1 Solving the Next-to-Last Period

{sec:solving-the-nex

To reduce clutter, we now temporarily assume that $\mathcal{G}_t = 1$ for all t, so that the $\mathcal{G}$ terms from the earlier derivations disappear, and setting $t = T$ the problem in the second-to-last period of life can now be expressed as

$$\mathrm{v}_{(t-1)}(m) = \max_{\mathrm{c}} \quad \mathrm{u}(\mathrm{c}) + \mathrm{v}_{(t-1)\rightarrow}(\overbrace{m - \mathrm{c}}^{a}) \tag{1}$$ {eq:vEndTm1}

where

$$\begin{aligned} \mathrm{v}_{(t-1)\rightarrow}(a) &= \beta \mathrm{v}_{\leftarrow t}(a) \\ &= \beta \mathbb{E}_{\leftarrow t} \left[\mathrm{v}_t(\underbrace{a\mathcal{R}_t + \boldsymbol{\theta}_t}_{m_t}) \right] \end{aligned}$$

Using (0) $t = T$; (1) $\mathrm{v}_t(m) = \mathrm{u}(m)$; (2) the definition of $\mathrm{u}(m)$; and (3) the definition of the expectations operator,

$$\mathrm{v}_{\leftarrow t}(a) = \int_0^\infty \frac{(a\mathcal{R}_t + \vartheta)^{1-\rho}}{1-\rho} d\mathcal{F}(\vartheta) \tag{2}$$ {eq:NumDefInt}

where $\mathcal{F}(\boldsymbol{\theta})$ is the cumulative distribution function for $\boldsymbol{\theta}$.

```
# A 'raw' solution to the T-1 solution works, but is very slow:
```

This maximization problem implicitly defines a 'local function' $\mathrm{c}_{t-1}(m)$ that yields optimal consumption in period $t-1$ for any specific numerical level of resources like $m = 1.7$.

But because there is no general analytical solution to this problem, for any given m we must use numerical computational tools to find the c that maximizes the expression. This is excruciatingly slow because for every potential c to be considered, a definite integral over the interval $(0, \infty)$ must be calculated numerically, and numerical integration is *very* slow (especially over an unbounded domain!).

1.1 Discretizing the Distribution

Our first speedup trick is therefore to construct a discrete approximation to the lognormal distribution that can be used in place of numerical integration. That is, we want to approximate the expectation over $\boldsymbol{\theta}$ of a function $g(\boldsymbol{\theta})$ by calculating its value at set of $n_{\boldsymbol{\theta}}$ points $\boldsymbol{\theta}_i$, each of which has an associated probability weight w_i:

$$\begin{aligned} \mathbb{E}[g(\boldsymbol{\theta})] &= \int_{\underline{\boldsymbol{\theta}}}^{\bar{\boldsymbol{\theta}}} (\vartheta) d\mathcal{F}(\vartheta) \\ &\approx \sum_{\boldsymbol{\theta}=1}^{n} w_i g(\boldsymbol{\theta}_i) \end{aligned}$$

(because adding n weighted values to each other is enormously faster than general-purpose numerical integration).

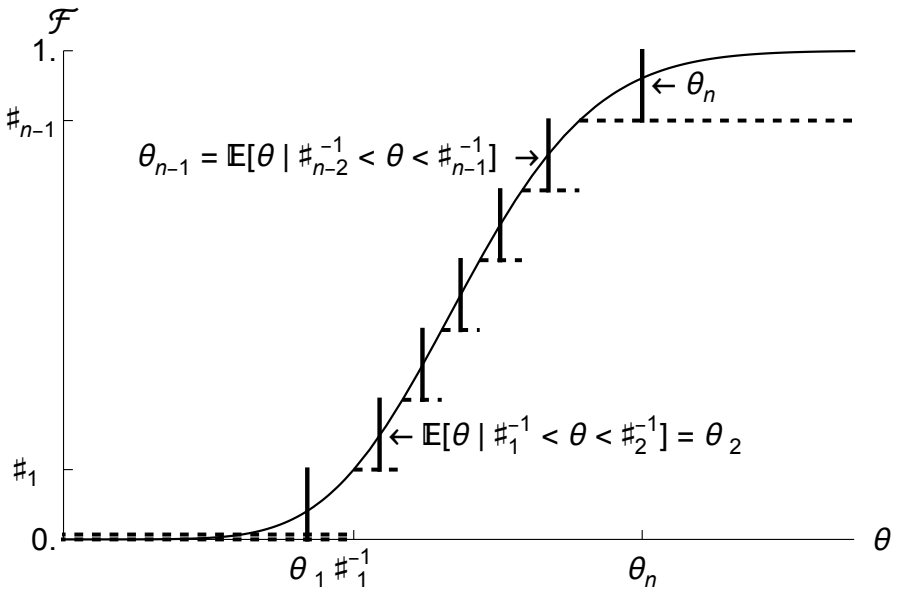

Figure 1 Equiprobable Discrete Approximation to Lognormal Distribution $\mathcal{F}$

{fig:discreteapprox}

Such a procedure is called a 'quadrature' method of integration; Tanaka and Toda (2013) survey a number of options, but for our purposes we choose the one which is easiest to understand: An 'equiprobable' approximation (that is, one where each of the values of $\boldsymbol{\theta}_i$ has an equal probability, equal to $1/n_{\boldsymbol{\theta}}$).

We calculate such an n-point approximation as follows.

Define a set of points from $\sharp_0$ to $\sharp_{n_{\boldsymbol{\theta}}}$ on the $[0,1]$ interval as the elements of the set $\sharp = \{0, 1/n, 2/n, \ldots, 1\}$.[1] Call the inverse of the $\boldsymbol{\theta}$ distribution $\mathcal{F}^{-1}$, and define the points $\sharp_i^{-1} = \mathcal{F}^{-1}(\sharp_i)$. Then the conditional mean of $\boldsymbol{\theta}$ in each of the intervals numbered 1 to n is:

$$\boldsymbol{\theta}_i \equiv \mathbb{E}[\boldsymbol{\theta} | \sharp_{i-1}^{-1} \leq \boldsymbol{\theta} < \sharp_i^{-1}] = \int_{\sharp_{i-1}^{-1}}^{\sharp_i^{-1}} \vartheta \, d\mathcal{F}\,(\vartheta), \tag{3}$$

and when the integral is evaluated numerically for each i the result is a set of values of $\boldsymbol{\theta}$ that correspond to the mean value in each of the n intervals.

The method is illustrated in Figure 1. The solid continuous curve represents the "true" CDF $\mathcal{F}(\boldsymbol{\theta})$ for a lognormal distribution such that $\mathbb{E}[\boldsymbol{\theta}] = 1$, $\sigma_{\boldsymbol{\theta}} = 0.1$. The short vertical line segments represent the $n_{\boldsymbol{\theta}}$ equiprobable values of $\boldsymbol{\theta}_i$ which are used to approximate this distribution.[2]

Because one of the purposes of these notes is to connect the math to the code that solves the math, we display here a brief snippet from the notebook that constructs these points.

[1] These points define intervals that constitute a partition of the domain of $\mathcal{F}$.

[2] More sophisticated approximation methods exist (e.g. Gauss-Hermite quadrature; see Kopecky and Suen (2010) for a discussion of other alternatives), but the method described here is easy to understand, quick to calculate, and has additional advantages briefly described in the discussion of simulation below.

```
# This is a snippet of code that constructs mass points
# for the equiprobable representation of the problem
```

We now substitute our approximation (4) for $\mathrm{v}_{(t-1)_{\rightarrow}}(a)$ in (1) which is simply the sum of $n_{\boldsymbol{\theta}}$ numbers and is therefore easy to calculate (compared to the full-fledged numerical integration (2) that it replaces).

$$\mathrm{v}_{(t-1)_{\rightarrow}}(a) = \beta \left(\frac{1}{n_{\boldsymbol{\theta}}}\right) \sum_{i=1}^{n_{\boldsymbol{\theta}}} \frac{(\mathcal{R}_t a + \boldsymbol{\theta}_i)^{1-\rho}}{1-\rho} \tag{4}$$ {eq:vDiscrete}

```
# The code that corresponds to evaluation of the discretized max
    problem is
```

1.2 The Approximate Consumption and Value Functions

Given any particular value of m, a numerical maximization tool can now find the c that solves (1) in a reasonable amount of time.

The notebook code responsible for computing an estimated consumption function begins in "Solving the Model by Value Function Maximization," where a vector containing a set of possible values of market resources m is created (in the code, various m vectors have names beginning `mVec`; in these notes we will use boldface italics to represent vectors, so we can refer to our collection of m points as $\boldsymbol{m}$ with values indexed by brackets: $\boldsymbol{m}[1]$ is the first entry in the vector, up to a last entry $\boldsymbol{m}[-1]$; we arbitrarily (and suboptimally) pick the first five integers as our five `mVec` gridpoints (in the code, `mVec_int`$= \{0., 1., 2., 3., 4.\}$)).

