g-the-next}

1 Solving the Next-to-Last Period

This section examines the second-to-last period in detail, illustrating a number of powerful techniques for speeding and improving its solution; in doing so it illustrates how the pile `P` (subsection 4.4) is built backward. We set $\mathcal{G}_t = 1$ here to reduce clutter.

bstitution}

1.1 A Direct Expression for c_{T-1}

If $t = T$, the second-to-last-period decision-perch problem is

vEndPrdTm1}

$$\mathrm{v}_{\sim(t-1)}(m) = \max_{c} \quad \mathrm{u}(c) + \mathrm{v}_{\succ(t-1)}(\overbrace{m-c}^{a}) \,\mathsf{U} \tag{1}$$

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

:NumDefInt}

$$\mathrm{v}_{\prec(t)}(a) = \int_0^\infty \frac{(a\mathcal{R}_t + \vartheta)^{1-\rho}}{1-\rho} d\mathcal{F}(\vartheta) \,\mathsf{U}, \tag{2}$$

where $\mathcal{F}(\theta)$ is the cumulative distribution function for θ, this maximization problem implicitly defines a 'period-and-stage-local function' c that yields optimal consumption in period $t-1$ for any specific numerical level of resources like $m = 1.7$. The explicit statement of the problem is

q:cFuncTm1}

$$\mathrm{c}_{\sim(t-1)}(m) = \arg\max_{c} \quad \mathrm{u}(c) + \beta \int_0^\infty \frac{((m-c)\mathcal{R}_t + \vartheta)^{1-\rho}}{1-\rho} d\mathcal{F}(\vartheta). \,\mathsf{U} \tag{3}$$

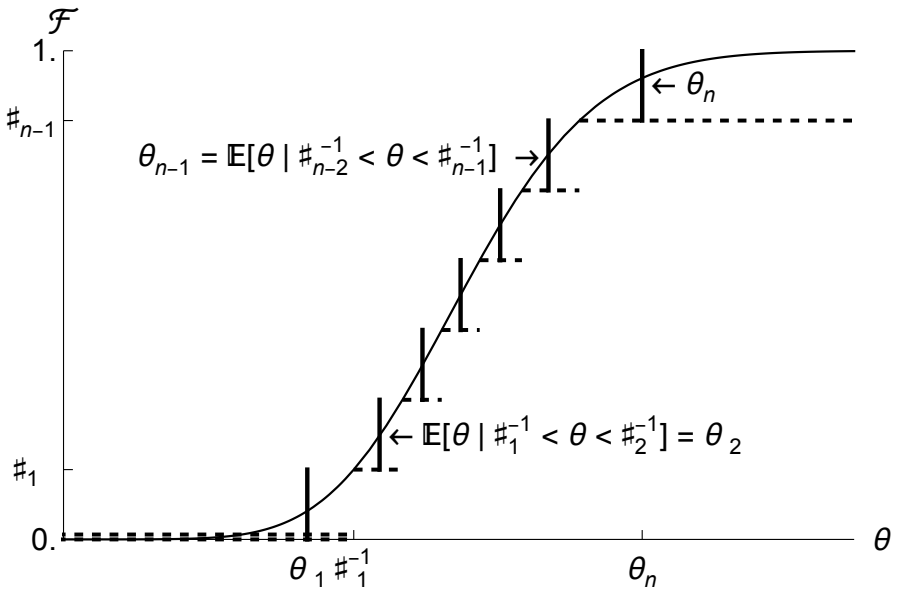
But because there is no general analytical solution, for any given m we must use numerical tools to find the c that maximizes the expression. This is excruciatingly slow: for every candidate c, a definite integral over $(0, \infty)$ must be calculated numerically.

1.2 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 θ of a function $g(\theta)$ by calculating its value at set of n_θ points θ_i, each of which has an associated probability weight w_i:

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

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



reteapprox}

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

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 θ_i has an equal probability, equal to $1/n_\theta$).

We calculate such an n-point approximation as follows.

Define a set of points from $\sharp_0$ to $\sharp_{n_\theta}$ 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 θ distribution $\mathcal{F}^{-1}$, and define the points $\sharp_i^{-1} = \mathcal{F}^{-1}(\sharp_i)$. Then the conditional mean of θ in each of the intervals numbered 1 to n is:

$$\theta_i \equiv \mathbb{E}[\theta | \sharp_{i-1}^{-1} \leq \theta < \sharp_i^{-1}] = \int_{\sharp_{i-1}^{-1}}^{\sharp_i^{-1}} \vartheta \, d\mathcal{F}\,(\vartheta), \cup \tag{4}$$

and when the integral is evaluated numerically for each i the result is a set of values of θ 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 $\mathcal{F}(\theta)$ for a lognormal distribution such that $\mathbb{E}[\theta] = 1$, $\sigma_\theta = 0.1$. The short vertical line segments represent the n_θ equiprobable values of θ_i which are used to approximate this distribution.[2]

The following notebook snippet constructs these points.

We now substitute our approximation (5) for $\mathrm{v}_{\succ(t-1)}(a)$ in (1) which is simply the sum of n_θ numbers and is therefore easy to calculate (compared to the full-fledged numerical integration (2) that it replaces).

[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.

:vDiscrete}

$$\mathrm{v}_{\succ(t-1)}(a) = \beta \left(\frac{1}{n_\theta}\right) \sum_{i=1}^{n_\theta} \frac{(\mathcal{R}_t a + \theta_i)^{1-\rho}}{1-\rho} \cup \tag{5}$$

1.3 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 responsible for computing an estimated consumption function begins in "Solving the Model by Value Function Maximization," where a vector of possible values of market resources m is created. In these notes we use $\mathbf{m}$ for such a vector (e.g. $\mathbf{m}[1]$ the first entry, $\mathbf{m}[-1]$ the last). For illustration we take the grid to be the first five nonnegative integers, $\{0, 1, 2, 3, 4\}$.

