda_1(-1) + \lambda_2(1+r) = 0$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_1} = 0; \quad c_1 + a = y_1$$
$$\frac{\partial \mathcal{L}}{\partial \lambda_2} = 0; c_2 = (1+r)a + y_2$$

From the first three of these we get

$$u'(c_1) = \beta(1+r)u'(c_2)$$

Purely to make it easier to see how this compares to what we will do next let $\beta(1+r) = 1$ to get,[36]

$$u'(c_1) = u'(c_2)$$

Because there is one asset it is possible to shift consumption between the two periods and equate the marginal utitlities. Complete markets are key to this as we will see in the next two examples.

Now, we will introduce a shock, z, in the second period that takes on of two possible values z_1 or z_2. Let p be the probability of $z = z_1$, so then $1 - p$ is the probability of $z = z_2$. What we are about to solve is an *in*complete market model; we will explain later why this is incomplete markets. The household problem is,

$$\begin{aligned} \max_{\{c_1, c_2(z_1), c_2(z_2), a\}} u(c_1) + \beta[pu(c_2(z_1)) + (1-p)u(c_2(z_2))] \\ \text{s.t. } c_1 + a = y_1 \\ c_2(z_1) = (1+r)a + y_2(z_1) \\ c_2(z_2) = (1+r)a + y_2(z_2) \end{aligned}$$

where $c_2(z_1)$ is consumption in the second period when $z = z_1$, $c_2(z_2)$ is consumption in the second period when $z = z_2$. Notice that the difference caused by the shock is that the income/endowment is $y_2(z_1)$ vs $y_2(z_2)$; so we can think of the shock as being low/high earnings. We will assume, without loss of generality, that $y_2(z_1) < y_2(z_2)$. Notice how the asset a allows us to shift consumption between period one and period two, but there is no way to shift consumption between the two states/shocks in period two; this is the incomplete markets. Writing the Lagrangian we get,

$$\begin{aligned} \mathcal{L}(c_1, c_2(z_1), c_2(z_2), a, \lambda_1, \lambda_2, \lambda_3) = u(c_1) + \beta[pu(c_2(z_1)) + (1-p)u(c_2(z_2))] + \lambda_1(y_1 - c_1 - a) \\ + \lambda_2((1+r)a + y_2(z_1) - c_2(z_1)) + \lambda_3((1+r)a + y_2(z_2) - c_2(z_2)) \end{aligned}$$

[36] In a Representative Agent model it is typically true that $\beta(1+r) = 1$ in general equilibrium.

The first-order conditions of this are

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial c_1} = 0; &\quad u'(c_1) + \lambda_1(-1) = 0 \\
\frac{\partial \mathcal{L}}{\partial c_2(z_1)} = 0; &\quad p\beta u'(c_2(z_1)) + \lambda_2(-1) = 0 \\
\frac{\partial \mathcal{L}}{\partial c_2(z_2)} = 0; &\quad (1-p)\beta u'(c_2(z_2)) + \lambda_3(-1) = 0 \\
\frac{\partial \mathcal{L}}{\partial a} = 0; &\quad \lambda_1(-1) + \lambda_2(1+r) + \lambda_3(1+r) = 0 \\
\frac{\partial \mathcal{L}}{\partial \lambda_1} = 0; &\quad c_1 + a = y_1 \\
\frac{\partial \mathcal{L}}{\partial \lambda_2} = 0; &\, c_2(z_1) = (1+r)a + y_2(z_1) \\
\frac{\partial \mathcal{L}}{\partial \lambda_3} = 0; &\, c_2(z_2) = (1+r)a + y_2(z_2)
\end{aligned}$$

From the first, second, third and fourth, we get,

$$u'(c_1) = (1+r)\beta[pu'(c_2(z_1)) + (1-p)u'(c_2(z_2))]$$

Again, purely to simplify to something easier to look at, let $\beta(1+r) = 1$, to get,

$$u'(c_1) = pu'(c_2(z_1)) + (1-p)u'(c_2(z_2))$$

so with incomplete markets we have that the household is able to shift between period one and period two consumption using the asset, a, and so it sets the marginal utility of consumption in period one equal to the expected marginal utility of consumption in period two (expected=sum weighted by probabilities). Note that we had $y_2(z_1) < y_2(z_2)$ and so it follows from the period 2 budget constraints that $c_2(z_1) < c_2(z_2)$ (as a must be the same for both), from which if follows that $u'(c_2(z_1)) > u'(c_2(z_2))$ (because u is increasing and concave; standard assumptions that $u' > 0$ and $u'' < 0$; note that the later says that there is decreasing marginal utility). Combining $u'(c_2(z_1)) > u'(c_2(z_2))$ with $u'(c_1) = pu'(c_2(z_1)) + (1-p)u'(c_2(z_2))$ we get that $u'(c_2(z_2)) < u'(c_1) < u'(c_2(z_1))$ (take the 'weighted sum'/'average' of any two positive numbers and you will get something that is inbetween them).

Now we will look at the two period model with a shock in the second period that can take two values, again. But this time we will solve the complete markets version. For complete markets we will need two assets, with different returns depending on states, and the easiest way to do this will be to have asset a_1 which returns $(1+r)a_1$ when $z = z_1$ and zero when $z = z_2$, as well as asset a_2 which returns zero when $z = z_1$ and $(1+r)a_2$ when $z = z_2$.[37]

[37] These are essentially the 'Arrow-Debreu securities' (essentially as normally would have prices, rather than rate of return r). What matters is that we have enough assets with different returns to 'span' the space of the shocks (and time periods). An intuitive way to think about it is that we need to be able to trade between all the states independently of the others, in this model we have three states: period one, period two with shock one, and period two with shocks two. To be able to

The household problem is,

$$\max_{\{c_1,c_2(z_1),c_2(z_2),a_1,a_2\}} u(c_1)+\beta[pu(c_2(z_1))+(1-p)u(c_2(z_2))]$$
$$\text{s.t. } c_1+a_1+a_2=y_1$$
$$c_2(z_1)=(1+r)a_1+y_2(z_1)$$
$$c_2(z_2)=(1+r)a_2+y_2(z_2)$$

Note that, e.g., the zero return of a_2 is implicit in the period 2 state 1 budget contraint (you could write it explicity as $0 * a_2$). The Lagrangian is,

$$\begin{aligned}\mathcal{L}(c_1,c_2(z_1),c_2(z_2),a_1,a_2,\lambda_1,\lambda_2,\lambda_3)&=u(c_1)+\beta[pu(c_2(z_1))+(1-p)u(c_2(z_2))]+\lambda_1(y_1-c_1-a_1-a_2)\\&+\lambda_2((1+r)a_1+y_2(z_1)-c_2(z_1))+\lambda_3((1+r)a_2+y_2(z_2)-c_2(z_2))\end{aligned}$$

The first-order conditions of this are

$$\frac{\partial\mathcal{L}}{\partial c_1}=0;\quad u'(c_1)+\lambda_1(-1)=0$$
$$\frac{\partial\mathcal{L}}{\partial c_2(z_1)}=0;\quad p\beta u'(c_2(z_1))+\lambda_2(-1)=0$$
$$\frac{\partial\mathcal{L}}{\partial c_2(z_2)}=0;\quad (1-p)\beta u'(c_2(z_2))+\lambda_3(-1)=0$$
$$\frac{\partial\mathcal{L}}{\partial a_1}=0;\quad \lambda_1(-1)+\lambda_2(1+r)=0$$
$$\frac{\partial\mathcal{L}}{\partial a_2}=0;\quad \lambda_1(-1)+\lambda_3(1+r)=0$$
$$\frac{\partial\mathcal{L}}{\partial \lambda_1}=0;\quad c_1+a=y_1$$
$$\frac{\partial\mathcal{L}}{\partial \lambda_2}=0; c_2(z_1)=(1+r)a_1+y_2(z_1)$$
$$\frac{\partial\mathcal{L}}{\partial \lambda_3}=0; c_2(z_2)=(1+r)a_2+y_2(z_2)$$

From various combinations of the first five we get,

$$u'(c_1)=(1+r)\beta pu'(c_2(z_1))$$
$$u'(c_1)=(1+r)\beta(1-p)u'(c_2(z_2))$$
$$pu'(c_2(z_1))=(1-p)u'(c_2(z_2))$$

Complete markets make it possible to trade independently between any two states (we have three, period 1, period

shift between all of these we need one less asset than there are states; or you can think of it as as many assets as there are states, not counting the state we are already in.

2 shock 1, period 2 shock 2; note that, e.g., one more unit of a_1 and one less unit of a_2 'trades' between period 2 shock 1 and period 2 shock 2, with no impact on period 1). So complete markets mean we equalize the (discounted) marginal utility between any two states, adjusted for the probability of the state and for the 'price' of transfering between states ($1 + r$ between period one and either of the period two states, 1 between the two period 2 states).

The most important difference to notice between the incomplete markets and complete markets models we just solved is that in the complete markets economy we had that we equate (with adjustments for prices, discounting, and probability) the marginal utility for each and every state. By contrast in the incomplete markets model it was not always possible to equate between some states, in our case the two period 2 states, and so agents are in this sense more 'exposed' to the risk of bad shocks that arise in some states; they are less able to insure against them. Because agents cannot insure so well against the shocks it follows that they are more 'affected' by the shocks and end up very different to each other based on which shocks hit them over their lifetime. The idiosyncratic shocks that hit a household in a complete markets model are largely irrelevant as households are perfectly insured agains them; so different shocks do not give rise to differences between agents in a complete markets model.