1.3 An Interpolated Consumption Function

{subsec:LinInterp}

We can now apply our solution to (1) to each of the values in $\boldsymbol{m}$, generating a corresponding optimal $\boldsymbol{c}$. This is called 'sampling' the consumption function. Using the ordered pairs $\{\boldsymbol{m}, \boldsymbol{c}\}$ we can create a piecewise linear 'interpolating function' (a 'spline') which when applied to any input $\boldsymbol{m}[1] \leq m \leq \boldsymbol{m}[-1]$ will yield the value of c that corresponds to a linear 'connect-the-dots' interpolation of the value of c from the values of the two nearest computed $\{m, c\}$ points.[3]

This is accomplished in "An Interpolated Consumption Function," which generates an interpolating function that we designate $\grave{\mathrm{c}}_{t-1}(m)$.

Figures 2 and 3 show plots of the constructed $\grave{\mathrm{c}}_{t-1}$ and $\grave{\mathrm{v}}_{t-1}$. While the $\grave{\mathrm{c}}_{t-1}$ function looks very smooth, the fact that the $\grave{\mathrm{v}}_{t-1}$ function is a set of line segments is very evident. This figure provides the beginning of the intuition for why trying to approximate the value function directly is a bad idea (in this context).[4]

[3] For a useful treatment of various kinds of interpolation appropriate for different questions, see

[4] For some problems, especially ones with discrete choices, value function approximation is unavoidable; nevertheless, even in such problems, the techniques sketched below can be very useful across much of the range over which the problem is defined.

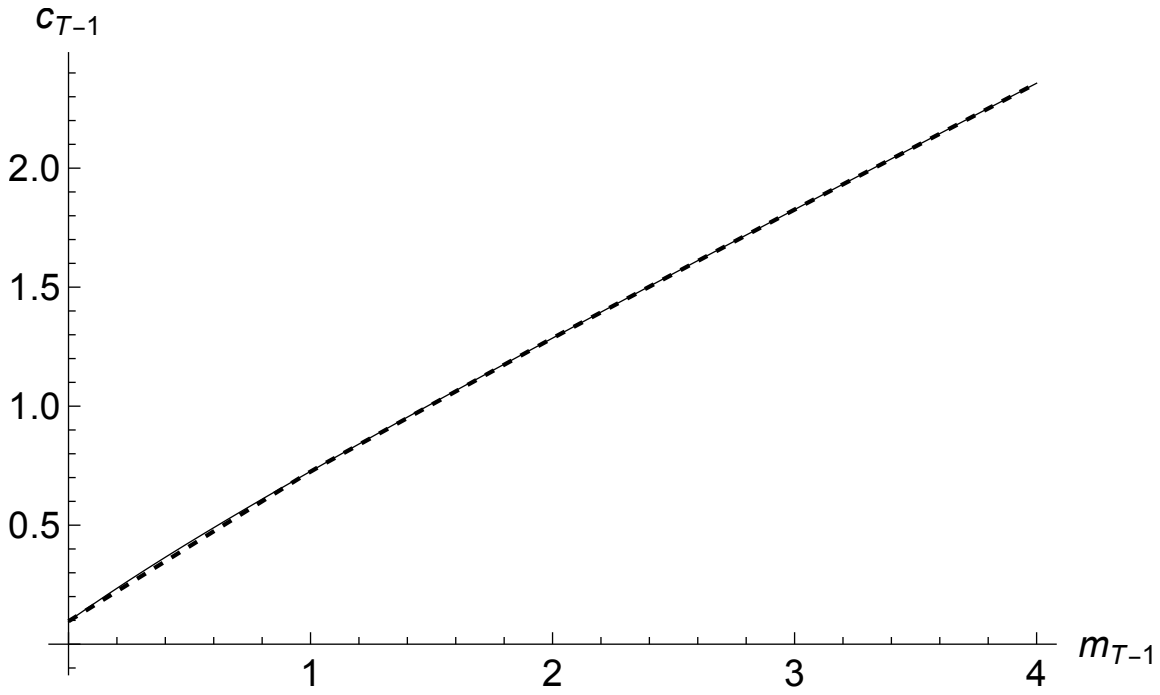

Figure 2 $\mathrm{c}_{T-1}(m)$ (solid) versus $\grave{\mathrm{c}}_{T-1}(m)$ (dashed)

{fig:PlotcTm1Simpl

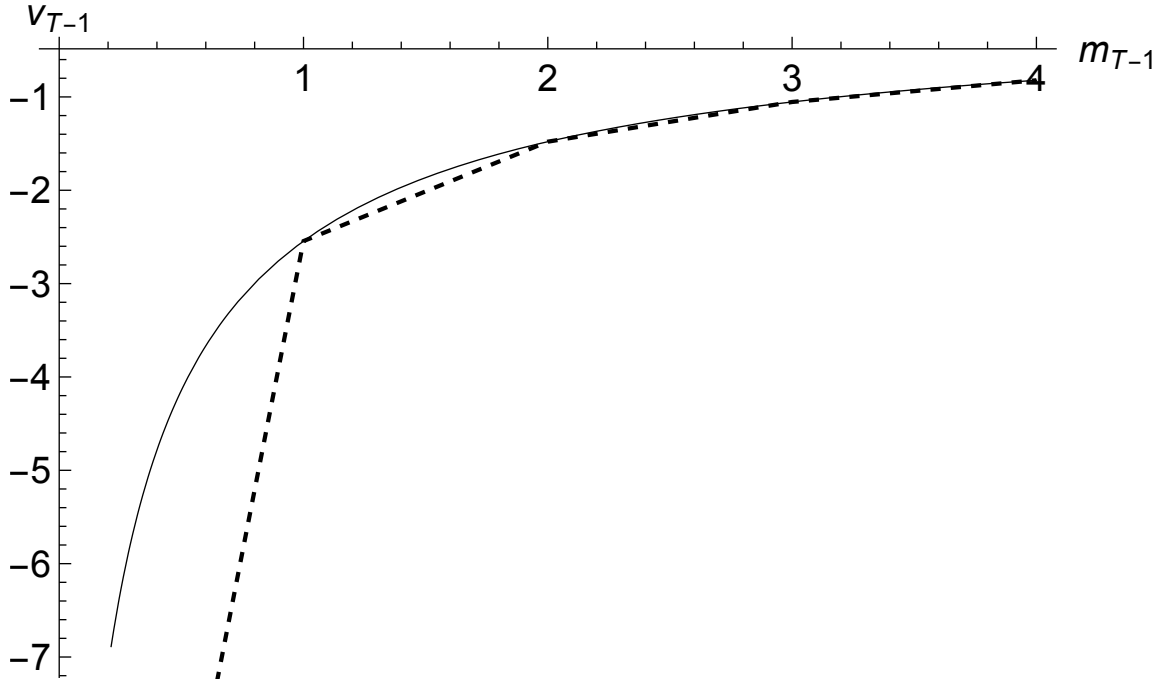

Figure 3 v_{T-1} (solid) versus $\grave{\mathrm{v}}_{T-1}(m)$ (dashed)

{fig:PlotVTm1Simp

1.4 Interpolating Expectations

Piecewise linear 'spline' interpolation as described above works well for generating a good approximation to the true optimal consumption function. However, there is a clear inefficiency in the program: Since it uses equation (1), for every value of m the program must calculate the utility consequences of various possible choices of c (and therefore a_{t-1}) as it searches for the best choice.

For any given index j in $\boldsymbol{m}[j]$, the algorithm, as it searches for the corresponding optimal a, the algorithm will end up calculating $\mathrm{v}_{(t-1)_{\rightarrow}}(\tilde{a})$ for many $\tilde{a}$ values close to the optimal a_{t-1}. Indeed, even when searching for the optimal a for a *different* m (say $\boldsymbol{m}[k]$ for $k \neq j$) the search process might compute $\mathrm{v}_{(t-1)_{\rightarrow}}(a)$ for an a close to the correct optimal a for $\boldsymbol{m}[j]$. But if that difficult computation does not correspond to the exact solution to the $\boldsymbol{m}[k]$ problem, it is discarded.

To avoid solving the problem independently over and over again for multitudes of values of a that are close to each other, we can employ the same interpolation technique used above to construct a direct numerical approximation to the value function: Define a vector of possible values for end-of-period assets at time $t-1$, $\boldsymbol{a}$ (`aVec` in the code). Next, construct $\boldsymbol{v} = \mathrm{v}_{t-1}(\boldsymbol{a})$ using equation (4); then construct an approximation $\grave{\mathrm{v}}_{(t-1)_{\rightarrow}}(a)$ by passing the vectors $\boldsymbol{a}$ and $\boldsymbol{v}$ as arguments to a piecewise-linear interpolator (e.g., the one in `scipy.interpolate`).

The notebook section "Interpolating Expectations," now interpolates the expected value of *ending* the period with a given amount of assets.[5]

Figure 4 compares the true value function to the approximation produced by following the interpolation procedure; the approximated and exact functions are of course identical at the gridpoints of $\boldsymbol{a}$ and they appear reasonably close except in the region below $m = 1$.

In all figs, replace gothic h with notation corresponding to the lecture notes.

Nevertheless, the consumption rule obtained when the approximating $\grave{\mathrm{v}}_{(t-1)_{\rightarrow}}(a_{t-1})$ is used instead of $\mathrm{v}_{(t-1)_{\rightarrow}}(a_{t-1})$ is surprisingly bad, as shown in figure 5. For example, when m goes from 2 to 3, $\grave{\mathrm{c}}_{t-1}$ goes from about 1 to about 2, yet when m goes from 3 to 4, $\grave{\mathrm{c}}$ goes from about 2 to about 2.05. The function fails even to be concave, which is distressing because Carroll and Kimball (1996) prove that the correct consumption function is strictly concave in a wide class of problems that includes this one.