1.4 An Interpolated Consumption Function

:LinInterp}

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

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).[3]

1.5 Interpolating Expectations

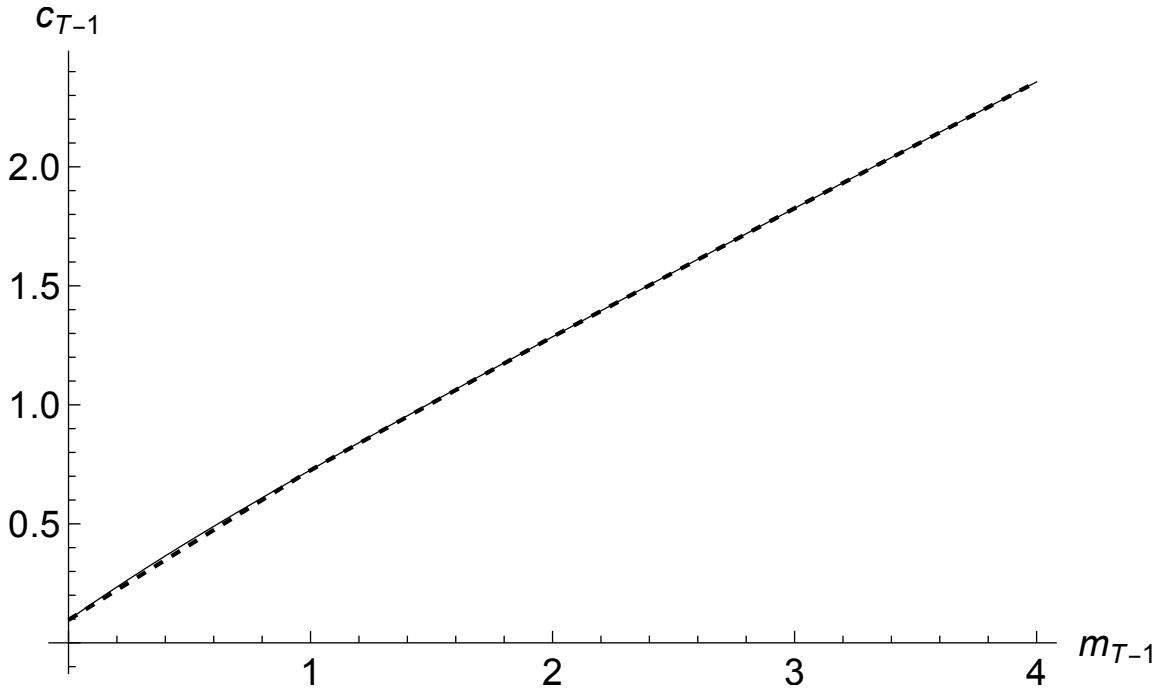
Piecewise linear 'spline' interpolation as described above works well for generating a good approximation to the true optimal consumption function. However, there is a clear inefficiency in the program: Since it uses equation (1), for every value of m the program must calculate the utility consequences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

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

The notebook section "Interpolating Expectations," now interpolates the expected value of *ending* the period with a given amount of assets.[4]

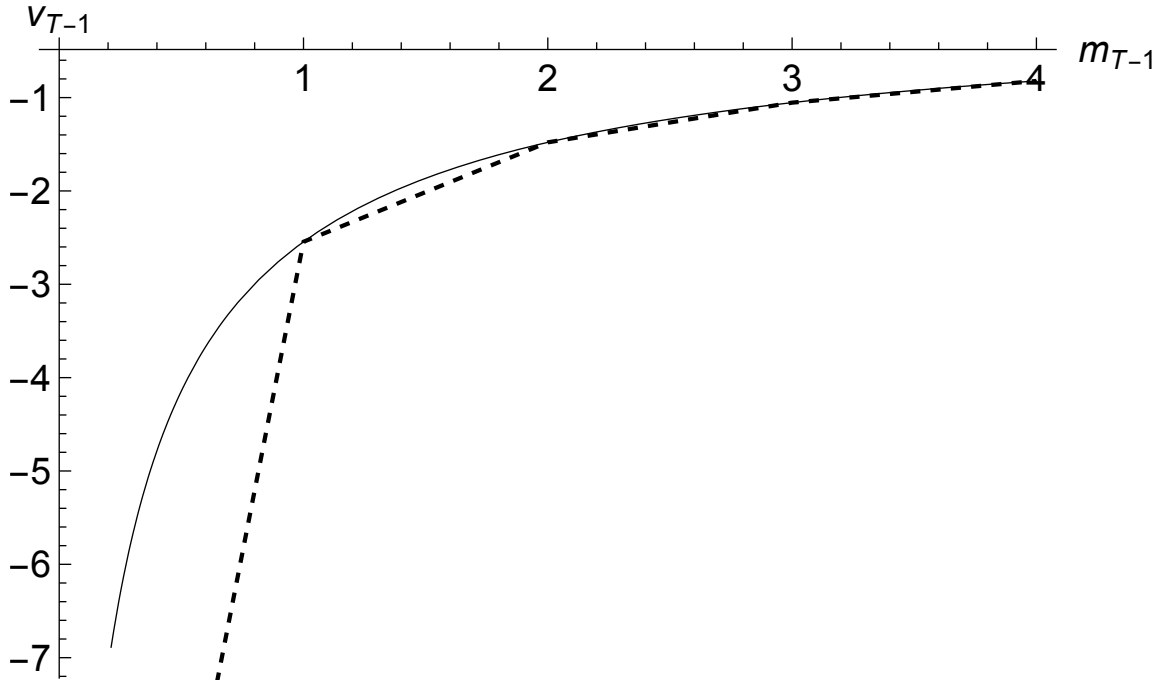
[3] For some problems, especially ones with discrete choices, value function approximation is unavoidable; nevertheless, even in such problems, the techniques sketched below can be very useful across much of the range over which the problem is defined.