3.5.1 Representative Agent

Complete Markets are not the same thing as a Representative Agent. But complete markets are a prerequisite (a necessary condition) for the existence of a Representative Agent. We won't attempt to fully explain the concept of a Representative Agent here as it involve competitive equilibrium, while none of the life-cycle models considered in this document are general equilibrium models (see the companion document on OLG models for the extension of life-cycle models to general equilibrium, linked in Section 5).

The basic idea is that instead of solving a model with lots of different households (whether due to ex-ante differences, or simply ex-post different due to receiving different idiosyncratic shocks) we can just solve a model with a Representative Agent[38] and get the same answer. Since Representative Agent models are easier to solve, if they give the same answer we are better off just using Representative Agents. The intuition for creating a Representative Agent is that any competitive equilibrium of a model containing lots of different households can be recreated (in terms of the aggregates, like total consumption, or total assets) as the competitive equlibrium of a model with a Representative Agent as long as we choose the preferences appropriately (as a weighted sum of the individual households, with the weights based on the lagrange multipliers of their maximization problem); and as long as markets are complete.

Obviously the idea that heterogeneity does not matter is a strong assumption, and one that often fails to be realistic. Representative Agent models are 'simpler' and more parsimonious than heterogenous agent models, and for that reason are still useful for some things where either heterogeneity is unimportant, or where it would simply

[38]Technically it is not one single agent, but a continuum of agents that are all identical, but we can solve it like there was one agent as all the other agents are identical. The continuum of agents is needed to allow that the model has a competitive equilibrium, rather than being a monopoly.

be impossible to solve the incomplete markets version of the model. But Representative Agents models should be treated with caution where they differ from heterogenous agent models. It is worth knowing that while you can create a Representative Agent that recreates all the same model aggregates, like consumption and assets, as would occur in a complete markets competitive equilibrium, you cannot create a Representative Agent for the welfare of these agents, and so all welfare analysis in Representative Agent models is actually not well microfounded. Just like it is still useful to teach very simple models in first-year undergraduate economics classes, it is still useful to use Representative Agent models in teaching.

Because we need the concept of competitive equilibrium to make sense of the details of Representative Agent models I won't attempt it here. A decent 'simple' example is given in 'Lecture 1: Complete Markets' by Florian Scheuer (if you know of a nicer explanation/example please let me know so I can link it here instead)

4 Further building up Life-Cycle Models

4.1 Life-Cycle model 20: Idiosyncratic shocks that depend on age

We will now look at how to set up shock that depend on age: both the grid points and the transition matrix can change with age. There are two ways to do this, either by creating matices *z_grid_J* and *pi_z_J* which contain *z_grid* and *pi_z* for each age, or by using a function that takes (age dependent) parameters as inputs and returns the grid and transition matrix for that age, called *ExogShockFn* in the codes. Life-Cycle Model A5 shows how to create *z_grid_J* and *pi_z_J* for an AR(1) process whose parameters depend on age. Here will will demonstrate how to use $ExogShockFn$.

We will use Life-Cycle Model 8, modifying it so that the transiton matrix for shocks depends on age (but the grid does not). That model has an exogenous shock z that takes two values, representing employment and unemployment, respectively. Empirically the unemployment rate is higher for the young, and job turnover is also higher for the young. We will therefore change the transition matrix so that the probability of remaining in the employment state increases in age; I have not attempted to calibrate the numbers seriously, they are purely illustrative.

We create an 'ExogShockFn' which takes in some parameters and returns *z_grid* and *pi_z*; the dependence on age comes because the parameters themselves (can) depend on age. If you look at the contents of ExogShockFn you can see how the transition matrix is created to depend on age.

The codes start all households at age $j = 1$ in the stationary distribution for the transition matrix on z at age $j = 1$. So if the transition matrix did not change with age then nor would the distribution of agents over z (the distribution of agents over a could change, but not over z). We plot the life-cycle profile showing the fraction of the population in the 'unemployment' state, and can see how it changes with age because the transition matrix on z is changing with age.

Note that the number of grid points on z must remain unchanged in age. Of course by making the probability of transitioning into some of them equal to zero you can effectively remove some.

4.2 Life-Cycle model 21: Idiosyncratic medical shocks in retirement

We just saw how to allow exogenous shock grids and transition path to depend on age. When doing this we can easily also reinterpret these exogenous shocks to have different meanings at different ages. Notice how we give significance to the values of z via the return function and the FnsToEvaluate. So if we change the return function and FnsToEvaluate to interpret z (or any other state) in a different way at different ages then the meaning of z is different for different ages. We have already seen in Life-Cycle Model 20 (and A5) how to change the grid values and transition matrix for z for different ages, so this makes it easy to completely repurpose z at different ages.

We will modify Life-Cycle Model 8, which had a shock z that takes two values representing employment and unemployment, respectively. Notice that in that model retirees never work, and so the shock was irrelevant/ignored for all retirees. So let's repurpose the shock during retirement. We know retirees often do not run down their savings as a basic life-cycle model would predict. We have already seen one possible reason, that they want to leave bequests.[39] Another possible reason that the elderly do not run down savings is that they want substantial savings to pay for potentially large medical expenses. We will repurpose the shock z to be a medical expense shock during retirement, to create precisely this motive.[40]

So we are going to have the exogenous shock z represent employment/unemployment during working age, and then medical expenses during retirement. We use ExogShockFn to do this, and will just set one grid for working age and another grid for retirement; likewise we set one transition matrix for working age and another for retirement.[41]

Note that the number of grid points on z must remain unchanged in age. Of course by making the probability of transitioning into some of them equal to zero you can effectively remove some.

If you run this model and Life-Cycle Model 8, the only difference between the two is the medical shocks, and you will be able to see their impact in increased savings by the elderly in the life-cycle profiles.

4.3 Life-Cycle model 22: Deterministic Economic/productivity growth

The economy grows over time, as do incomes, as productivity rises. We will add a deterministic rate of growth of producitivity, g, at which incomes rise. The way to solve a model with deterministic growth is to look for a

[39] We modelled warm-glow bequests, but there were more a simply way to capture something like wanting to leave model for descendents (or to charity), and should be understood as a simple way to capture a complex behaviour, rather than the 'warm glow' being taken too seriously.

[40] Obviously the importance of medical expense shocks is likely to be much larger in countries with private health systems vs public health systems. The literature also points to the potentially large costs of elderly-care in nursing homes and the like as important to the savings of the elderly.

[41] We could of course set these to change with age —e.g., medical shocks become more likely as households get older— but to keep things easy to follow we will just keep the same grids and transition matrices except for the change from working age to retirement.

balanced growth path by converting the model into 'per technology unit' terms, which makes the model stationary. Then solve the stationary model, and then if wanted we can add growth back into the simulations of the stationary model.[42] Solving the stationary model can be done in the usual manner.

To be able to do this we need to use a non-seperable utility function.[43] Conceptually the reason is simple enough. If incomes grow (which is the point of our deterministic productivity growth g) then people will consume more and so the marginal utility of consumption falls. Then seperable utility implies that equating marginal utilities therefore requires a falling marginal utility of leisure, which means ever increasing leisure. So seperable utility together with growing incomes drives hours worked to decrease towards zero. With non-seperable utility the marginal utility of leisure is increasing in consumption (this is meaning of non-seperable) and so (optimal) leisure does not continually increase as consumption increases, and so hours worked to not tend to zero.

For the renormalization to work we need to be careful about what exactly grows. When using a model with exogenous labor we would have the endowment income grow at rate g, whereas when using a model with endogenous labor we need it to be the wage (per unit of labor supply/per hour worked) that grows at rate g.

A brief further discussion of what kinds of utility functions can be used with models that have growth is appropriate. You are probably familiar with the neoclassical growth model (a.k.a. Solow or Solow-Swan model). The general characterization of prefererences consistent with the existence of a steady-state in a model with deterministic producitivity growth are 'Uzawa prefereces'. Two extensions may also be of interest. The first is the extension of Uzawa preferences to models with endogneous human capital, by Grossman, Helpman, Oberfield, and Sampson (2017). The second is Boppart and Krusell (2020) which provides preferences consistent with the combination of both income growth and moderate decrease in hours worked in high income countries over the 20th century.

We will first do two models without idiosyncratic shocks. Let's start with the exogenous labor case, as it is slightly simpler, and then do the endogenous labor case. After these we solve the endogenous labor case with idiosyncratic shocks. For Assignment 4 you are provided with just the model set up for exogenous labor supply with idiosyncratic shocks and deterministic income growth; solving the equations to renormalize the the exogenous labor case is left as Assignment 4 (a code 'solution' is provided).

4.3.1 Deterministic income growth with exogenous labor supply

I will explain the method using a very simple life-cycle model to make clear the concepts, and then simply write out what Life-Cycle Model 10 becomes with deterministic growth.

[42] Notice that this is exactly the same thing as when solving the neoclassical growth model (a.k.a. Solow or Solow-Swan model); solve the balanced growth path by dividing by the technology level (and population), and then solving for the steady-state. Note that the difference between stationary equilibrium and steady-state is minor, they are often used interchangably in conversation.