1.5 Value Function versus First Order Condition

{subsec:vVsuP}

Loosely speaking, our difficulty reflects the fact that the consumption choice is governed by the *marginal* value function, not by the *level* of the value function (which is the object that we approximated). To understand this point, recall that a quadratic utility function exhibits risk aversion because with a stochastic c,

$$\mathbb{E}[-(c - \not{c})^2] < -(\mathbb{E}[c] - \not{c})^2 \tag{5}$$

(where $\not{c}$ is the 'bliss point' which is assumed always to exceed feasible c). However, unlike the CRRA utility function, with quadratic utility the consumption/saving *behavior* of

[5] What we are doing here is closely related to 'the method of parameterized expectations' of den Haan and Marcet (1990); the only difference is that our method is essentially a nonparametric version.

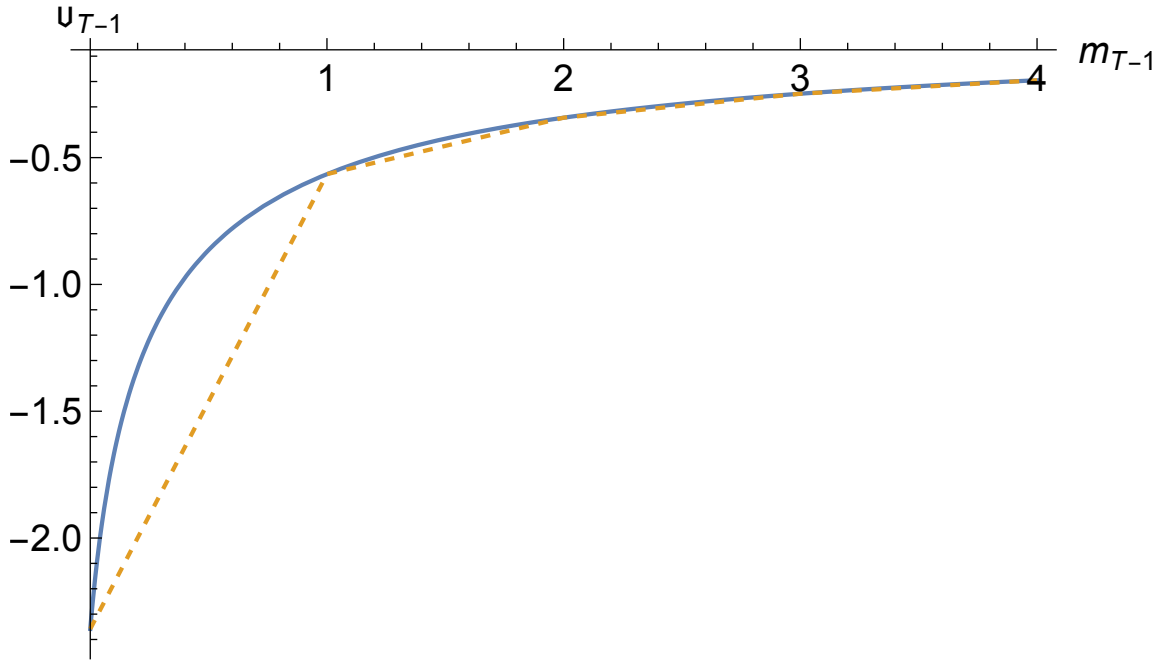

Figure 4 End-Of-Period Value $\mathrm{v}_{(t-1)_{\rightarrow}}(a_{t-1})$ (solid) versus $\grave{\mathrm{v}}_{(T-1)_{\rightarrow}}(a_{T-1})$ (dashed)

{fig:PlotOTm1Raw

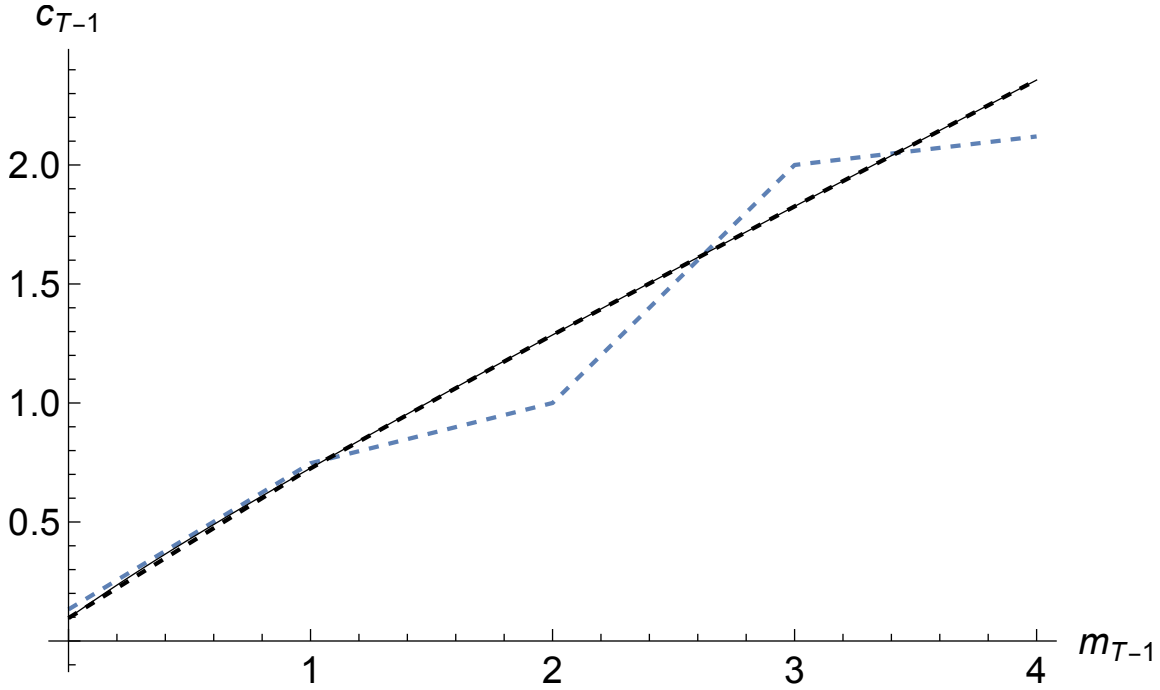

Figure 5 $\mathrm{c}_{T-1}(m)$ (solid) versus $\grave{\mathrm{c}}_{T-1}(m)$ (dashed)

{fig:PlotComparecT

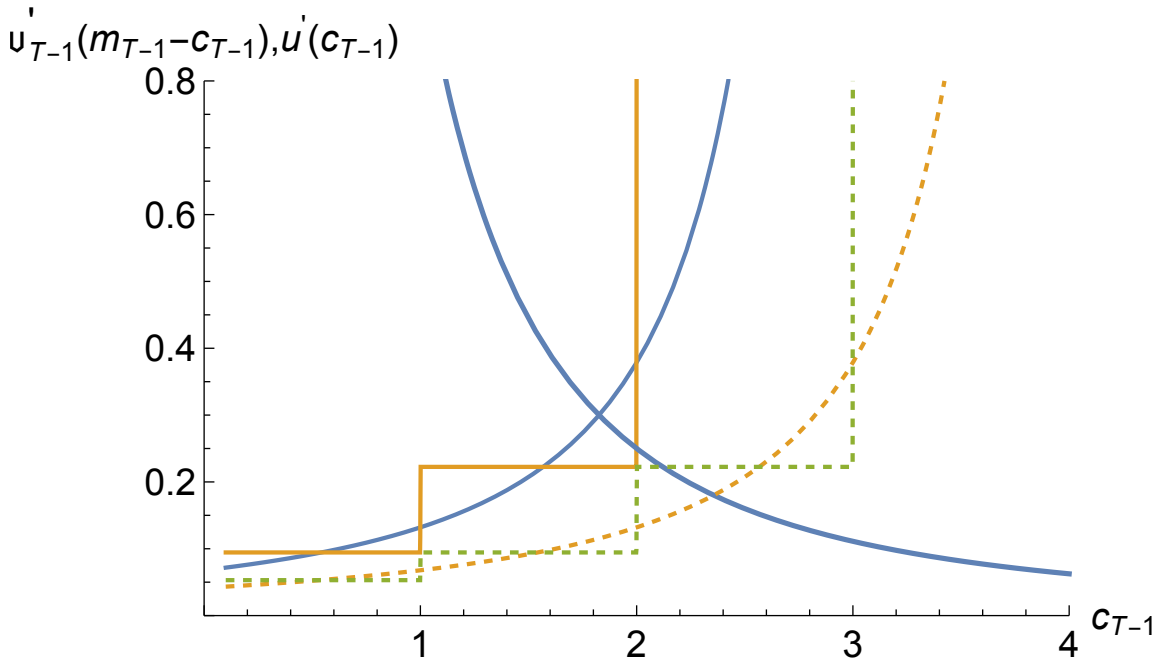

Figure 6 $\mathrm{u}^c(c)$ versus $\mathrm{v}^a_{(T-1)\rightarrow}(3-c), \mathrm{v}^a_{(T-1)\rightarrow}(4-c), \grave{\mathrm{v}}^a_{(T-1)\rightarrow}(3-c), \grave{\mathrm{v}}^a_{(T-1)\rightarrow}(4-c)$ {fig:PlotuPrimeVSC

consumers is unaffected by risk since behavior is determined by the first order condition, which depends on *marginal* utility, and when utility is quadratic, marginal utility is unaffected by risk:

$$\mathbb{E}[-2(c-\not{c})] = -2(\mathbb{E}[c]-\not{c}). \tag{6}$$

Intuitively, if one's goal is to accurately capture choices that are governed by marginal value, numerical techniques that approximate the *marginal* value function will yield a more accurate approximation to optimal behavior than techniques that approximate the *level* of the value function.