[4] 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.



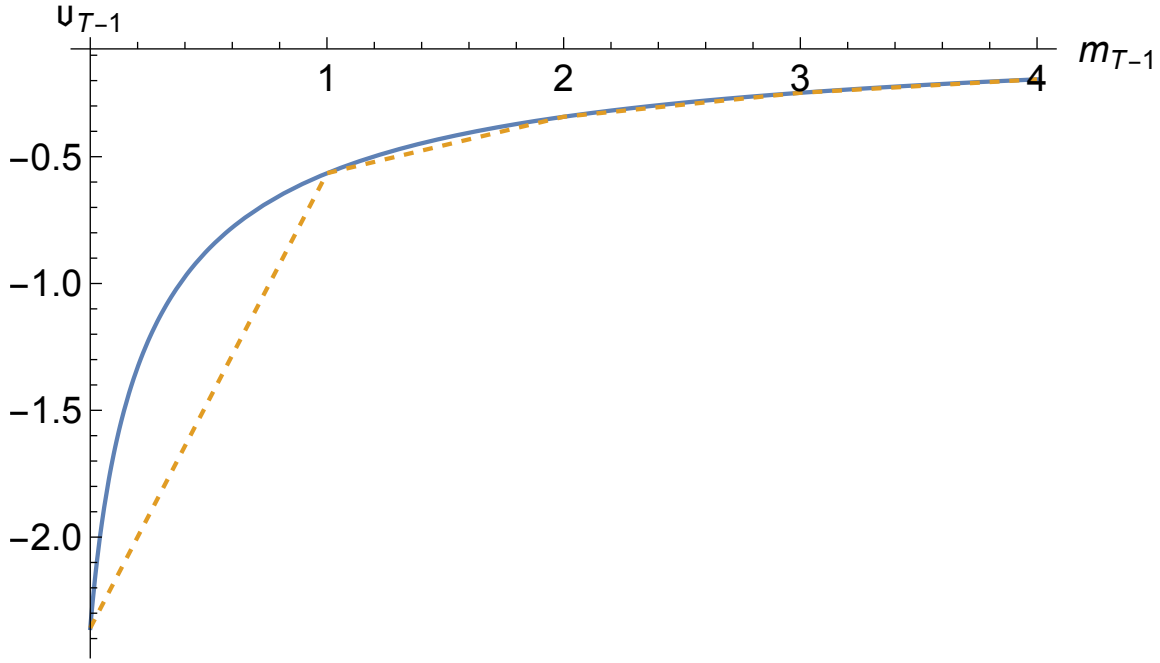
cTm1Simple}

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



VTm1Simple}

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



m1RawVSInt}

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

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 **a** and they appear reasonably close except in the region below $m = 1$.

Nevertheless, the consumption rule obtained when the approximating $\grave{v}_{\succ(t-1)}(a_{t-1})$ is used instead of $v_{\succ(t-1)}(a_{t-1})$ is surprisingly bad, as shown in figure 5. For example, when m goes from 2 to 3, $\grave{c}_{t-1}$ goes from about 1 to about 2, yet when m goes from 3 to 4, $\grave{c}_{t-1}$ 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.