[43] Non-seperable utility was discussed Assignment 2 (just after Life-Cycle model 10).

Consider the following household problem,

$$\sum_{j=1}^{J} \frac{c_j^{1-\sigma}}{1-\sigma}$$
$$c_j + a_{j+1} = (1+r)a_j + y_j$$
$$y_j = (1+g)y_{j-1}, \quad y_1 \text{ given}$$

Notice that y_j is growing at rate g; the deterministic income growth.

Trying to compute the solution to this problem in it's current form is possible (because there is a final period, there is a maximum amount that income will end up being)[44], but we would require very large grids as we would need to deal with the smallest income in period 1 and the largest income in period J, as well as the wide range of asset holdings that would result. We will therefore rewrite everything in such a way as to 'remove' the growth. The intuition is that $y_j = (1+g)^{j-1}y_1$, and so if we can just divide everything by $(1+g)^{j-1}$ then the growth would 'disappear'.

First, rewrite y_j, to get

$$\sum_{j=1}^{J} \beta^{j-1} \frac{c_j^{1-\sigma}}{1-\sigma}$$
$$c_j + a_{j+1} = (1+r)a_j + (1+g)^{j-1}y_1$$

and now divide the budget constraint through by $(1+g)^{j-1}$ to get

$$\sum_{j=1}^{J} \beta^{j-1} \frac{c_j^{1-\sigma}}{1-\sigma}$$
$$\frac{c_j}{(1+g)^{j-1}} + \frac{a_{j+1}}{(1+g)^{j-1}} = (1+r)\frac{a_j}{(1+g)^{j-1}} + y_1$$

We want this to be simpler, notice that we could 'get rid' of the denominator on c_j by defining a 'renormalized' variable $\hat{c}_j \equiv \frac{c_j}{(1+g)^{j-1}}$. We could do the same with a_j, defining $\hat{a}_j \equiv \frac{a_j}{(1+g)^{j-1}}$. We would then get

$$\sum_{j=1}^{J} \beta^{j-1} \frac{(\hat{c}_j(1+g)^{j-1})^{1-\sigma}}{1-\sigma}$$
$$\hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + y_1$$

[44]This would not be true in an infinite horizon model, nor if we extended this to an OLG model.

notice both that we made the substitution in the utility function as well, notice also how we could substitute $\frac{a_{j+1}}{(1+g)^{j-1}}$ with $(1+g)\hat{a}_{j+1}$.

We have one more step, and for this step the functional form of the utility function becomes important (especially in the model with endogenous labor). Right now the utility function 'changes' every period (there is a j inside it), but we can pull the $(1+g)^{j-1}$ out in front to get

$$\sum_{j=1}^{J} \beta^{j-1}((1+g)^{j-1})^{1-\sigma} \frac{(\hat{c}_j)^{1-\sigma}}{1-\sigma}$$
$$\hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + y_1$$

and with a little bit of rewriting the powers, we get

$$\sum_{j=1}^{J} \beta^{j-1}((1+g)^{1-\sigma})^{j-1} \frac{(\hat{c}_j)^{1-\sigma}}{1-\sigma}$$
$$\hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + y_1$$

The important thing here is that now neither the utility function, $\frac{(\hat{c}_j)^{1-\sigma}}{1-\sigma}$, nor the budget constraint $\hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + y_1$ has parameters that change with the period j (when we started y_j changed with j). In fact the deterministic growth now just appears as an additional discount factor, note that we could group the discounts factors to get, $(\beta(1+g)^{1-\sigma})^{j-1}$. So solving the model with deterministic growth, after renormalizing, just looks like a model with a different discount factor, and that is easy enough to solve.

Note that if the model had retirement and pensions, then you would either end up with the renormalized pension decreasing in j, or you could consider pensions which are indexed to the growth (if you are looking at a specific country you should follow the pension legislation).

Note, no code is provided to solve this model, you can hopefully figure it out based on the code for the model with deterministic wage growth and endogenous labor supply and idiosyncratic shocks that follows shortly.

4.3.2 Deterministic income growth with endogenous labor supply

Consider the following household problem,

$$\sum_{j=1}^{J} \beta^{j-1} \frac{c_j^{\sigma_1} l_j^{1-\sigma_1})^{1-\sigma_2}}{1-\sigma_2}$$
$$c_j + a_{j+1} = (1+r)a_j + w_j(1-l_j)$$
$$w_j = (1+g)w_{j-1}, \quad w_1 \text{ given}$$

Notice that w_j is growing at rate g; the deterministic wage growth.

There are two key differences with endogenous labor. The first is the utility function which must be non-seperable. We already discussed the intuition for this and we will see mathematically that the non-seperability is needed to be able to get the $(1+g)^{j-1}$ term out of the (renormalized) utility function so that it just becomes an additional discount factor. The second is that now we have the deterministic growth in the wage, w_j and not in the labor income $w_j(1-l_j)$[45]

From here the steps are almost exactly the same as with exogenous labor supply, first we can divide the whole budget constraint by $(1+g)^{j-1}$ and define $\hat{c}_j \equiv \frac{c_j}{(1+g)^{j-1}}$ and $\hat{a}_j \equiv \frac{a_j}{(1+g)^{j-1}}$; note that we do not change l_j, because it multiplies the w_j. Substituting we get

$$\begin{aligned} \sum_{j=1}^{J} \beta^{j-1} \frac{(\hat{c}_j (1+g)^{j-1})^{\sigma_1} l_j^{1-\sigma_1})^{1-\sigma_2}}{1-\sigma_2} \\ \hat{c}_j + \hat{a}_{j+1} = (1+r)\hat{a}_j + w_1(1-l_j) \\ w_j = (1+g)w_{j-1}, \quad w_1 \text{ given} \end{aligned}$$

note the l_j and w_1. We can now do exactly the same step of taking the growth term out of the utility function and making it appear as just an extra discount factor,

$$\begin{aligned} \sum_{j=1}^{J} (\beta(1+g)^{\sigma_1(1-\sigma_2)})^{j-1} \frac{(\hat{c}_j)^{\sigma_1} l_j^{1-\sigma_1})^{1-\sigma_2}}{1-\sigma_2} \\ \hat{c}_j + \hat{a}_{j+1} = (1+r)\hat{a}_j + w_1(1-l_j) \\ w_j = (1+g)w_{j-1}, \quad w_1 \text{ given} \end{aligned}$$

where the discount factor is now $\beta(1+g)^{\sigma_1(1-\sigma_2)}$.

Note, no code is provided to solve this model, you can hopefully figure it out based on the code for the model with deterministic wage growth and endogenous labor supply and idiosyncratic shocks that follows shortly.

4.3.3 Deterministic income growth with endogenous labor supply and exogenous shocks

Let's extend Life-Cycle Model 9 to include deterministic income growth, as part of which we will have to switch to a non-seperable utility function. To make things simple I am going to index pensions to the growth of wages.[46]

[45] The reason for this is more obvious in an OLG model where things will fail to add up at the aggregate (whole economy) level unless we put the deterministic growth in the wage. In the present life-cycle it is, roughly speaking, because labor supply is a decision variable that gets determined endogenously and so is not something that we can renormalize in terms of.

[46] This is not necessary, just makes the renormalized model a bit simpler.

So we start with the household problem,

$$
\begin{aligned}
V(a_j, z_j, j) \quad = \quad & \max_{c_j, a_{j+1}, h_j} \frac{c_j^{\sigma_1}(1-h_j)^{1-\sigma_1})^{1-\sigma_2}}{1-\sigma_2} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(a_{j+1}) \\
& +s_j \beta E[V(a_{j+1}, z_{j+1]}, j+1)|z] \\
& \text{if } j < Jr: \; c_j + a_{j+1} = (1+r)a_j + w_j \kappa_j z_j h_j \\
& \text{if } j >= Jr: \; c_j + a_{j+1} = (1+r)a_j + pension_j \\
& 0 \leq h \leq 1, a_{j+1} \geq 0 \\
& log(z_j) = \rho_z log(z_{j-1}) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z}) \\
& w_j = (1+g)w_{j-1}, \quad w_1 \text{ given} \\
& pension_j = (1+g)pension_{j-1}, \quad pension_{Jr} \text{ given}
\end{aligned}
$$

notice that both wage, w_j, and pension, $pension_j$ grow at deterministic rate g (from w_1 and $pension_{Jr}$, respectively). Note that I have had to change notation slightly from Life-Cycle Model 9, in the sense that we use c_j rather than c, and similarly for the other variables, purely because it makes the renormalization look more like those we have already seen.