The first order condition of the maximization problem in period $T-1$ is:

$$\begin{aligned} \mathrm{u}^c(\mathrm{c}) &= \beta \mathbb{E}_{\rightarrow(T-1)}[\mathsf{R}\mathrm{u}^c(c_t)] \\ \mathrm{c}^{-\rho} &= \mathsf{R}\beta \left(\frac{1}{n_{\boldsymbol{\theta}}}\right) \sum_{i=1}^{n_{\boldsymbol{\theta}}} (\mathsf{R}(m-\mathrm{c}) + \boldsymbol{\theta}_i)^{-\rho} . \end{aligned} \tag{7}$$ {eq:FOCTm1}

In the notebook, the "Value Function versus the First Order Condition" section completes the task of finding the values of consumption which satisfy the first order condition in (7) using the `brentq` function from the `scipy` package.

The downward-sloping curve in Figure 6 shows the value of $\mathrm{c}^{-\rho}$ for our baseline parameter values for $0 \leq \mathrm{c} \leq 4$ (the horizontal axis). The solid upward-sloping curve shows the value of the RHS of (7) as a function of c under the assumption that $m = 3$. Constructing this figure is time-consuming, because for every value of c plotted we

must calculate the RHS of (7). The value of c for which the RHS and LHS of (7) are equal is the optimal level of consumption given that $m = 3$, so the intersection of the downward-sloping and the upward-sloping curves gives the (approximated) optimal value of c. As we can see, the two curves intersect just below c = 2. Similarly, the upward-sloping dashed curve shows the expected value of the RHS of (7) under the assumption that $m = 4$, and the intersection of this curve with $\mathrm{u}^c(\mathrm{c})$ yields the optimal level of consumption if $m = 4$. These two curves intersect slightly below c = 2.5. Thus, increasing m from 3 to 4 increases optimal consumption by about 0.5.

Now consider the derivative of our function $\grave{\mathrm{v}}_{(t-1)}(a_{t-1})$. Because we have constructed $\grave{\mathrm{v}}_{(t-1)}$ as a linear interpolation, the slope of $\grave{\mathrm{v}}_{(t-1)}(a_{t-1})$ between any two adjacent points $\{\boldsymbol{a}[i],,\boldsymbol{a}[i+1]\}$ is constant. The level of the slope immediately below any particular gridpoint is different, of course, from the slope above that gridpoint, a fact which implies that the derivative of $\grave{\mathrm{v}}_{(t-1)_\rightarrow}(a_{t-1})$ follows a step function.

The solid-line step function in Figure 6 depicts the actual value of $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(3-\mathrm{c})$. When we attempt to find optimal values of c given m using $\grave{\mathrm{v}}_{(t-1)_\rightarrow}(a_{t-1})$, the numerical optimization routine will return the c for which $\mathrm{u}^c(\mathrm{c}) = \grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(m-\mathrm{c})$. Thus, for $m = 3$ the program will return the value of c for which the downward-sloping $\mathrm{u}^c(\mathrm{c})$ curve intersects with the $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(3-\mathrm{c})$; as the diagram shows, this value is exactly equal to 2. Similarly, if we ask the routine to find the optimal c for $m = 4$, it finds the point of intersection of $\mathrm{u}^c(\mathrm{c})$ with $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(4-\mathrm{c})$; and as the diagram shows, this intersection is only slightly above 2. Hence, this figure illustrates why the numerical consumption function plotted earlier returned values very close to c = 2 for both $m = 3$ and $m = 4$.

We would obviously obtain much better estimates of the point of intersection between $\mathrm{u}^c(\mathrm{c})$ and $\mathrm{v}^a_{(t-1)_\rightarrow}(m-\mathrm{c})$ if our estimate of $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}$ were not a step function. In fact, we already know how to construct linear interpolations to functions, so the obvious next step is to construct a linear interpolating approximation to the *expected marginal value of end-of-period assets function* at the points in $\boldsymbol{a}$:

$$\mathrm{v}^a_{(t-1)_\rightarrow}(\boldsymbol{a}) = \beta \mathsf{R}\left(\frac{1}{n_{\boldsymbol{\theta}}}\right)\sum_{i=1}^{n_{\boldsymbol{\theta}}}(\mathcal{R}_t\boldsymbol{a}+\boldsymbol{\theta}_i)^{-\rho} \tag{8}$$ {eq:vEndPrimeTm1}

yielding $\boldsymbol{v}^a_{(t-1)_\rightarrow}$ (the vector of expected end-of-period-$(T-1)$ marginal values of assets corresponding to `aVec`), and construct $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(a_{t-1})$ as the linear interpolating function that fits this set of points.

The results are shown in Figure 7. The linear interpolating approximation looks roughly as good (or bad) for the *marginal* value function as it was for the level of the value function. However, Figure 8 shows that the new consumption function (long dashes) is a considerably better approximation of the true consumption function (solid) than was the consumption function obtained by approximating the level of the value function (short dashes).

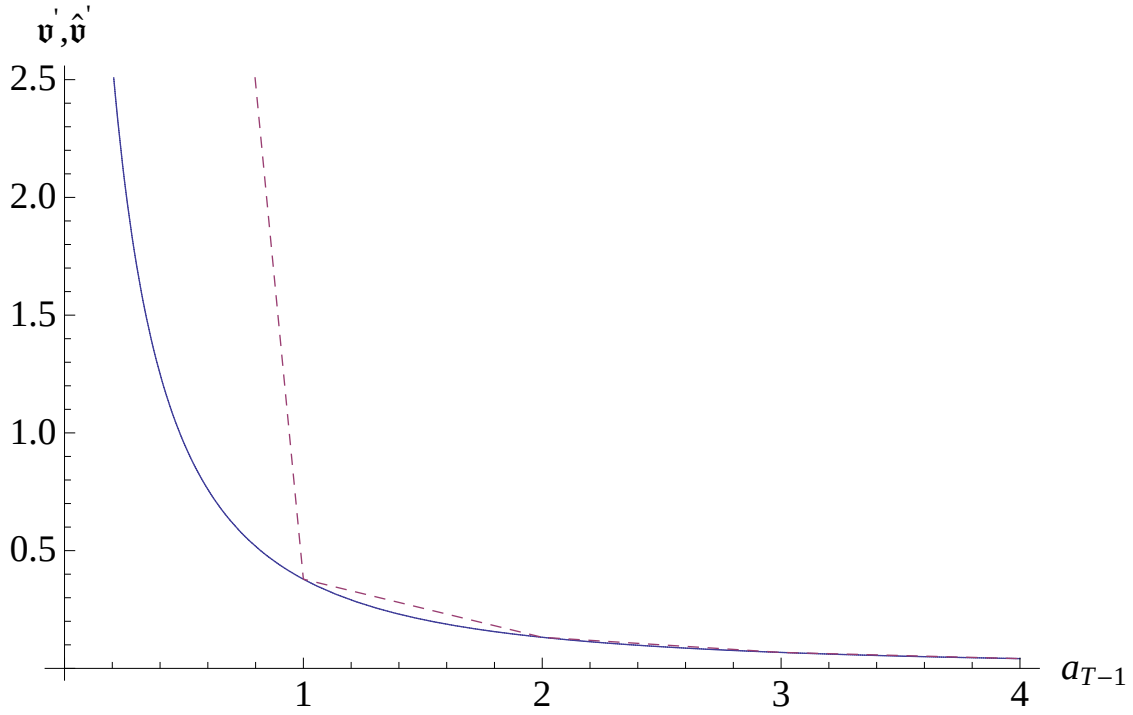

Figure 7 $\mathrm{v}^{a}_{(t-1)_{\rightarrow}}(a_{t-1})$ versus $\grave{\mathrm{v}}^{a}_{(t-1)_{\rightarrow}}(a_{t-1})$