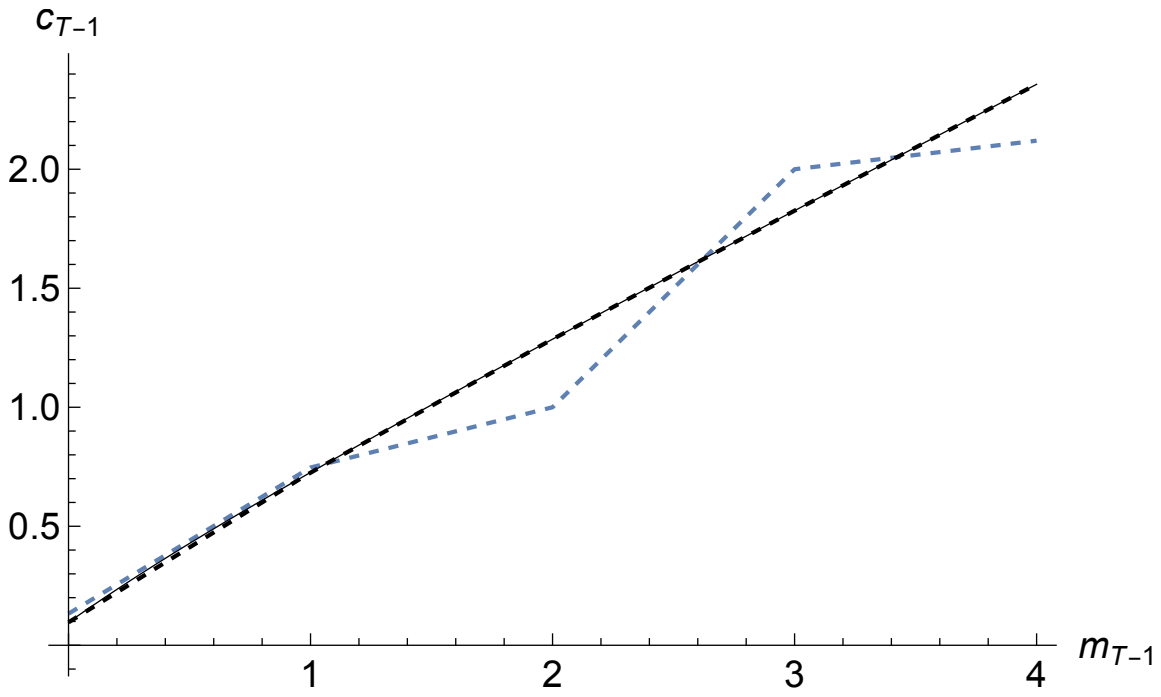
bsec:vVsuP}

1.6 Value Function versus First Order Condition

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,

$$\mathbb{E}[-(c - \not{c})^2] < -(\mathbb{E}[c] - \not{c})^2 \cup \tag{6}$$

(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 consumers is unaffected by risk since behavior is determined by the first order condition,



{parecTm1AB}

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

which depends on *marginal* utility, and when utility is quadratic, marginal utility is unaffected by risk:

$$\mathbb{E}[-2(c-\not{c})] = -2(\mathbb{E}[c]-\not{c}). \mathsf{U} \tag{7}$$

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.

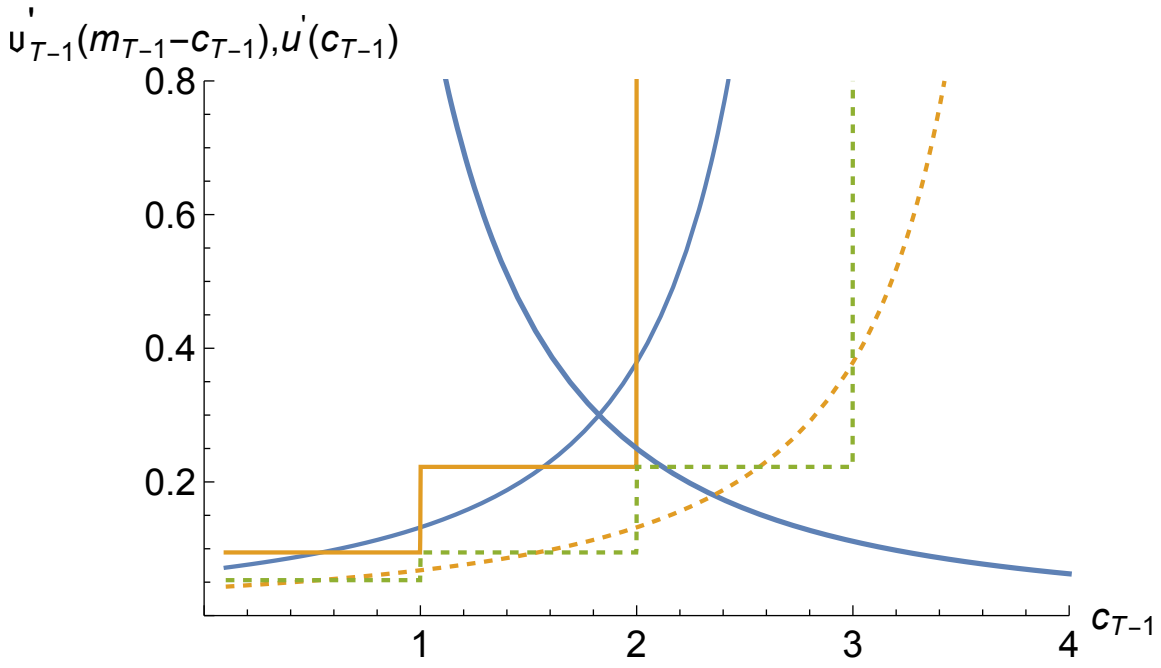
The first order condition of the maximization problem in period $T-1$ is:

{eq:FOCTm1}

$$\begin{aligned} \mathrm{u}^c(c) &= \beta \mathbb{E}_{\succ(T-1)}[\mathsf{R}\mathrm{u}^c(c_t)] \\ c^{-\rho} &= \mathsf{R}\beta \left(\frac{1}{n_\theta}\right) \sum_{i=1}^{n_\theta} (\mathsf{R}(m-c)+\theta_i)^{-\rho} \,. \mathsf{U} \end{aligned} \tag{8}$$

The downward-sloping curve in Figure 6 shows the value of $c^{-\rho}$ for our baseline parameter values for $0 \leq c \leq 4$ (the horizontal axis). The solid upward-sloping curve shows the value of the RHS of (8) as a function of c under the assumption that $m = 3$. Optimal consumption given $m = 3$ is the c at which the two curves intersect—just below $c = 2$. The dashed curve shows the same for $m = 4$; its intersection with $\mathrm{u}^c(c)$ is slightly below $c = 2.5$, so increasing m from 3 to 4 raises optimal consumption by about 0.5.

Now consider the derivative of $\grave{\mathrm{v}}_{(t-1)}$. Because the function is piecewise linear, its



meVSOPrime}

Figure 6 $\mathrm{u}^c(c)$ versus $\mathrm{v}^{\partial}_{\succ(T-1)}(3-c), \mathrm{v}^{\partial}_{\succ(T-1)}(4-c), \grave{\mathrm{v}}^{\partial}_{\succ(T-1)}(3-c), \grave{\mathrm{v}}^{\partial}_{\succ(T-1)}(4-c)$

derivative $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(a_{t-1})$ is a step function: constant between adjacent gridpoints, with jumps at each gridpoint.