We can do the same renormalization as before, defining $\hat{c}_j \equiv \frac{c_j}{(1+g)^{j-1}}$ and $\hat{a}_j \equiv \frac{a_j}{(1+g)^{j-1}}$, and leaving h_j and z_j unchanged. We can substitute these in and then extract the term on the utility function to be an additional discount factor to get,

$$
\begin{aligned}
V(\hat{a}_j, z_j, j) \quad = \quad & \max_{\hat{c}_j, \hat{a}_{j+1}, h_j} \frac{\hat{c}_j^{\sigma_1}(1-h_j)^{1-\sigma_1})^{1-\sigma_2}}{1-\sigma_2} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(\hat{a}_{j+1}(1+g)^j) \\
& +s_j \beta E[V(\hat{a}_{j+1}, z_{j+1]}, j+1)|z] \\
& \text{if } j < Jr: \; \hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + w_1 \kappa_j z_j h_j \\
& \text{if } j >= Jr: \; \hat{c}_j + (1+g)\hat{a}_{j+1} = (1+r)\hat{a}_j + pension_{Jr} \\
& 0 \leq h \leq 1, \hat{a}_{j+1} \geq 0 \\
& log(z_j) = \rho_z log(z_{j-1}) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z})
\end{aligned}
$$

note that we now just have w_1 and $pension_{Jr}$. Notice also the $warmglow(\hat{a}_{j+1}(1+g)^j)$; we could just take the $(1+g)^j$ out of the warmglow function as really it is just equivalent changing two of the warm-glow parameters, $warmglow1$ and $warmglow2$. When coding we will take advantage of this to rewrite the warm-glow as $(1-s_j)\beta(1+g)^{-1}\mathbb{I}_{(j>=Jr+10)} warmglow(\hat{a}_{j+1}(1+g))$, as mentioned this is really just reparametrizing, not changing anything of substance.

We will solve this model in the standard way, and simulate some panel data with it. Note that what we simulate will be $\hat{a}$, $w_1\kappa_j z_j h$, and h. We will then create panel data for the original model, by changing these to a, $w_j\kappa_j z_j h$ and h; this requires multiplying the first two by $(1+g)^{j-1}$ but leaving the third unchanged (we need to modify to get a and w_j).

4.3.4 Assignment 4: Deterministic income growth with exogenous labor supply and exogenous shocks

Life-Cycle Model 10, with deterministic growth in the stochastic endowment is given by,

$$
\begin{aligned}
V(a,z,j) \quad &= \quad \max_{c,aprime} \frac{c^{1-\sigma}}{1-\sigma} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\
&\qquad\qquad +s_j \beta E[V(aprime, zprime, j+1)|z] \\
&\text{if } j < Jr: \; c + aprime = (1+r)a + w_j kappa_j z \\
&\text{if } j >= Jr: \; c + aprime = (1+r)a + pension_j \\
&aprime \geq 0 \\
&log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z}) \\
&w_j = (1+g) w_{j-1}, \quad w_1 \text{ given} \\
&pension_j = (1+g) pension_{j-1}, \quad pension_{Jr} \text{ given}
\end{aligned}
$$

for Assignment 4 you first need to convert this model into one where there is a stationary equilibrium (by normalizing by the income growth), and then solve the model with code. Then use the solution to the stationary model to simulate some panel data, and then put the deterministic income growth back into the panel data.

We mentioned previously that with exogenous labor supply it is income that grows, while with endogenous labor it is important that it is the wage that grows. So how come we have the growth in the 'wage' in this exogenous growth model? Because a 'wage' is somewhat meaningless for exogenous labor, and notice that there is no difference between $w_j kappa_j z$ growing and w_j growing if they are all constants (or stationary stochastic processes).

A solution code for Assignment 4 is provided, but try to solve without looking at it.

4.4 Life-Cycle model 23: Quasi-Hyperbolic Discounting

Quasi-Hyperbolic discounting is a way to model 'impatience'. Impatient households are those that take decisions today, which put little weight on the future, and which their future-self would not like. Every model until now has used 'exponential discounting', with every future period discounted by factor β. The idea of quasi-hyperbolic discounting is that while you still use β to discount between any two future periods, you use an additional discount factor β_0 to discount between the current period and next period; so you take decisions today that put little weight on the future.

There are two types of quasi-hyperbolic discounting, called *sophisticated* and *naive*, based on whether are sophisticated and recognise that your future-self will suffer the same impatience problems as you do, or whether you are naive and simply (incorrectly) assume your future-self will not suffer from the same impatience, respectively.

We will use Life-Cycle Model 9, and simply change to quasi-hyperbolic discounting. This is easy to code as we essentially just tell $vfoptions$ to use quasi-hyperbolic discounting and add the 'additional' discount factor. Near the start of the codes we choose which of naive and sophisticated quasi-hyperbolic discounting to use. Notice that quasi-hyperbolic discounting affects how the household makes decisions are therefore needs to be in $vfoptions$, but it is irrelavant once the policy function is known, and therefore is not in $simoptions$.

We start with the naive quasi-hyperbolic discounting problem. Notice first that the naive quasi-hyperbolic discounter (naively) believes their future self will act like an exponential discounter. So we first define the 'continuation value function' which is just the exponential discounting value function,

$$
\begin{aligned}
V(a,z,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
& +s_j\beta E[V(aprime,zprime,j+1)|z] \\
& \text{if } j<Jr:\ c+aprime=(1+r)a+w\kappa_j zh \\
& \text{if } j>=Jr:\ c+aprime=(1+r)a+pension \\
& 0\leq h\leq 1, aprime\geq 0 \\
& log(zprime)=\rho_z log(z)+\epsilon,\ \epsilon\sim N(0,\sigma^2_{\epsilon,z})
\end{aligned}
$$

we can then define the naive quasi-hyperbolic discounters value function, which we denote $\tilde{V}$ in terms of this,

$$
\begin{aligned}
\tilde{V}(a,z,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta_0\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
& +s_j\beta_0\beta E[V(aprime,zprime,j+1)|z] \\
& \text{if } j<Jr:\ c+aprime=(1+r)a+w\kappa_j zh \\
& \text{if } j>=Jr:\ c+aprime=(1+r)a+pension \\
& 0\leq h\leq 1, aprime\geq 0 \\
& log(zprime)=\rho_z log(z)+\epsilon,\ \epsilon\sim N(0,\sigma^2_{\epsilon,z})
\end{aligned}
$$

There are a few things worth noting. That on the right-hand side we have the 'continuation' next period value function V, while on the left we have the naive quasi-hyperbolic discounters value function $\tilde{V}$. That the discount factor is $\beta_0\beta$, as we are applying the additional discount factor β_0 that captures quasi-hyperbolic discounting.[47]

[47] I also apply β_0 to discounting the warm-glow.

Now we can think about the sophisticated quasi-hyperbolic discounter. Again we will need to define a 'continuation' value function, but now it is more complex as the sophisticated quasi-hyperbolic discounter knows that their future self will follow the policy function of a sophisticated quasi-hyperbolic discounter, but discount between future periods based on the exponential discount factor (see document linked below for explantion). We will go the reverse order, first defining the value function of the sophisticated quasi-hyperbolic discounter, $\hat{V}$,

$$\begin{aligned}
\hat{V}(a,z,j) \quad &= \quad \max_{\hat{c},\hat{a}prime,\hat{h}} \frac{\hat{c}^{1-\sigma}}{1-\sigma} - \psi\frac{\hat{h}^{1+\eta}}{1+\eta} + (1-s_j)\beta_0\beta\mathbb{I}_{(j>=Jr+10)}warmglow(\hat{a}prime) \\
&\qquad + s_j\beta_0\beta E[\underline{\mathrm{V}}(\hat{a}prime, zprime, j+1)|z] \\
&\qquad \text{if } j < Jr: \ \hat{c} + \hat{a}prime = (1+r)a + w\kappa_j z\hat{h} \\
&\qquad \text{if } j >= Jr: \ \hat{c} + \hat{a}prime = (1+r)a + pension \\
&\qquad 0 \leq \hat{h} \leq 1, \hat{a}prime \geq 0 \\
&\qquad log(zprime) = \rho_z log(z) + \epsilon, \ \epsilon \sim N(0, \sigma^2_{\epsilon,z})
\end{aligned}$$

notice that I have denoted the policy variables with hats, $\hat{c}, \hat{a}prime, \hat{h}$. Note that we have the additional discount factor β_0. The continuation value function is now $\underline{\mathrm{V}}$ which is the policies of the sophisticated quasi-hyperbolic discounter but evaluated with the exponential discount factor, and is given by

$$\begin{aligned}
\underline{\mathrm{V}}(k,z,j) \quad &= \quad \frac{\hat{c}^{1-\sigma}}{1-\sigma} - \psi\frac{\hat{h}^{1+\eta}}{1+\eta} + (1-s_j)\beta_0\beta\mathbb{I}_{(j>=Jr+10)}warmglow(\hat{a}prime) \\
&\qquad + s_j\beta_0\beta E[\underline{\mathrm{V}}(\hat{a}prime, zprime, j+1)|z]
\end{aligned}$$

where in a slight abuse of notation I am now using $\hat{c}, \hat{a}prime, \hat{h}$ to signify the optimal policies (the argmax) from the value function problem of the sophisticated quasi-hyperbolic discounter.

This example is not really supposed to teach you quasi-hyperbolic discounting, it is more to explain how to implement it. If you want to understand quasi-hyperbolic discounting, both the theory and how to use it with VFI Toolkit in both infinite and finite-horizon models, check: https://www.vfitoolkit.com/updates-blog/2021/exotic-preferences-epstein-zin-quasi-hyperbolic/

4.5 Life-Cycle model 24: Permanent Types: Solving fixed-types

We will use permanent types, which is a way to solve N_i household models at once. Our first model is just that of Life-Cycle model 11, with endogenous labor and both persistant and transitory shocks to labor efficiency units. In this model we add a 'fixed effect' to the labor efficiency units, denoted α_i, something common in the literature. This is the only change to the model. Permanent types, or PTypes, is how VFI Toolkit handles different types of agents, it is much more general than just fixed-types as will be seen in later Life-Cycle models, but for now we simply use it for this purpose.