{fig:PlotOPRawVSI

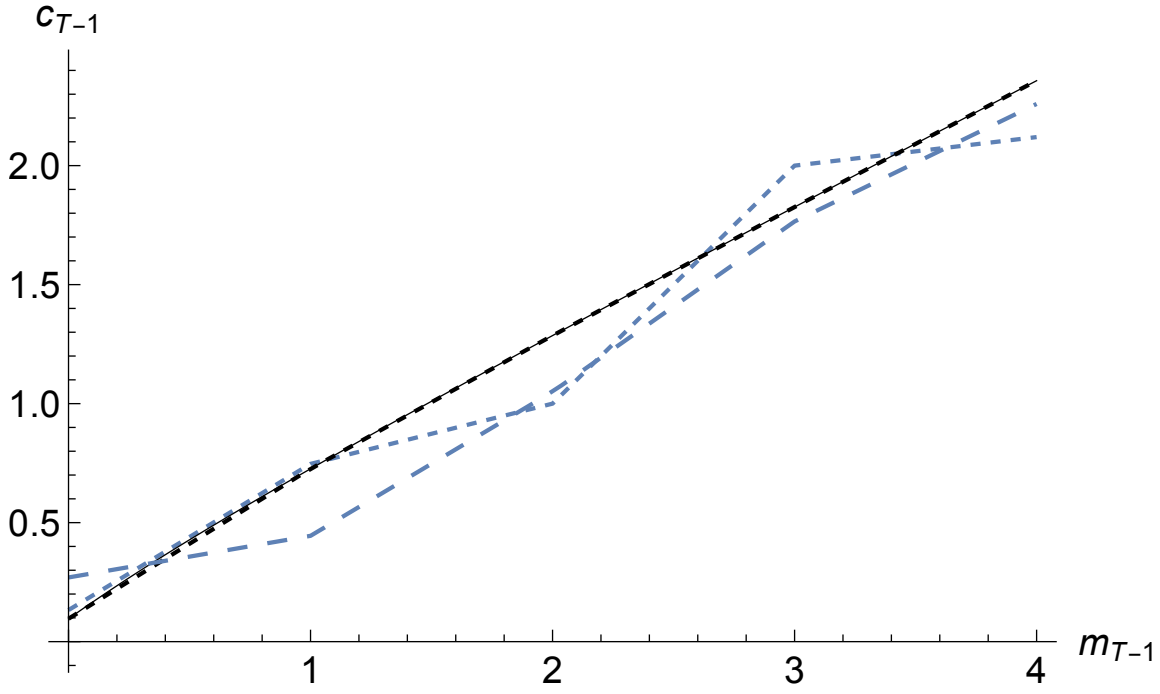

Figure 8 $\mathrm{c}_{t-1}(m)$ (solid) Versus Two Methods for Constructing $\grave{\mathrm{c}}_{t-1}(m)$

{fig:PlotcTm1ABC}

1.6 Transformation

{subsec:transformat

Even the new-and-improved consumption function diverges notably from the true solution, especially at lower values of m. That is because the linear interpolation does an increasingly poor job of capturing the nonlinearity of $\mathrm{v}^{a}_{(t-1)\rightarrow}(a_{t-1})$ at lower and lower levels of a.

This is where we unveil our next trick. To understand the logic, start by considering the case where $\mathcal{R}_t = \beta = \mathcal{G}_t = 1$ and there is no uncertainty (that is, we know for sure that income next period will be $\boldsymbol{\theta}_t = 1$). The final Euler equation (recall that we are still assuming that $t = T$) is then:

$$c_{t-1}^{-\rho} = c_{t}^{-\rho}. \tag{9}$$

In the case we are now considering with no uncertainty and no liquidity constraints, the optimizing consumer does not care whether a unit of income is scheduled to be received in the future period t or the current period $t-1$; there is perfect certainty that the income will be received, so the consumer treats its PDV as equivalent to a unit of current wealth. Total resources available at the point when the consumption decision is made is therefore are comprised of two types: current market resources m and 'human wealth' (the PDV of future income) of $h_{t-1} = 1$ (because it is the value of human wealth as of the end of the period, there is only one more period of income of 1 left).

The well-known optimal solution is to spend half of total lifetime resources in period $t-1$ and the remainder in period $t(=T)$. Since total resources are known with certainty to be $m + h_{t-1} = m + 1$, and since $\mathrm{v}^{m}(m) = \mathrm{u}^{c}(\mathrm{c})$, this implies that

$$\mathrm{v}^{m}_{t-1}(m) = \left(\frac{m+1}{2}\right)^{-\rho}. \tag{10}$$

{eq:vPLin}

Of course, this is a highly nonlinear function. However, if we raise both sides of (10) to the power $(-1/\rho)$ the result is a linear function:

$$\left[\mathrm{v}^{m}_{t-1}(m)\right]^{-1/\rho} = \frac{m+1}{2}. \tag{11}$$

This is a specific example of a general phenomenon: A theoretical literature discussed in Carroll and Kimball (1996) establishes that under perfect certainty, if the period-by-period marginal utility function is of the form $c_t^{-\rho}$, the marginal value function will be of the form $(\gamma m_t + \zeta)^{-\rho}$ for some constants $\{\gamma, \zeta\}$. This means that if we were solving the perfect foresight problem numerically, we could always calculate a numerically exact (because linear) interpolation.

To put the key insight in intuitive terms, the nonlinearity we are facing springs in large part from the fact that the marginal value function is highly nonlinear. But we have a compelling solution to that problem, because the nonlinearity springs largely from the fact that we are raising something to the power $-\rho$. In effect, we can 'unwind' all of the nonlinearity owing to that operation and the remaining nonlinearity will not be nearly so great. Specifically, applying the foregoing insights to the end-of-period value function

$\mathrm{v}^a_{t-1}(a)$, we can define an 'inverse marginal value' function

$$\Lambda^a_{t\rightarrow}(a) \equiv \left(\mathrm{v}^a_{t\rightarrow}(a)\right)^{-1/\rho} \tag{12}$$ {eq:cGoth}

which would be linear in the perfect foresight case.[6] We then construct a piecewise-linear interpolating approximation to the Λ^a_t function, $\grave{\Lambda}^a_{t\rightarrow}(a_t)$, and for any a that falls in the range $\{\boldsymbol{a}[1], \boldsymbol{a}[-1]\}$ we obtain our approximation of marginal value from:

$$\grave{\mathrm{v}}^a_t(a) = [\grave{\Lambda}^a_t(a)]^{-\rho} \tag{13}$$

The most interesting thing about all of this, though, is that the Λ^a_t function has another interpretation. Recall our point in (15) that $\mathrm{u}^c(c_t) = \mathrm{v}^a_{\rightarrow}(m_t - c_t)$. Since with CRRA utility $\mathrm{u}^c(c) = c^{-\rho}$, this can be rewritten and inverted

$$\begin{aligned} (c_t)^{-\rho} &= \mathrm{v}^a_{\rightarrow}(a_t) \\ c_t &= \left(\mathrm{v}^a_{t\rightarrow}(a)\right)^{-1/\rho}. \end{aligned} \tag{14}$$

What this means is that for any given a, if we can calculate the marginal value associated with ending the period with that a, then we can learn the level of c that the consumer must have chosen if they ended up with that a as the result of an optimal unconstrained choice. This leads us to an alternative interpretation of Λ^a. It is the function that reveals, for any ending a, how much the agent must have consumed to (optimally) get to that a. We will therefore henceforth refer to it as the 'consumed function:'

$$\grave{\mathrm{c}}_{t\rightarrow}(a_t) \equiv \grave{\Lambda}^a_{t\rightarrow}(a_t). \tag{15}$$ {eq:consumedfn}

Thus, for example, for period $t-1$ our procedure is to calculate the vector of $\boldsymbol{c}$ points on the consumed function:

$$\boldsymbol{c} = \mathrm{c}_{(t-1)\rightarrow}(\boldsymbol{a}) \tag{16}$$ {eq:consumedfnvecs

with the idea that we will construct an approximation of the consumed function $\grave{\mathrm{c}}_{(t-1)\rightarrow}$ as the interpolating function connecting these $\{\boldsymbol{a}, \boldsymbol{c}\}$ points.

1.7 The Natural Borrowing Constraint and the a_{t-1} Lower Bound

{subsec:LiqConstrS

This is the appropriate moment to ask an awkward question: How should an interpolated, approximated 'consumed' function like $\grave{\mathrm{c}}_{(t-1)\rightarrow}(a_{t-1})$ be extrapolated to return an estimated 'consumed' amount when evaluated at an a_{t-1} outside the range spanned by $\{\boldsymbol{a}[1], ..., \boldsymbol{a}[n]\}$?