The solid-line step function in Figure 6 depicts the actual value of $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(3-c)$. When we attempt to find optimal values of c given m using $\grave{\mathrm{v}}_{\succ(t-1)}(a_{t-1})$, the numerical optimization routine will return the c for which $\mathrm{u}^c(c) = \grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(m-c)$. Thus, for $m = 3$ the program will return the value of c for which the downward-sloping $\mathrm{u}^c(c)$ curve intersects with the $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(3-c)$; 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 $\mathrm{u}^c(c)$ with $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(4-c)$; 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 $\mathrm{u}^c(c)$ and $\mathrm{v}^{\partial}_{\succ(t-1)}(m-c)$ if our estimate of $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}$ 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 $\mathbf{a}$:

ndPrimeTm1}

$$\mathrm{v}^{\partial}_{\succ(t-1)}(\mathbf{a}) = \beta \mathsf{R} \left(\frac{1}{n_\theta}\right) \sum_{i=1}^{n_\theta} (\mathcal{R}_t \mathbf{a} + \theta_i)^{-\rho} \; \mathsf{U} \tag{9}$$

yielding $\mathbf{v}^{\partial}_{\succ(t-1)}$ (the vector of expected end-of-period-$(T-1)$ marginal values of assets

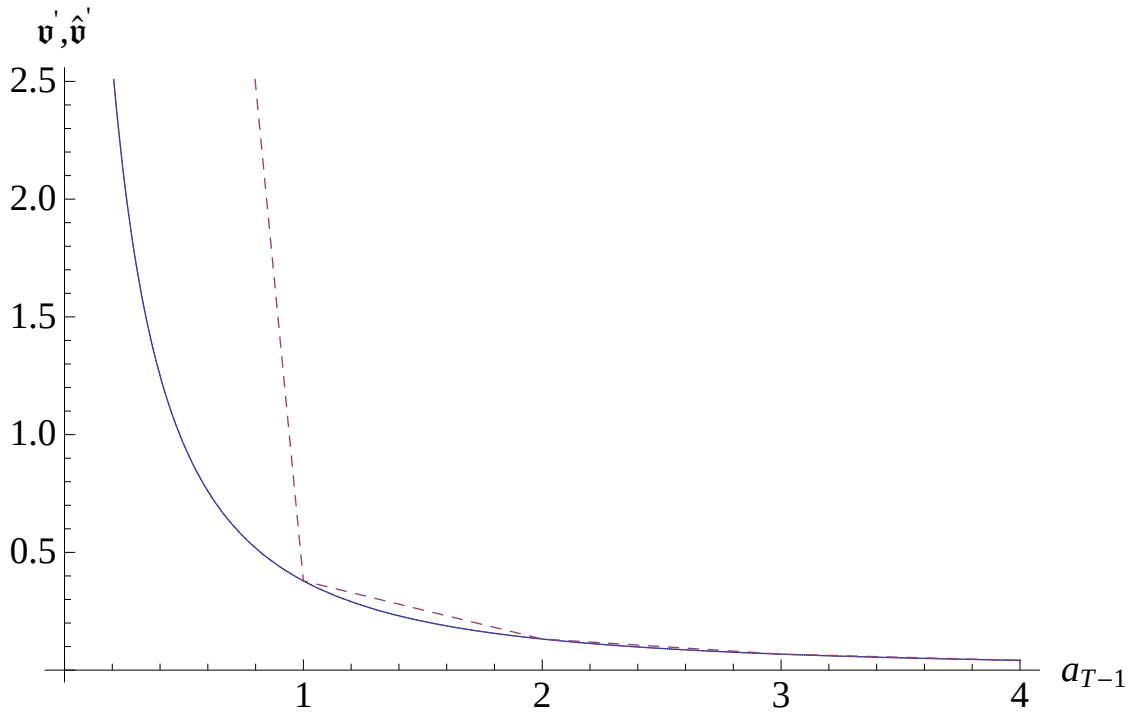


Figure 7 $\mathrm{v}^{\partial}_{\succ(t-1)}(a_{t-1})$ versus $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(a_{t-1})$

OPRawVSFOC}

corresponding to `aVec`), and construct $\grave{\mathrm{v}}^{\partial}_{\succ(t-1)}(a_{t-1})$ 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).