We add a parameter, α_i, which depends on the permanent types indexed by i. We will use N_i different permanent types of agents, and simply by setting the parameter α_i to have N_i different values the toolkit will automatically recognise that this parameter depends on the permanent type and respond appropriately.

Our household value function now includes a fixed-effect α_i to determine idiosyncratic producitivity units,

$$
\begin{aligned}
V_i(a,z,e,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\
& \quad + s_j \beta E[V_i(aprime, zprime, eprime, j+1)|zprime] \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j \alpha_i zeh \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& 0 \leq h \leq 1, aprime \geq 0 \\
& log(zprime) = \rho_z log(z) + \epsilon, \; \epsilon \sim N(0, \sigma^2_{\epsilon,z}) \\
& log(e) \sim N(0, \sigma^2_e)
\end{aligned}
$$

Notice that z is AR(1) and e is i.i.d. normal. Note that the codes use 'e' variables for i.i.d. shocks, see Life-Cycle model 11 if you have not previously seen this feature.

Notice that the value function now depends on i: V_i. So we are now essentially solving N_i seperate problems.

In the codes we will use $N\_i = 5$. We just set parameter α_i to be a vector of length N_i and the toolkit will automatically realise that this parameter depends on the permanent type and act appropriately. The other noteworthy change is that we have to switch all the commands to *_PType*, e.g., *ValueFnIter_Case1_FHorz_PType* instead of *ValueFnIter_Case1_FHorz*, and similarly for all commands around the agent distribution, function evaluation, etc. Of course the 'shape' of the results changes: V now becomes a structure, with *V.ptype*001,..., *V.ptype*005 being the value functions for each of the five permanent types.[48] Similarly thinks like the agent distribution will now have one distribution for each permanent type. When we calculate something like life-cycle profiles the output now provides both a result conditional on each permanent type, and a grouped result.

While it is not necessary for solving the value function problem we do need to state the weights of each of the different permanent types to solve the stationary distribution. This is a vector-parameter which we will call *alphadist*, and we need to put the name of this in *PTypeDistParamNames*, which we then pass to the stationary distribution commands. This information gets encoded into the StationaryDist, and so we do not need to include it seperately later when doing things like life-cycle profiles.

This example plots life-cycle profiles for α_i, both for each permanent type, and grouped across the permanent types. This provides a nice example to show how the toolkit has interpreted α_i is a parameter that differs by permanent type, and how the grouped value is simply the values of α_i summed according to their weights in *alphadist*.

[48] *ptype*001, etc., are the default 'names' given to the fixed-types, as later Life-Cycle model examples using permanent types will make clear we can actually choose our own names.

4.6 Life-Cycle model 25: Using Names for Permanent Types: patient and impatient

We consider a model with two agents who differ by the value of their discount factor parameter, β. We can use names for the agents, and so call them 'patient' and 'impatient'. The model is just that of Life-Cycle Model 11, except that there are now two agents with different values of β_i, indexed by i. The main purpose of this example is to show how we can use names for permanent types of agents in codes.

Our household value function now includes two permanent types that differ by their value of β_i, the discount factor parameter.

$$\begin{aligned} V_i(a,z,e,j) \quad = \quad & \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta_i \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\ & \quad +s_j \beta E[V_i(aprime,zprime,eprime,j+1)|zprime] \\ & \text{if } j < Jr: \ c + aprime = (1+r)a + w\kappa_j zeh \\ & \text{if } j >= Jr: \ c + aprime = (1+r)a + pension \\ & 0 \leq h \leq 1, aprime \geq 0 \\ & log(zprime) = \rho_z log(z) + \epsilon, \ \epsilon \sim N(0,\sigma^2_{\epsilon,z}) \\ & log(e) \sim N(0,\sigma^2_e) \end{aligned}$$

Notice that z is AR(1) and e is i.i.d. normal. Note that the codes use 'e' variables for i.i.d. shocks, see Life-Cycle model 11 if you have not previously seen this feature.

Rather than use $N\_i = 2$, like we did in Life-Cycle model 24, we can instead give the agents names using $Names\_i = \{'patient', 'impatient'\}$. The PType commands will automatically use these names for all outputs. We set the different parameter values as $Params.beta.patient = 0.96$ and $Params.beta.impatient = 0.9$. After this, other than using $Names\_i$ everywhere we had N_i in Life-Cycle model 24, there is no other change we need to make.

The names we give the permanent types are automatically used for output, hence we get $V.patient$ and $V.impatient$ as the two value functions for the agent types. Commands for things like life-cycle profiles give us both the two life-cycle profiles conditional on each agent type (e.g., $AgeConditionalStats.earnings.patient.Mean$ and $AgeConditionalStats.earnings.impatient.Mean$) and the life-cycle profiles for the whole population (e.g., $AgeConditionalStats.earnings.Mean$).

4.7 Life-Cycle model 26: More Permanent Types

There are lots of other things you can do with permanent types: different return functions, different number of periods, different exogenous shock processes, and much much more. Hopefully more examples coming soon :D

If there is a particular thing you think would be good to see, please email or use forum: discourse.vfitoolkit.com.

4.8 Life-Cycle model 28: Two decision variables (dual-earner household)

We solve a model of a household in which there are two people (e.g., a married household). They make a joint decision about how much each will work. This involves two decisions variables for the two labor supply choices. We will set up each household member to have effective hours shocks that are a combination of one AR(1) persistent shock and one i.i.d. shock (as in Life-Cycle model 11), for each person (so two of each for the household). We will also allow for correlation between the persistent shocks of the two household members, and for correlation between the transitory shocks of the two household members. We will also allow different deterministic age-dependent labor efficiency units, κ_j, for each spouse.

Our household value function now has $z1$ and $e1$ for the first spouse, and $z2$ and $e2$ for the second spouse. We also have $\kappa_{j,1}$ and $\kappa_{j,2}$. We add a disutility term for each spouses labor supply, but the household has joint consumption.

$$
\begin{aligned}
V(a, z_1, z_2, e_1, e_2, j) \quad = \quad & \max_{c, aprime, h_1, h_2} \frac{c^{1-\sigma}}{1-\sigma} - \psi \frac{h_1^{1+\eta}}{1+\eta} - \psi \frac{h_2^{1+\eta}}{1+\eta} + (1-s_j)\beta \mathbb{I}_{(j>=Jr+10)} warmglow(aprime) \\
& \quad + s_j \beta E[V(aprime, z_1prime, z_2prime, e_1prime, e_2prime, j+1) | z_1, z_2] \\
& \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_{j,1} z_1 e_1 h_1 + w\kappa_{j,2} z_2 e_2 h_2 \\
& \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
& 0 \leq h \leq 1, aprime \geq 0 \\
& log([z_1prime; z_2prime]) = \rho_z log([z_1; z_2]) + \epsilon, \; \epsilon \sim N(0, \Sigma_{\epsilon,z}^2) \\
& log([e_1, e_2) \sim N(0, \Sigma_e^2)
\end{aligned}
$$

Notice that $[z_1, z_2]$ is VAR(1) (in logs) and $[e_1, e_2]$ is i.i.d. normal (in logs).

We are allowing for z_1 and z_2 to follow a VAR(1) (so both ρ_z and $\Sigma_{\epsilon,z}$ are 2-by-2 matrices). We do not require that $\Sigma_{\epsilon,z}$ be diagonal, so the innovations themselves can be correlated. We similarly allow for e_1 and e_2 to be correlated (so Σ_e is not diagonal).

Correlated shocks, which use jointly-determined grids, are explained in Life-Cycle model A9.

This model demonstrates how to use two decision variables, you need to put them as the first two entries to the ReturnFn and also to all FnsToEvaluate. The model output includes calculating the life-cycle labor supply for each spouse seperately, as well as the total household labor supply.

A common variation of the model used in practice would add a fixed cost of working (e.g., $-\mathbb{I}_{h_2>0} * fc$, where fc is a constant) to capture that the second spouse in many households will sometimes supply zero labor.[49] Note that implementing this is just a simple modification of the ReturnFn.

Because the household has two labor supply decisions the two spouses are able to insure each other against

[49] Because we have $0 \leq h_2 \leq 1$ together with the shape of the utility function it would never be optimal to set $h_2 = 0$ in the absence of a fixed cost of working non-zero hours.

poor labor market outcomes. This is known as 'intra-household' insurance. If you are interested in this kind of model, see Ortigueira and Siassi (2013).

5 What Next?

If you want to learn to solve more complex life-cycle models you will need to learn how to code them yourself, and the numerical methods used to do so. There are a lot of materials online,[50] and a fast method like endogenous grid points for solving the value function problem is probably a good place to start. If you just want to use one asset life-cycle models then the VFI Toolkit is good enough for almost everything, with sole exception of 'simulated likelihood estimation' of the models (typically from panel data).[51]

If you are interested in building Macroeconomic models then OLG models are the next obvious step. In life-cycle models we just have individual households. An OLG model combines the life-cycle model of a household with general equilibrium to create a model of the entire economy. Take a look at: An Introduction to OLG models.[52]

If you have any questions about the material, or spot a typo in the codes, or would just like to ask a clarifying question, etc., please use the forum: discourse.vfitoolkit.com

[50]Check https://github.com/KennethJudd/CompEcon2020 for the underlying theory on computation, there are no end of more applied resources and where you should look depends heavily on what kind of problem you want to solve.