For most canned piecewise-linear interpolation tools like scipy.interpolate, when the 'interpolating' function is evaluated at a point outside the provided range, the algorithm extrapolates under the assumption that the slope of the function remains constant beyond its measured boundaries (that is, the slope is assumed to be equal to the slope of nearest piecewise segment *within* the interpolated range); for example, if the bottommost gridpoint is $a_1 = \boldsymbol{a}[1]$ and the corresponding consumed level is $c_1 = \mathrm{c}_{(t-1)\rightarrow}(a_1)$ we could

[6] There is a corresponding inverse for the value function: $\Lambda_{t\rightarrow}(a_t) = ((1-\rho)\mathrm{v}_{t\rightarrow})^{1/(1-\rho)}$, and for the marginal marginal value function etc.

calculate the 'marginal propensity to have consumed' $\varkappa_1 = \grave{\mathrm{c}}^a_{(t-1)_\rightarrow}(a_1)$ and construct the approximation as the linear extrapolation below $\boldsymbol{a}[1]$ from:

$$\grave{\mathrm{c}}_{(t-1)_\rightarrow}(a) \equiv c_1 + (a - a_1)\varkappa_1. \tag{17}$$ {eq:ExtrapLin}

To see that this will lead us into difficulties, consider what happens to the true (not approximated) $\mathrm{v}^a_{(t-1)_\rightarrow}(a_{t-1})$ as a_{t-1} approaches a quantity we will call the 'natural borrowing constraint': $\underline{a}_{t-1} = -\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$. From (8) we have

$$\lim_{a \downarrow \underline{a}_{t-1}} \mathrm{v}^a_{(t-1)_\rightarrow}(a) = \lim_{a \downarrow \underline{a}_{t-1}} \beta \mathsf{R} \left(\frac{1}{n_{\boldsymbol{\theta}}}\right) \sum_{i=1}^{n_{\boldsymbol{\theta}}} (a\mathcal{R}_t + \boldsymbol{\theta}_i)^{-\rho}. \tag{18}$$

But since $\underline{\boldsymbol{\theta}} = \boldsymbol{\theta}_1$, exactly at $a = \underline{a}_{t-1}$ the first term in the summation would be $(-\underline{\boldsymbol{\theta}} + \boldsymbol{\theta}_1)^{-\rho} = 1/0^\rho$ which is infinity. The reason is simple: $-\underline{a}_{t-1}$ is the PDV, as of $t-1$, of the *minimum possible realization of income* in t ($\mathcal{R}_t \underline{a}_{t-1} = -\boldsymbol{\theta}_1$). Thus, if the consumer borrows an amount greater than or equal to $\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$ (that is, if the consumer ends $t-1$ with $a_{t-1} \leq -\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$) and then draws the worst possible income shock in period t, they will have to consume zero in period t, which yields $-\infty$ utility and $+\infty$ marginal utility.

As Zeldes (1989) first noticed, this means that the consumer faces a 'self-imposed' (or, as above, 'natural') borrowing constraint (which springs from the precautionary motive): They will never borrow an amount greater than or equal to $\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$ (that is, assets will never reach the lower bound of $\underline{a}_{t-1}$). The constraint is 'self-imposed' in the precise sense that if the utility function were different (say, Constant Absolute Risk Aversion), the consumer might be willing to borrow more than $\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$ because a choice of zero or negative consumption in period t would yield some finite amount of utility.[7]

This self-imposed constraint cannot be captured well when the $\mathrm{v}^a_{(t-1)_\rightarrow}$ function is approximated by a piecewise linear function like $\grave{\mathrm{v}}^m_{(t-1)_\rightarrow}$, because it is impossible for the linear extrapolation below $\underline{a}$ to correctly predict $\mathrm{v}^a_{(t-1)_\rightarrow}(\underline{a}_{t-1}) = \infty$.

So, the marginal value of saving approaches infinity as $a \downarrow \underline{a}_{t-1} = -\underline{\boldsymbol{\theta}}\mathcal{R}_t^{-1}$. But this implies that $\lim_{a \downarrow \underline{a}_{t-1}} \mathrm{c}_{(t-1)_\rightarrow}(a) = (\mathrm{v}^a_{(t-1)_\rightarrow}(a))^{-1/\rho} = 0$; that is, as a approaches its 'natural borrowing constraint' minimum possible value, the corresponding amount of worst-case c must approach *its* lower bound: zero.

The upshot is a realization that all we need to do to address these problems is to prepend each of the $\boldsymbol{a}_{\mathtt{t}-1}$ and $\boldsymbol{c}_{\mathtt{t}-1}$ from (16) with an extra point so that the first element in the mapping that produces our interpolation function is $\{\underline{a}_{t-1}, 0.\}$. This is done in section "The Self-Imposed 'Natural' Borrowing Constraint and the a_{t-1} Lower Bound" of the notebook.

Figure 9 shows the result. The solid line calculates the exact numerical value of the consumed function $\mathrm{c}_{(t-1)_\rightarrow}(a)$ while the dashed line is the linear interpolating approximation $\grave{\mathrm{c}}_{(t-1)_\rightarrow}(a)$. This figure illustrates the value of the transformation: The true function

The vertical axis should be relabeled - not gothic c anymore, instead Λ^a

[7] Though it is very unclear what a proper economic interpretation of negative consumption might be – this is an important reason why CARA utility, like quadratic utility, is increasingly not used for serious quantitative work, though it is still useful for teaching purposes.

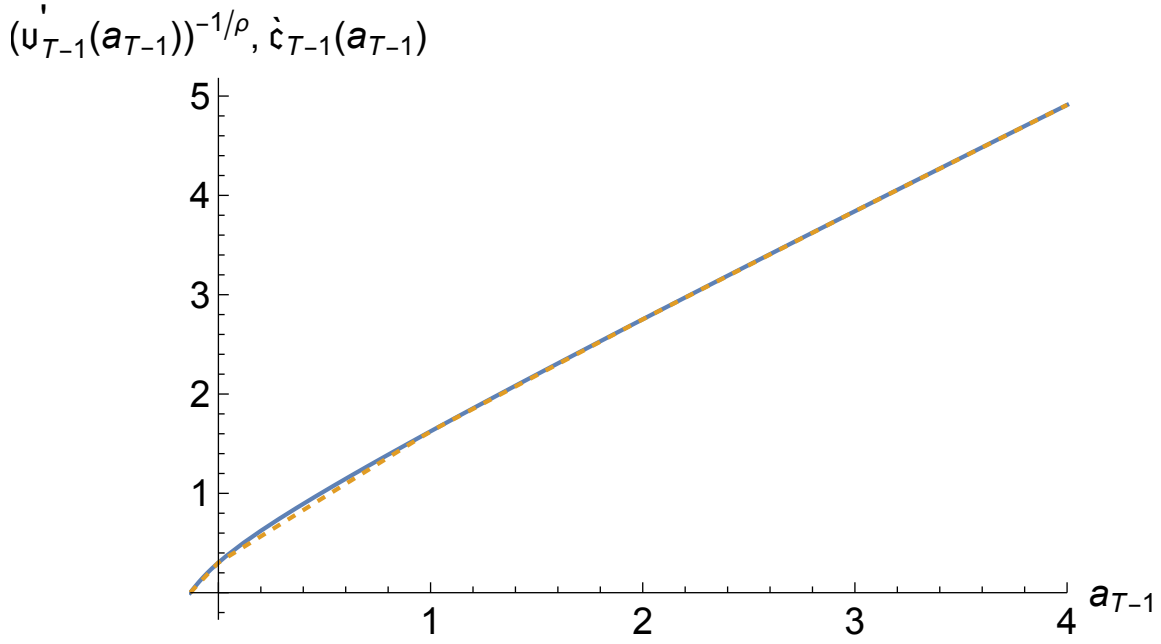

Figure 9 True $\Lambda^{a}_{(t-1)_{\rightarrow}}(a)$ vs its approximation $\grave{\Lambda}^{a}_{(t-1)_{\rightarrow}}(a)$ {fig:GothVInvVSGo

is close to linear, and so the linear approximation is almost indistinguishable from the true function except at the very lowest values of a.

Figure 10 similarly shows that when we generate $\grave{\mathfrak{v}}^{a}_{(t-1)_{\rightarrow}}(a)$ using our augmented $[\grave{\mathfrak{c}}_{(t-1)_{\rightarrow}}(a)]^{-\rho}$ (dashed line) we obtain a *much* closer approximation to the true marginal value function $\mathfrak{v}^{a}_{(t-1)_{\rightarrow}}(a)$ (solid line) than we obtained in the previous exercise which did not do the transformation (Figure 7).[8]

fix the problem articulated in the footnote

1.8 The Method of Endogenous Gridpoints ('EGM')

{subsec:egm}

The solution procedure we articulated above for finding $\mathfrak{c}_{t-1}(m)$ still requires us, for each point in $\boldsymbol{m}_{\mathtt{t}-1}$, to use a numerical rootfinding algorithm to search for the value of $\mathfrak{c}$ that solves $\mathfrak{u}^{c}(\mathfrak{c}) = \mathfrak{v}^{a}_{(t-1)_{\rightarrow}}(m - \mathfrak{c})$. Though sections 1.6 and 1.7 developed a highly efficient and accurate procedure to calculate $\grave{\mathfrak{v}}^{a}_{(t-1)_{\rightarrow}}$, those approximations do nothing to eliminate the need for using a rootfinding operation for calculating, for an arbitrary m, the optimal c. And rootfinding is a notoriously computation-intensive (that is, slow!) operation.

Fortunately, it turns out that there is a way to completely skip this slow rootfinding step. The method can be understood by noting that we have already calculated, for a set of arbitrary values of $\boldsymbol{a} = \boldsymbol{a}_{\mathtt{t}-1}$, the corresponding $\boldsymbol{c}$ values for which this $\boldsymbol{a}$ is optimal.

[8] The vertical axis label uses $\mathfrak{v}'$ as an alternative notation for what in these notes we designate as $\mathfrak{v}^{a}_{\rightarrow}$). This will be fixed.