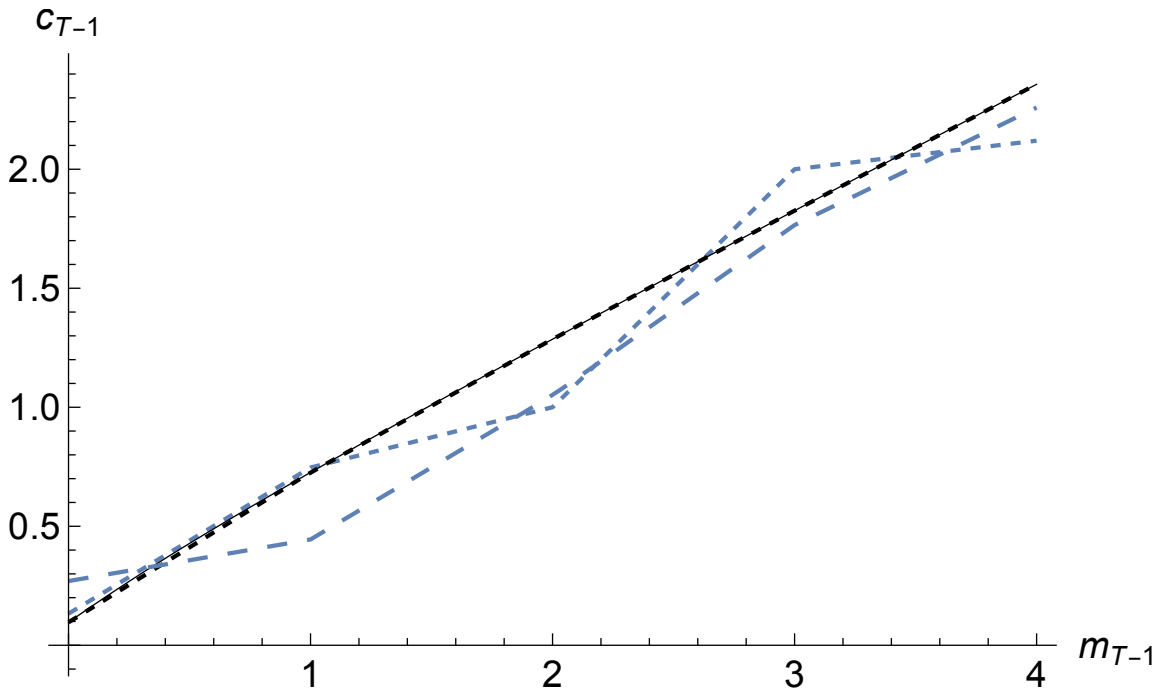
1.7 Transformation

sformation}

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 $\mathrm{v}^{\partial}_{\succ(t-1)}$ 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 $\theta_t = 1$). The final Euler equation (recall that we are still assuming that $t = T$) is then:

$$c_{t-1}^{-\rho} = c_t^{-\rho}. \mathsf{U} \tag{10}$$



lotcTm1ABC}

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

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 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).

{eq:vPLin}

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

Of course, this is a highly nonlinear function. However, if we raise both sides of (11) to the power $(-1/\rho)$ the result is a linear function:

$$\left[\mathrm{v}^{\partial}_{\sim(t-1)}(m)\right]^{-1/\rho} = \frac{m+1}{2}. \tag{12}$$

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.

The key insight is that much of the nonlinearity in v^{∂} comes from raising to the power $-\rho$. By inverting that operation (raising to $-1/\rho$), we can 'unwind' it, and the remaining nonlinearity is much smaller. Specifically, applying the foregoing insights to the end-of-period value function $\mathrm{v}^{\partial}_{\sim(t-1)}(a)$, we can define an 'inverse marginal value' function

{eq:cGoth}
$$\Lambda^{\partial}_{\succ}(a) \equiv \left(\mathrm{v}^{\partial}_{\succ}(a)\right)^{-1/\rho} \,\mathsf{U} \tag{13}$$

which would be linear in the perfect foresight case.[5] We then construct a piecewise-linear interpolating approximation to the Λ^{∂}_{t} function, $\grave{\Lambda}^{\partial}_{\succ}(a_t)$, and for any a that falls in the range $\{\mathbf{a}[1], \mathbf{a}[-1]\}$ we obtain our approximation of marginal value from:

$$\grave{\mathrm{v}}^{\partial}_{\sim}(a) = [\grave{\Lambda}^{\partial}(a)]^{-\rho} \,\mathsf{U} \tag{14}$$

The most interesting thing about all of this, though, is that the Λ^{∂}_{t} function has another interpretation. Recall our point in (23) that $\mathrm{u}^{c}(c_t) = \mathrm{v}^{\partial}_{\succ}(m_t - c_t)$. Since with CRRA utility $\mathrm{u}^{c}(c) = c^{-\rho}$, this can be rewritten and inverted

$$\begin{aligned} (\mathrm{c}_{\succ}(a))^{-\rho} &= \mathrm{v}^{\partial}_{\succ}(a) \\ \mathrm{c}_{\succ}(a) &= \left(\mathrm{v}^{\partial}_{\succ}(a)\right)^{-1/\rho} . \,\mathsf{U} \end{aligned} \tag{15}$$

This gives Λ^{∂} a concrete interpretation: for any ending a, it reveals how much the agent must *have consumed* to (optimally) reach that a. We will therefore henceforth refer to it as the 'consumed function:'

consumedfn}
$$\grave{\mathrm{c}}_{\succ}(a) \equiv \grave{\Lambda}^{\partial}_{\succ}(a_t) . \,\mathsf{U} \tag{16}$$

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

umedfnvecs}
$$\mathbf{c} = \mathrm{c}_{\succ(t-1)}(\mathbf{a}) \,\mathsf{U} \tag{17}$$

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

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

elfImposed}

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

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

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

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 $a_1 = \mathbf{a}[1]$ and the corresponding consumed level is $c_1 = \mathrm{c}_{\succ(t-1)}(a_1)$ we could calculate the 'marginal propensity to have consumed' $\varkappa_1 = \grave{\mathrm{c}}^{\partial}_{\succ(t-1)}(a_1)$ and construct the approximation as the linear extrapolation below $\mathbf{a}[1]$ from:

:ExtrapLin}

$$\grave{\mathrm{c}}_{\succ(t-1)}(a) \equiv c_1 + (a - a_1)\varkappa_1 \,. \mathsf{U} \tag{18}$$

To see that this will lead us into difficulties, consider what happens to the true (not approximated) $\mathrm{v}^{\partial}_{\succ(t-1)}(a_{t-1})$ as a_{t-1} approaches a quantity we will call the 'natural borrowing constraint': $\underline{a}_{t-1} = -\underline{\theta}\mathcal{R}_t^{-1}$. From (9) we have

$$\lim_{a \downarrow \underline{a}_{t-1}} \mathrm{v}^{\partial}_{\succ(t-1)}(a) = \lim_{a \downarrow \underline{a}_{t-1}} \beta \mathsf{R} \left(\frac{1}{n_\theta}\right) \sum_{i=1}^{n_\theta} (a\mathcal{R}_t + \theta_i)^{-\rho} \,. \mathsf{U} \tag{19}$$