[51]Simulated likelihood estimation requires solving the model a lot of times, say 1 million times. If you take the run time for the codes here and multiply them by 1 million you will see how long that would take.

[52]I have not yet uploaded this, I will make this a link once it is ready, likely early 2022.

References

Timo Boppart and Per Krusell. Labor supply in the past, present, and future: A balanced-growth perspective. *Journal of Political Economy*, 128(1), 2020. doi: https://doi.org/10.1086/704071.

Christopher Carroll. Theoretical foundations of buffer stock saving. *Quantitative Economics*, Forthcoming:AAA, 2021.

Christopher Carroll, Edmund Crawley, Jiri Slacalek, Kiichi Tokuoka, and Matthew White. Sticky expectations and consumption dynamics. *American Economic Journal: Macroeconomics*, 12(3):40–76, 2020.

Simone Civales, Luis Diez-Catalan, and Fatih Fazilet. Discretizing a process with non-zero skewness and high kurtosis. *Unpublished manuscript*, 1:1, 2017.

Leland E. Farmer and Alexis Akira Toda. Discretizing nonlinear, non-gaussian markov processes with exact conditional moments. *Quantitative Economics*, 8(2):651–683, 2017.

Giulio Fella, Giovanni Gallipoli, and Jutong Pan. Markov-chain approximations for life-cycle models. *Review of Economic Dynamics*, 34:183–201, 2019.

Gene M. Grossman, Elhanan Helpman, Ezra Oberfield, and Thomas Sampson. Balanced growth despite uzawa. *American Economic Review*, 107:4, 2017.

Ayse Imrohoroglu. Cost of business cycles with indivisibilities and liquidity constraints. *Journal of Political Economy*, 97(6):1368–1383, 1989.

Greg Kaplan. Inequality and the life-cycle. *Quantitative Economics*, 3(3):471–525, 2012.

Fatih Karahan and Serdar Ozkan. On the persistence of income shocks over the life cycle: Evidence, theory, and implications. *Review of Economic Dynamics*, 16(3):452–476, 2013. doi: https://doi.org/10.1016/j.red.2012.08.001.

Karen Kopecky and Richard Suen. Finite state markov-chain approximations to highly persistent processes. *Review of Economic Dynamics*, 13(3):701–714, 2010.

Bartosz Maćkowiak and Mirko Wiederholt. Business cycle dynamics under rational inattention. *Review of Economic Studies*, 82(4):1502–1532, 2015. doi: https://doi.org/10.1093/restud/rdv027.

Salvador Ortigueira and Nawid Siassi. How important is intra-household risk sharing for savings and labor supply? *Journal of Monetary Economics*, 2013. doi: https://doi.org/10.1016/j.jmoneco.2013.05.007.

Jeno Pál and John Stachurski. Fitted value function iteration with probability one contractions. *Journal of Economic Dynamics and Control*, 37:251–264, 2013.

Josep Pijoan-Mas. Precautionary savings or working longer hours? *Review of Economic Dynamics*, 9(2):326–352, 2006.

G. Rouwenhorst. Asset pricing implications of equilibrium business cycle models. In Cooley T., editor, *Frontiers of Business Cycle Research*, chapter 10. Princeton University Press, 1995.

Kenichiro Tanaka and Alexis Akira Toda. Discrete approximations of continuous distributions by maximum entropy. *Economics Letters*, 118(3):445–450, 2013.

George Tauchen. Finite state markov-chain approximations to univariate and vector autoregressions. *Economics Letters*, 20:177–181, 1986.

George Tauchen and Robert Hussey. Quadrature-based methods for obtaining approximate solutions to nonlinear asset pricing models. *Econometrica*, 59(2):371–396, 1991.

A Alternative exogenous shock processes

These are a variety of life-cycle models that modify Life-Cycle Models 9 and 10 to illustrate a variety of alternative kinds of exogenous shocks that you could use. The first five have a single exogenous state, and so modify Life-Cycle Model 9. The models A6 and A7 both use two exogenous shocks that interact, and so modify Life-Cycle Model 10.

A.1 Life-Cycle Model A1: AR(1) persistent shocks, alternative quadrature methods

We revisit Life-Cycle Model 9, which has AR(1) shocks that we discretized using the Farmer-Toda method (Farmer and Toda, 2017). Methods of discretizing shocks are known as 'quadrature methods'.[53] The Farmer-Toda quadrature method is just one approach to turning AR(1) processes into a discrete markov process (a grid and a transition matrix). In Life-Cycle Model A1 we show how to implement three alternatives: Rouwenhorst, Tauchen, and Tauchen-Hussey.[54] These alternatives have been widely used in the past, but the Farmer-Toda method performs better so I recommend you always use it, except when the autocorrelation coefficient of your AR(1) is ≥ 0.99 in which case Rouwenhorst method is recommended.[55]

Since there is no change in the model itself we will not repeat it here, see Life-Cycle Model 9. As a matter of good practice I recommend always checking that your results are not overly sensitive to your discretization (e.g., if you use 5 grid points, makes sure changing to 7 does not massively impact your results), you should also always describe in your paper which method you use to discretize shocks (so that other people can understand what you did; put it in the appendix, but it should be there somewhere).

A.2 Life-Cycle Model A2: AR(1) persistent shocks with gaussian-mixture innovations

The only change we make from Life-Cycle Model 9 is that instead of z being a AR(1) with iid normal innovations, it is an AR(1) with non-gaussian innovations. Specifically we model z as an AR(1) with 'gaussian mixture shocks'. You can approximate real-valued functions using polynomials, similarly you can approximate probability distributions with a 'mixture' of normal distributions. So an AR(1) with gaussian mixture shocks is a way of modeling an AR(1) with complicated non-gaussian shocks, which are approximated with gaussian mixture shocks.

[53] There are two general numerical methods to solve integrals (which we need to calculate expectations) using computers. The one we use is called 'quadrature', which involves evaluating the integral at a number of points (in our case the integrals are stochastic, so it is a number of grids points for the shock). The second is known as Monte Carlo integration and is never used for life-cycle models (you can find it being used for an economic model in Pál and Stachurski (2013), but quadrature is better for anything except high-dimensional shocks). There are essentially two aspects to discretizing AR(1) models to a markov process, the quadrature step of choosing grids, and a second step of choosing the transition matrix probabilities. Some quadrature methods treat these as seperate steps, like Farmer-Toda and Tauchen, while others treat them jointly, like Tauchen-Hussey and Rouwenhorst.

[54] Rouwenhorst (1995), Tauchen (1986) and Tauchen and Hussey (1991); see also Kopecky and Suen (2010) about the accuracy of Rouwenhorst being better for highly-persistent AR(1) processes.

[55] The basis for this recommendation are the results reported in Farmer and Toda (2017).

We will use the method of Farmer and Toda (2017) to discretize it as a markov process. The rest of the model is unchanged so we do not report the whole model here. Instead we just report the equations for the AR(1) with gaussian mixture shocks.

A general AR(1) with non-gaussian shocks is given by,

$$x_t = (1-\rho)c + \rho x_{t-1} + \epsilon_t, \quad \epsilon_t \sim F$$

where the innovations ϵ_t are i.i.d. and their distribution is given by F.

An AR(1) with 'gaussian mixture shocks' is exactly the same just with the addition that F is a gaussian mixture. We can have a gaussian mixture of n different normal distributions, but we will start with an example with a mixture of two normal distributions to get the idea. We have two normal distributions, $N(\mu_1, \sigma_1^2)$, and $N(\mu_2, \sigma_2^2)$, a guassian mixture has probability p_1 of being drawn from the first of these, and probability p_2 of being drawn from the second (obviously $p_1 + p_2 = 1$, so that the total probability is one). So we can write a guassian mixture of two normal (gaussian) distributions as, $F = p_1 N(\mu_1, \sigma_1^2) + p_2 N(\mu_2, \sigma_2^2)$. A general gaussian mixture of n normal distributions can be written as

$$F = \sum_{i=1}^{n} p_i N(\mu_i, \sigma_i^2)$$

You can estimate a gaussian mixture by standard methods like maximum likelihood or moment matching.

Notice that in the code to implement this we are going to have to provide the vector of $\{p_i\}_{i=1}^n$, the vector of $\{\mu_i\}_{i=1}^n$ and the vector of $\{\sigma_i\}_{i=1}^n$. The way the code works it will also require the 'unconditional mean', which is c in the above formula.[56]

We model F as a gaussian mixture. A gaussian mixture is flexible, so it can model a complicated F, yet analytically tractable, as all moments and the moment generating function have closed-form expressions. To give some intuition on why gaussian mixtures are useful, try to visualize the following $0.5N(2, 0.5) + 0.5N(0, 0.02)$, it will have a 'usual' normal distribution around 2, but also a sharp spike around 0, giving us a distribution with two peaks. To understand why they are convenient computationally the key is that the normal distributions is completely defined by just it's first two moments, the mean and variance; all the higher moments are just combinations of these. So when we add normal distributions together in a gaussian mixture everything just becomes weighted sums of these first two moments of each of the normal distributions we are mixing together.