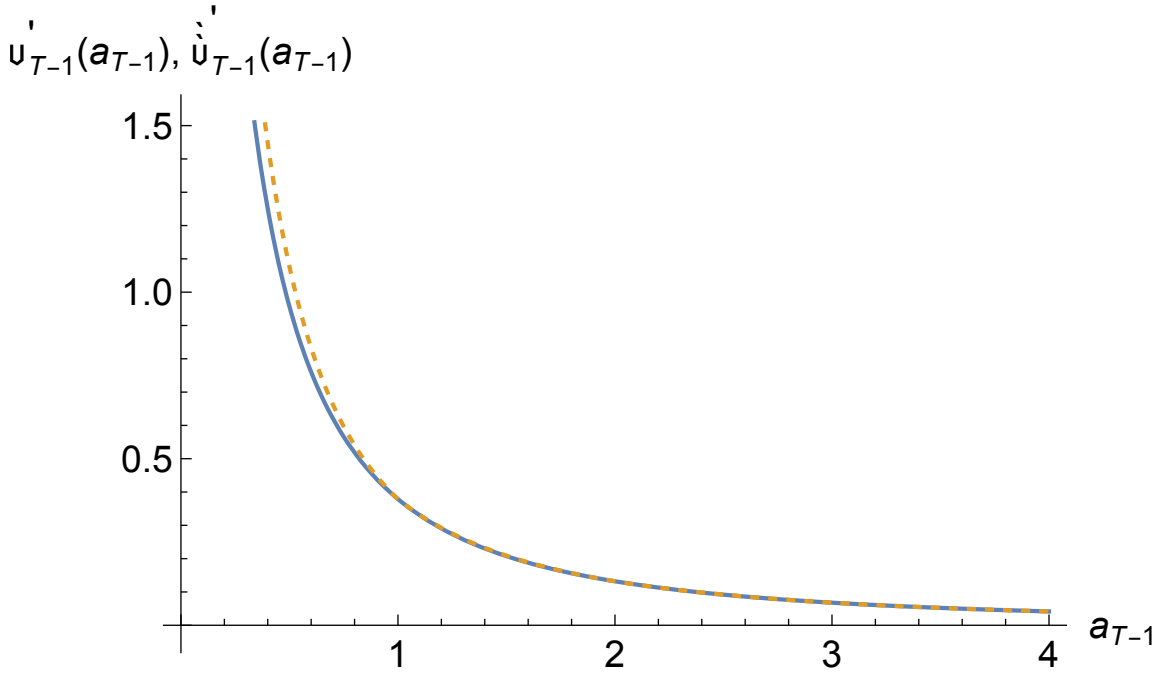

Figure 10 True $\mathrm{v}^{a}_{(t-1)_{\rightarrow}}(a)$ vs. $\check{\mathrm{v}}^{a}_{(t-1)_{\rightarrow}}(a)$ Constructed Using $\grave{\mathrm{c}}_{(t-1)_{\rightarrow}}(a)$

{fig:GothVVSGothC

But with mutually consistent values of $\boldsymbol{c}_{\mathtt{t}-1}$ and $\boldsymbol{a}_{\mathtt{t}-1}$ (consistent, in the sense that they are the unique optimal values that correspond to the solution to the problem), we can obtain the $\boldsymbol{m}_{\mathtt{t}-1}$ vector that corresponds to both of them from

$$\boldsymbol{m}_{\mathtt{t}-1} = \boldsymbol{c}_{\mathtt{t}-1} + \boldsymbol{a}_{\mathtt{t}-1}. \tag{19}$$

These m gridpoints are "endogenous" in contrast to the usual solution method of specifying some *ex-ante* (exogenous) grid of values of $\boldsymbol{m}$ and then using a rootfinding routine to locate the corresponding optimal consumption vector $\boldsymbol{c}$.

This routine is performed in the "Endogenous Gridpoints" section of the notebook. First, the `gothic.C_Tminus1` function is called for each of the pre-specfied values of end-of-period assets stored in `aVec`. These values of consumption and assets are used to produce the list of endogenous gridpoints, stored in the object `mVec_egm`. With the $\mathbf{c}$ values in hand, the notebook can generate a set of $\boldsymbol{m}_{\mathtt{t}-1}$ and $\boldsymbol{c}_{\mathtt{t}-1}$ pairs that can be interpolated between in order to yield $\grave{\mathrm{c}}_{t-1}(m)$ at virtually zero computational cost![9]

Rename gothic class, maybe to: EndPrd. Also, harmonize the notation in the notebook in the paper - for example, everwhere in the text we use cNrm for normalized consumption, but for some reason it is capital C in the gothic function.

One might worry about whether the $\{m, c\}$ points obtained in this way will provide a good representation of the consumption function as a whole, but in practice there are good reasons why they work well (basically, this procedure generates a set of gridpoints that is naturally dense right around the parts of the function with the greatest non-linearity). Figure 11 plots the actual consumption function c_{t-1} and the approximated consumption function $\grave{\mathrm{c}}_{t-1}$ derived by the method of endogenous grid points. Compared

[9] This is the essential point of Carroll (2006).

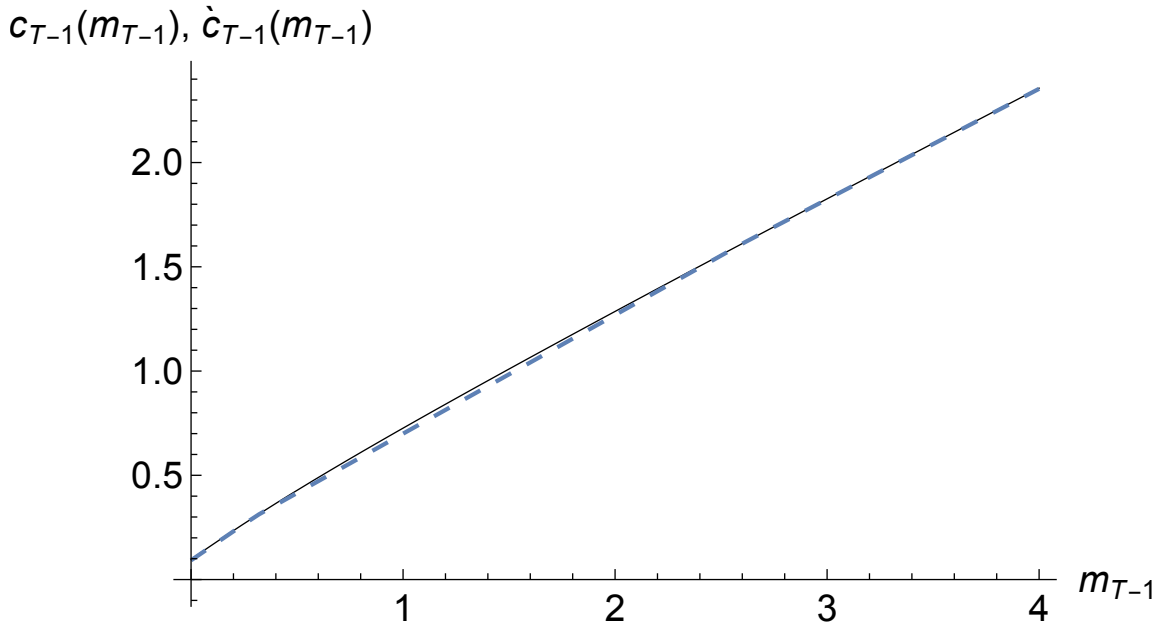

Figure 11 $\mathrm{c}_{t-1}(m)$ (solid) versus $\grave{\mathrm{c}}_{t-1}(m)$ (dashed)

{fig:ComparecTm1A

to the approximate consumption functions illustrated in Figure 8, $\grave{\mathrm{c}}_{t-1}$ is quite close to the actual consumption function.

1.9 Improving the a Grid

{subsec:improving-t

Thus far, we have arbitrarily used a gridpoints of $\{0., 1., 2., 3., 4.\}$ (augmented in the last subsection by $\underline{a}_{t-1}$). But it has been obvious from the figures that the approximated $\grave{\mathrm{c}}_{(t-1)\rightarrow}$ function tends to be farthest from its true value at low values of a. Combining this with our insight that $\underline{a}_{t-1}$ is a lower bound, we are now in position to define a more deliberate method for constructing gridpoints for a – a method that yields values that are more densely spaced at low values of a where the function is more nonlinear.

A pragmatic choice that works well is to find the values such that (1) the last value *exceeds the lower bound* by the same amount $\bar{a}$ as our original maximum gridpoint (in our case, 4.); (2) we have the same number of gridpoints as before; and (3) the *multi-exponential growth rate* (that is, $e^{e^{e^{\cdots}}}$ for some number of exponentiations n – our default is 3) from each point to the next point is constant (instead of, as previously, imposing constancy of the absolute gap between points).

Section "Improve the $\mathbb{A}_{grid}$" begins by defining a function which takes as arguments the specifications of an initial grid of assets and returns the new grid incorporating the multi-exponential approach outlined above.

Notice that the graphs depicted in Figures 12 and 13 are notably closer to their respective truths than the corresponding figures that used the original grid.