But since $\underline{\theta} = \theta_1$, exactly at $a = \underline{a}_{t-1}$ the first term in the summation would be $(-\underline{\theta} + \theta_1)^{-\rho} = 1/0^{\rho}$ 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 ($\mathcal{R}_t \underline{a}_{t-1} = -\theta_1$). Thus, if the consumer borrows an amount greater than or equal to $\underline{\theta}\mathcal{R}_t^{-1}$ (that is, if the consumer ends $t-1$ with $a_{t-1} \leq -\underline{\theta}\mathcal{R}_t^{-1}$) 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 $\underline{\theta}\mathcal{R}_t^{-1}$ (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 $\underline{\theta}\mathcal{R}_t^{-1}$ because a choice of zero or negative consumption in period t would yield some finite amount of utility.[6]

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

So, the marginal value of saving approaches infinity as $a \downarrow \underline{a}_{t-1} = -\underline{\theta}\mathcal{R}_t^{-1}$. But this implies that $\lim_{a \downarrow \underline{a}_{t-1}} \mathrm{c}_{\succ(t-1)}(a) = (\mathrm{v}^{\partial}_{\succ(t-1)}(a))^{-1/\rho} = 0$; 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 $\mathbf{a}_{\mathtt{t-1}}$ and $\mathbf{c}_{\mathtt{t-1}}$ from (17) 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

[6] 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.

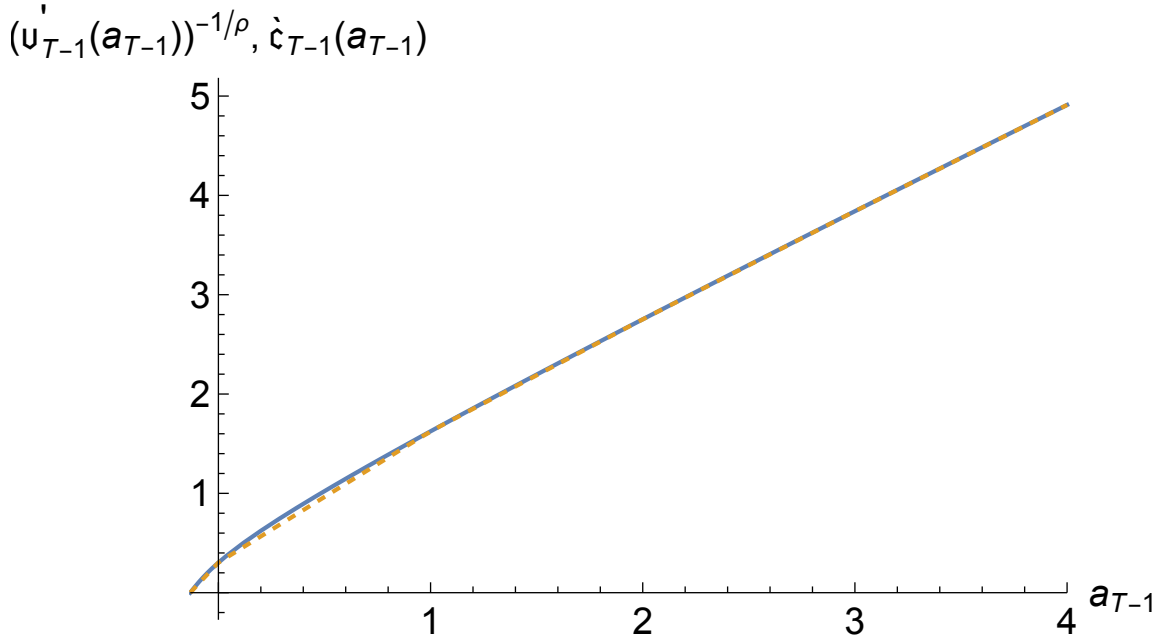


Figure 9 True $\Lambda^{\partial}_{\succ(t-1)}(a)$ vs its approximation $\grave{\Lambda}^{\partial}_{\succ(t-1)}(a)$

InvVSGothC}

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 $\mathfrak{c}_{\succ(t-1)}(a)$ while the dashed line is the linear interpolating approximation $\grave{\mathfrak{c}}_{\succ(t-1)}(a)$. This figure illustrates the value of the transformation: The true function is close to linear, and so the linear approximation is almost indistinguishable from the true function except at the very lowest values of a.