We use the Farmer-Toda method to discretize the AR(1) with gaussian mixture shocks. The Rouwenhorst and Tauchen-Hussey methods cannot possibly discretize this process as they are designed around just the first two moments. It is possible to extend the Tauchen method to AR(1) with gaussian mixture shocks, see Civales, Diez-Catalan, and Fazilet (2017).

[56] The code can actually discretize an AR(p) with gaussian mixture shocks, $(x_t - c) = \rho_1(x_{t-1} - c) + ... + \rho_p(x_{t-p} - c) + \epsilon_t, \quad \epsilon_t \sim F$, and F is a gaussian mixture, $F = \sum_{i=1}^{n} p_i N(\mu_i, \sigma_i^2)$. It was originally written by Farmer and Toda (2017), please cite them if you use it.

A.3 Life-Cycle Model A3: Permanent (unit-root/random-walk) shocks

The only change we make from Life-Cycle Model 9 is that instead of z being a AR(1) with iid normal innovations, we instead have z follow a random-walk. We won't rewrite the value function problem itself, we just need to change the AR(1) process, $log(zprime) = \rho_z log(z) + \epsilon$, $\epsilon \sim N(0, \sigma^2_{\epsilon,z})$, to instead be a (bounded) random walk process, $zprime = z + \epsilon_z$. Modelling permanent states typically requires more states for z to remain accurate and we use 101, specifically $z\_grid = linspace(0.2, 2, 101)'$ (101 points equally spaced from 0.2 to 2, inclusive). We just need to set the transition matrix π_z to be a (bounded) random walk. We will choose ϵ_z to be have a probability of 0.5 of staying at the current value of z, a probability of 0.25 of going one grid point lower, and a probability of 0.25 of going one grid point higher. At the minimum grid point we will have a probability of 0.5 of staying, and a probability of 0.5 of going one point higher. At the maximum grid point we will have a probability of 0.5 of staying, and a probability of 0.5 of goind one point lower. Notice that this is a random walk, as $E[zprime|z] = z$, everywhere except at the maximum and minimum values, hence it acts as a random walk, except when it runs into the bounds.

The only things we need to change in the code are n_z, z_grid, and pi_z.

A.3.1 Special case: Permanent shocks with exogenous labor

There is a very nice way to implement permanent random walk shocks in models with exogenous labor that means way fewer grid points are required than if we just took the same naive approach as we did to add permanent shocks with endogenous labor in Life-Cycle Model A3. The key is in how the permanent shock enters the budget constraint. We can use a functional form for the shocks in the budget constraint that allows us to simply renormalize the model endogenous states in terms of the innovations in the permanent shock variable. We therefore only need a grid on z of the innovations, rather than the actual values of the permanent shock itself; e.g., for $y_t = y_{t-1} + \epsilon_t$ we would just need a grid on innovations ϵ, rather than a grid on y, and so we can get much higher accuracy from any given grid size.

Implementing this requires a substantial change to the model (no pensions, no labor productivity units as a deterministic function of age, etc.) so we will not attempt to explain the model here, you can find the full model both as a readable code and implementing codes at: discourse.vfitoolkit.com/t/permanent-shocks-to-income/127

The method and model are explained in full detail in Carroll (2021).

Note, the way we have written the code it would be trivial to make the innovations ϵ_t follow an AR(1) process.

Note, this also shows you how to implement shocks to the discount factor, which are standard in models of the zero-lower bound as the way to generate the zero-lower bound (you would need to wrap the life-cycle model in a general equilibrium setting to look at this).

A.4 Life-Cycle Model A4: Second-Order Markov Processes (implementing an AR(2) persistent shock)

The VFI Toolkit can only handle first-order Markov processes, which are typically just referred to as markov processes. But fortunately you can rewrite any second-order Markov process with one state as a first-order Markov process with two states (the original state and the lag of that state). Using second-order Markovs processes is rare in economic models, but let's see how.

Before explaing how to convert the second-order Markov processes into a first-order Markov process we explain the details of how to change a Markov process with two variables into a single (vector valued) variable Markov process. We have already done this in the Life-Cycle Models that contain two exogenous shocks (such as Life-Cycle Model 11), but it will help to see it formally.

How to turn a Markov process with two variables into a single variable Markov process: Say you have a Markov process with two variables, z & y, we can treat them as a single vector-valued markov process $x = (z, y)$. Let z take the states $z_1, ...z_n$, and y take the states $y_1, ..., y_m$. We start with the simple example of how to combine the two when their transitions are completely independent of each other so as to illustrate the concepts and then treat the general case.

When z and y are independent Markov processs, $Pr(z_{t+1} = z_j) = Pr(z_{t+1} = z_i|z_t = z_j) = \pi^z_{ij}\ \forall i, j = 1, ...n$ and $Pr(y_{t+1} = y_j) = Pr(y_{t+1} = y_i|y_t = y_j) = \pi^y_{ij}\ \forall i, j = 1, ...m$, then we can define a new single-variable Markov process x simply by taking the Kronecker Product of z and y. Thus, x will have n times m states, $[x_1, .., x_{nm}]' = [z_1, ...z_n]' \otimes [y_1, ...y_m]'$, and it's transition matrix will be the Kronecker Product of their transition matrices; $\Pi^x = \Pi^z \otimes \Pi^y$. For the definition of the Kronecker product of two matrices see wikipedia. Note that in the codes you only need to combine the transition matrices to create the joint transition matrix; you simply include the grids as a 'stacked' column vector and the rest is done internally by VFI Toolkit.

Note that we can let z and y depend on each other (as in the case that z and y form a bivariate VAR(1), treated in Life-Cycle Model A7), although obviously the transition matrix becomes more complex. Another case of potential interest is when one of z and y is 'microeconomic' and the other is 'macroeconomic', which we capture by allowing the microeconomic to depend on the macroeconomic, but not vice-versa. Thus we assume, without loss of generality, that z_t is the microeconomic variable and evolves according to the transformation matrix defined by $Pr(z_t|z_{t-1}, y_t, y_{t-1})$, while y_t has the transformation matrix $Pr(y_t) = Pr(y_t|y_{t-1})$[57]. The transition matrix however is now more complicated; rather than provide a general formula you are referred to the example in Life-Cycle Model A8.

If you have three or more variables in vector of the first-order Markov process you can reduce this to a scalar first-order Markov process simply by iteratively combining pairs. For example with three variables, (z^1, z^2, z^3), start by first defining x^1 as the combination of z^1 & z^2 (combining them as described above), and then x as the

[57] Note that here I switch from describing Markov processs as $t + 1|t$ to $t|t - 1$, the difference is purely cosmetic.

combination of x^1 and z^3.

How to turn a second-order Markov process into a first-order Markov process: Suppose that z, with states $z_1, ...z_n$ is a second-order Markov process, that is $Pr(z_{t+1} = z_i) = Pr(z_{t+1} = z_i | z_t = z_j, z_{t-1} = z_k) \neq Pr(z_{t+1} = z_i | z_t = z_j)$. Consider now the vector (z_{t+1}, z_t). It turns out that this vector is in fact a vector first-order Markov process (define this periods state (z_{t+1}, z_t) in terms of last periods state (z_t, z_{t-1}) by defining this periods z_{t+1} as a function of last periods z_t & z_{t-1} following the original second-order Markov process, and define this periods z_t as being last periods z_t). Now we have a 'vector-valued' first-order Markov process on the vector (z_t, z_{t-1}).

Obviously this idea can be trivially extended to turn, e.g., a third-order Markov process into a first-order Markov process with three states.

Life-Cycle Model A4: idiosyncratic shock, AR(2) We will provide an example that takes Life-Cycle Model 9 and makes the sole change of switching from an AR(1) process for the exogenous shock on (log) labor productivity units z to instead being an AR(2) process.

$$z_t = \rho_{z,1} z_{t-1} + \rho_{z,2} z_{t-2}$$

We can use the Farmer-Toda method to discretize this, the command we use is actually for a more general AR(p) process with gaussian mixture innovations, but we will just use an AR(2) with gaussian innovations (we will have to code it as a 'mixture of one normal').

The Farmer-Toda method creates a first-order Markov process on (z_t, z_{t-1}), which we will refer to in the codes as (z_1, z_2). Note that while both z_1 and z_2 are exogenous state variables the return function will only actually use the current value z_1 (which represents z_t); both have to be passed as inputs to the return function as they are both states, but only one ends up being used in the return function (the other sill matters for things like expectations).

A.5 Life-Cycle Model A5: Age-dependent shocks

Empirical evidence shows that the (age-conditional) variance of income increases with age. One way to model this is to have shocks that increase in varance as age increases. If the shocks were transitory the age-conditional variance of consumption would not increase, because households would smooth consumption. So we also want these shocks to be persistent (so they imply a change in permanent-income and thus the life-cycle consumption hypothesis means consumption will also shift). How to model persistent shocks with a variance that increases with age? We can model it as a 'non-stationary AR(1)' process, and discretize this using the 'generalized-Rouwenhorst' of Fella, Gallipoli, and Pan (2019). The model is exactly the same as that of Life-Cycle Model 9, except for changes to the exogenous shock process z on labor productivity units. So we will only describe the new z process here; see Life-Cycle Model 9 for the full model.