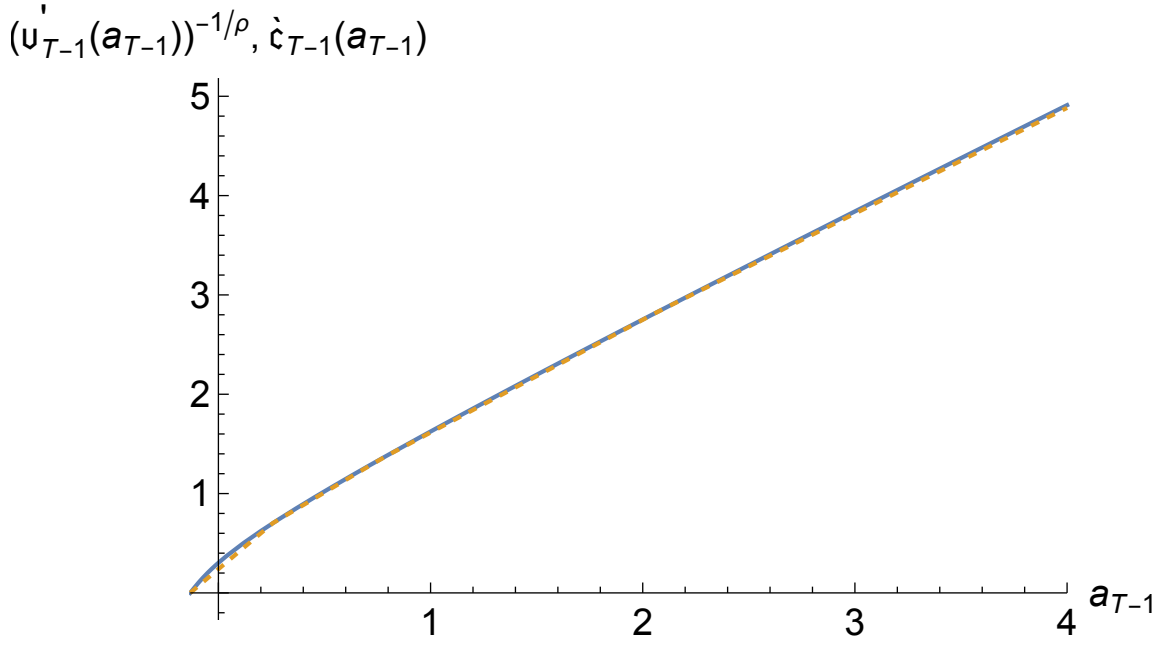

Figure 12 $\mathrm{c}_{(t-1)_\rightarrow}(a)$ versus $\grave{\mathrm{c}}_{(t-1)_\rightarrow}(a)$, Multi-Exponential `aVec`

{fig:GothVInvVSGo

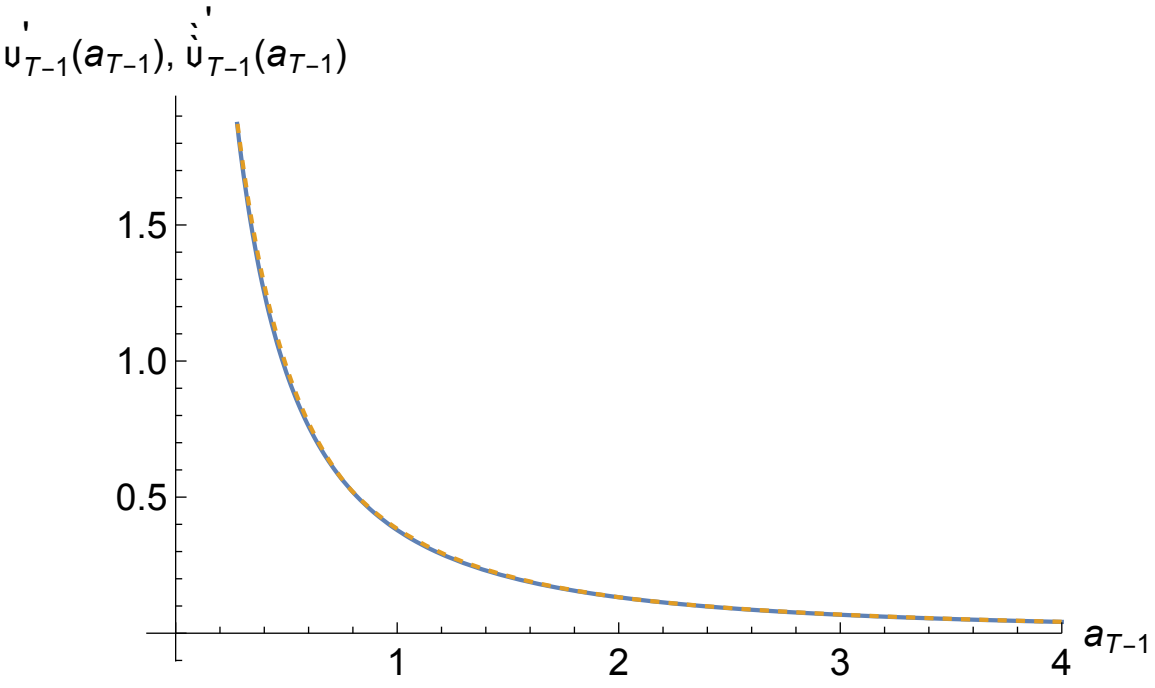

Figure 13 $\mathrm{v}^a_{(t-1)_\rightarrow}(a)$ vs. $\grave{\mathrm{v}}^a_{(t-1)_\rightarrow}(a)$, Multi-Exponential `aVec`

{fig:GothVVSGothC

1.10 Program Structure

In section "Solve for $c_t(m)$ in Multiple Periods," the natural and artificial borrowing constraints are combined with the endogenous gridpoints method to approximate the optimal consumption function for a specific period. Then, this function is used to compute the approximated consumption in the previous period, and this process is repeated for some specified number of periods.

The essential structure of the program is a loop that iteratively solves for consumption functions by working backward from an assumed final period, using the dictionary `cFunc_life` to store the interpolated consumption functions up to the beginning period. Consumption in a given period is utilized to determine the endogenous gridpoints for the preceding period. This is the sense in which the computation of optimal consumption is done recursively.

For a realistic life cycle problem, it would also be necessary at a minimum to calibrate a nonconstant path of expected income growth over the lifetime that matches the empirical profile; allowing for such a calibration is the reason we have included the $\{\mathcal{G}\}_t^T$ vector in our computational specification of the problem.

1.11 Results

The code creates the relevant $\grave{\text{c}}_t(m)$ functions for any period in the horizon, at the given values of m. Figure 14 shows $\grave{\text{c}}_{T-n}(m)$ for $n = \{20, 15, 10, 5, 1\}$. At least one feature of this figure is encouraging: the consumption functions converge as the horizon extends, something that Carroll (2023b) shows must be true under certain parametric conditions that are satisfied by the baseline parameter values being used here.

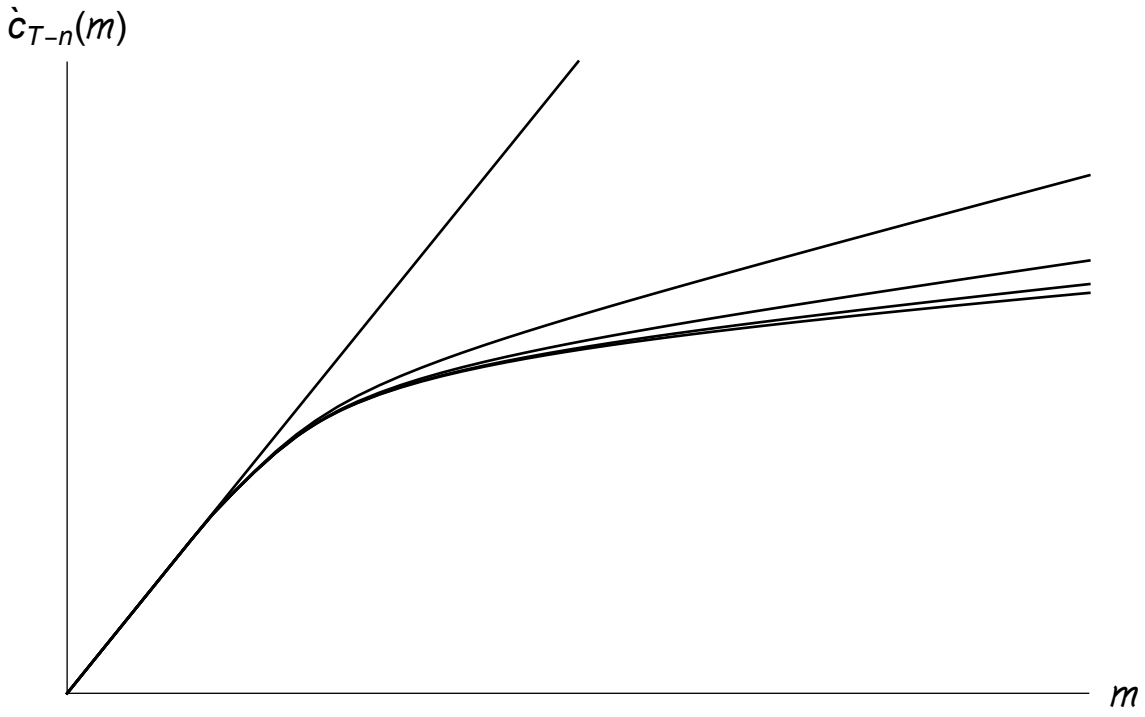

Figure 14 Converging $\grave{c}_{T-n}(m)$ Functions as n Increases

{fig:PlotCFuncsCon