The 'non-stationary AR(1)' process for z is given by,

$$z_j = \rho_j z_{j-1} + \epsilon_j, \quad \epsilon_j \sim N(0, \sigma_{\epsilon,z,j})$$

notice the difference from a standard AR(1) because ρ_j and $\sigma_{\epsilon,z,j}$ both depend on age. We assume that $\rho_j < 1$, for all $j = 1, 2, ..., J$. Let, $\sigma_{z,j}$ be the standard deviation of z_j, then it follows that it evolves according to $\sigma^2_{z,j} = \rho^2_j \sigma^2_{z,j-1} + \sigma^2_{\epsilon,z,j}$. A full definition requires us to additionally specify z_0, which we do below.

Note that we enforce that the constant term is zero, so there is no 'drift' in the 'non-stationary AR(1)' process. In the original paper (Fella, Gallipoli, and Pan, 2019) implementing the discretization also require one further assumption, namely that $z_0 = 0$ (it has probability 1 of equaling zero). In the codes this is the default setting, but other initial distributions for z_0 can be implemented using the 'options' input.

The 'non-stationary AR(1)' process provides an alternative to the permanent shocks we saw in Life-Cycle Model A3. Which is better is an empirical question. Note that an extended Tauchen method also exists (it is in VFI Toolkit, but not used in any of these examples).

A.6 Life-Cycle Model A6: Age-dependent shocks: persistent and transitory

NOT YET IMPLEMENTED

A.7 Life-Cycle Model A7: VAR(1) persistent shocks

In Life-Cycle Model 10 we used two idiosyncratic shocks, z_1 and z_2, modelling one as an AR(1) and the other as i.i.d. Normal. It is possible to model two shocks as a VAR(1) and that is what we do here, discretizing them using the Tauchen method for VAR(1). Since nothing changes in the model except the process on z_1 and z_2 we will only describe that here.

Let $z = [z_1; z_2]$, then we can write the bivariate VAR(1) as

$$z_t = \mu + \rho_z z_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \Sigma)$$

where μ is a 2-by-1 vector, ρ_z is a 2-by-2 matrix, ϵ_t is a 2-by-1 vector, and Σ is a 2-by-2 variance-covariance matrix.

We could write the same bivariate VAR(1) as the system of equations,

$$\begin{aligned} z_{1,t} &= \mu_1 + \rho_{z,1,1} z_{1,t-1} + \rho_{z,1,2} z_{2,t-1} + \epsilon_{1,t} \\ z_{2,t} &= \mu_2 + \rho_{z,2,1} z_{1,t-1} + \rho_{z,2,2} z_{2,t-1} + \epsilon_{2,t} \\ [\epsilon_{1,t}, \epsilon_{2,t}]' &\sim N(0, \Sigma) \end{aligned}$$

there relationship between the two being $\mu = [\mu_1; \mu_2]$ and $\rho_z = [\rho_{z,1,1}, \rho_{z,1,2}; \rho_{z,2,1}, \rho_{z,2,2}]$.

When using (our implementation of) the Tauchen method to discretize this VAR(1) we must also impose that Σ is diagonal, $\Sigma = [\sigma^2_{\epsilon,z,1},\ 0;\ 0,\ \sigma^2_{\epsilon,z,2}]$.

VAR(1) with three or more variables can be implemented in the obvious manner. For more details see the discretization codes themselves.

In addition to the Tauchen method for discretizing VAR(1) there are also codes that implement the Farmer-Toda method for discretizing VAR(1), but these create 'jointly dependent' grids on z_1 and z_2 so cannot currently be used by VFI Toolkit.[58]

A.8 Life-Cycle Model A8: Shocks that depend on each other: 'recession' and 'unemployment'

We will have two shocks each of which can take two values: z_1 which is employment/unemployment, just as in Life-Cycle Model 8, and z_2 which is recession/expansion. The idea is that agents care about the microeconomics shocks (which directly effect them), but they do not directly care about the macroeconomic shocks (that only effect them indirectly). The macroeconomic shocks influence the microeconomic shocks (a recession makes unemployment more likely) which we model as the transition probabilities of the microeconomic shocks depending on the state of the macroeconomic shock.

So we will have two shocks, the macroeconomic shock z_2 will have 'its own' transition matrix, and then the microeconomic shock z_1 will have a transition probability that depends on z_2, which we will define using the joint transition probability matrix on $z = (z_1, z_2)$.

This modelling technique originates with Imrohoroglu (1989) - "Cost of business cycles with indivisibilities and liquidity constraints" You can use "rational inattention" to provide refinement of this concept so that households do not react to the macroeconomic variables. See, Maćkowiak and Wiederholt (2015) - "Business Cycle Dynamics under Rational Inattention." Or as life-cycle model in Carroll, Crawley, Slacalek, Tokuoka, and White (2020) - Sticky Expectations and Consumption Dynamics" (sticky expectations is a reduced-form way to model the concept of rational inattention)

This is essentially an extension of life-cycle model 8, adding macroeconomic shock which will influence the transition of the microeconomic unemployment shock, but is irrelevant to the household problem directly, it only matters indirectly because of its influence on expectations of what the microeconomic shock will transition to in the future.

[58] We have been using a grid for z_1 and a grid for z_2, and the get a grid on $z = [z_1, z_2]$ by taking the kronecker product (cross product) of the two grids (the obvious way to create a grid on the two dimensions from two single dimension grids; this is all done internally by the codes so you won't see it). Farmer-Toda create a jointly-dependent grid on z, which is not the kronecker product of two seperate grids. Jointly-dependent grids are better, in principle, but require slightly different code to handle them and VFI Toolkit does not currently handle this.

The household value fn problem is,

$$\begin{aligned}
V(a,z1,z2,j) \quad &= \quad \max_{c,aprime,h} \frac{c^{1-\sigma}}{1-\sigma} - \psi\frac{h^{1+\eta}}{1+\eta} + (1-s_j)\beta\mathbb{I}_{(j>=Jr+10)}warmglow(aprime) \\
&\qquad +s_j\beta E[V(aprime,z1prime,z2prime,j+1)|z1,z2] \\
&\quad \text{if } j < Jr: \; c + aprime = (1+r)a + w\kappa_j z1h \\
&\quad \text{if } j >= Jr: \; c + aprime = (1+r)a + pension \\
&\quad 0 \leq h \leq 1, aprime \geq 0 \\
&\quad zprime = \pi_z(z), \text{ is a four-state markov}
\end{aligned}$$

which is similar to Life-Cycle Model 8, except for z_2 as a state that influences expecations. Note that z has four states (2 for z_1 times 2 for z_2).

We have to make a decision about peoples state in terms of the exogenous shock at birth, and we will arbitratily have them be employed-recession.

A.9 Life-Cycle Model A9: Jointly-determined Grids for Correlated Shocks

When we have two shocks that are correlated it makes sense to use a jointly-determined grid, as opposed to a kronecker grid (a.k.a. cross-product grid) as we have used until now. In a kronecker grid we choose, e.g., five points for $z1$ and five points for $z2$, and our grid on $(z1,z2)$ will be the cross-product (kronecker product) of these, that is the 25 points we get from all combinations of the 5 points on $z1$ with the 5 points on $z2$. But when the shocks are correlated this may not make much sense as a grid. Imagine that the two shocks have a strong positive correlation. Then it is a waste to have a point on the grid which is a small value for $z1$ and a large value for $z2$, as this will essentially never occur; a kronecker grid when the shocks are highly correlated will have many points with essentially zero weight. We can avoid this problem by using a 'jointly-determined grid', which in our present example would involve 25 points on $(z1,z2)$ chosen to reflect the correlation.

We will resolve Life-Cycle Model 11 (specifically 11A), with the only change being that the two shocks are now both Markov and correlated. Since the only change is to the shocks themselves we will not repeat the model itself here. The two shocks to idiosyncratic labor productivity units, $z1$ and $z2$, are assumed to follow a VAR(1),

$$\begin{bmatrix} z1_t \\ z2_t \end{bmatrix} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.7 \end{bmatrix} \begin{bmatrix} z1_{t-1} \\ z2_{t-1} \end{bmatrix} + \epsilon_t \text{where } \epsilon_t \sim N\left(0, \begin{bmatrix} 0.0303 & 0.0027 \\ 0.0027 & 0.0382 \end{bmatrix}\right)$$

Note that this process does not contain large correlation between $z1$ and $z2$ so the advantages of a jointly-determined grid rather than a kronecker grid will be minor (the higher the correlation the bigger the advantages as the more points would be 'wasted' by the kronecker grid). We discretize this process using the Farmer-Toda method for VAR(1) processes which will always produce jointly-determined grids.

In terms of solving the model there is essentially no change. The shape of 'z_grid' is now product(n_z)-by-length(n_z) instead of the standard sum(n_z)-by-1 (recall that n_z is a row vector containing the number of grid points for each z variable, e.g., n_z=[5,5]). VFI Toolkit recognises this difference and automatically interprets it as a jointly-determined grid and acts appropriately.

Obviously there is nothing stopping you from using jointly-determined grids when shocks are not correlated, just that there is no advantage from doing so.