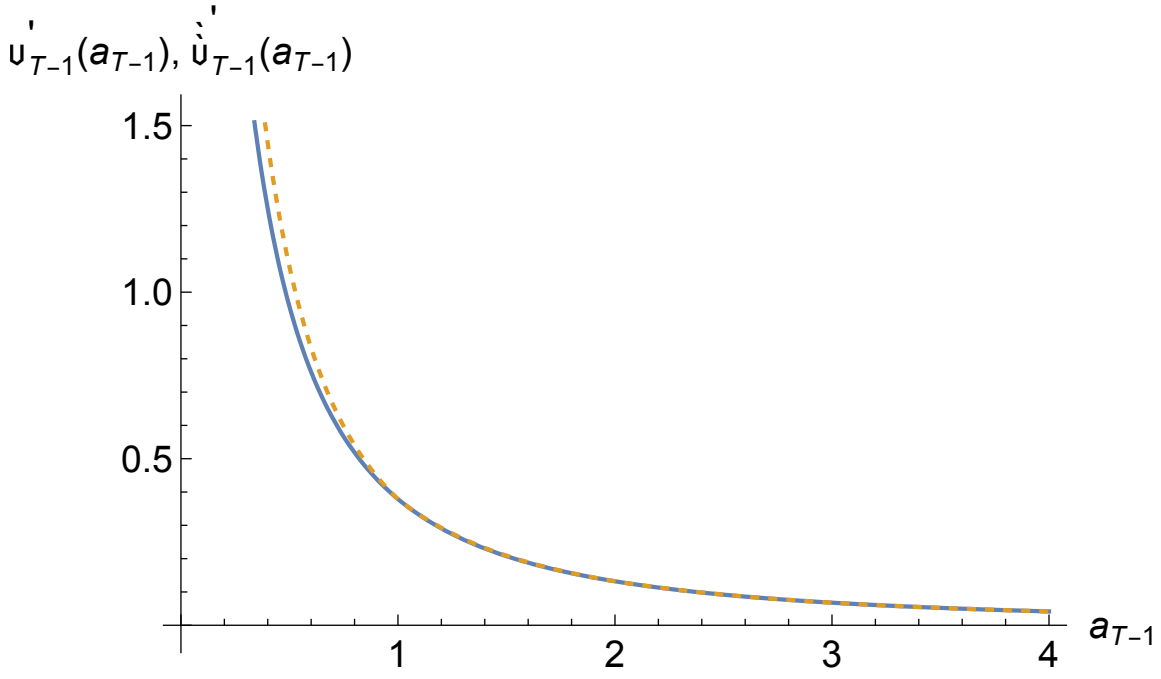
Figure 10 similarly shows that when we generate $\grave{\mathfrak{v}}^{\partial}_{\succ(t-1)}(a)$ using our augmented $[\grave{\mathfrak{c}}_{\succ(t-1)}(a)]^{-\rho}$ (dashed line) we obtain a *much* closer approximation to the true marginal value function $\mathfrak{v}^{\partial}_{\succ(t-1)}(a)$ (solid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

1.9 The Method of Endogenous Gridpoints ('EGM')

subsec:egm}

The solution procedure above for finding $\mathfrak{c}_{t-1}(m)$ still requires us, for each point in $\mathbf{m}_{t-1}$, to use a numerical rootfinding algorithm to search for the value of c that solves $\mathfrak{u}^{c}(c) = \mathfrak{v}^{\partial}_{\succ(t-1)}(m-c)$. Though sections 1.7 and 1.8 developed a highly efficient and accurate procedure to calculate $\grave{\mathfrak{v}}^{\partial}_{\succ(t-1)}$, those approximations do nothing to eliminate

[7] The vertical axis label uses $\mathfrak{v}'$ as an alternative notation for what in these notes we designate as $\mathfrak{v}^{\partial}_{\succ(t-1)}$.



VSGothCInv}

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

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 $\mathbf{a} = \mathbf{a}_{t-1}$, the corresponding $\mathbf{c}$ values for which this $\mathbf{a}$ is optimal.

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

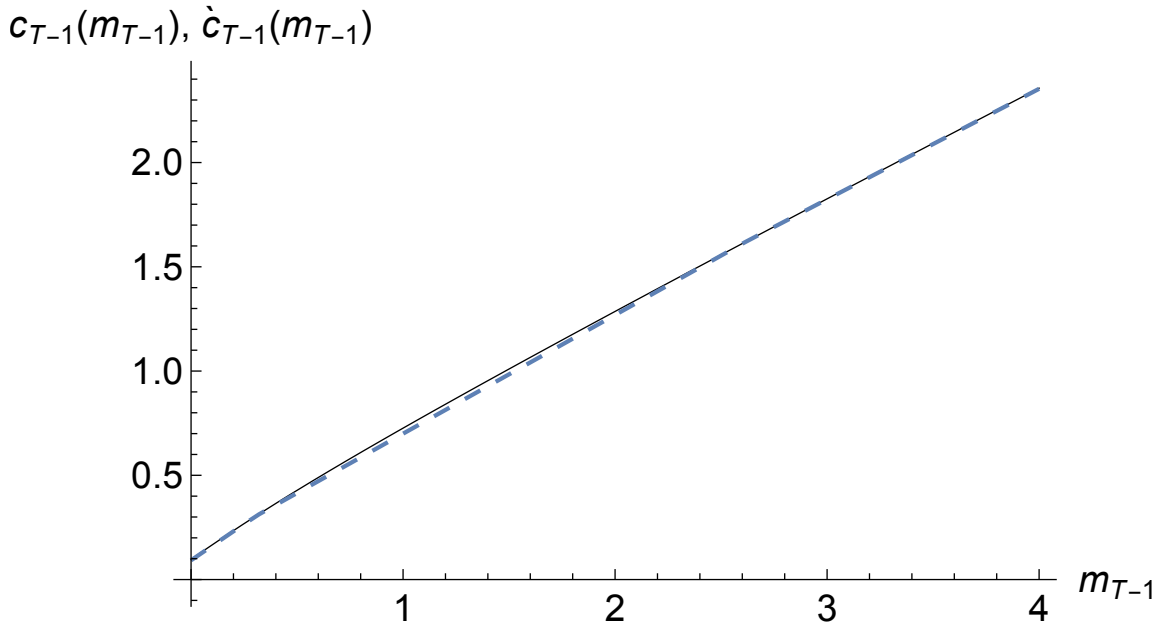
$$\mathbf{m}_{t-1} = \mathbf{c}_{t-1} + \mathbf{a}_{t-1}. \tag{20}$$

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

This routine is performed in the "Endogenous Gridpoints" section of the notebook. First, the `endOfPrd.cCntn_Tm1` function is called for each of the pre-specified 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 $\mathbf{m}_{t-1}$ and $\mathbf{c}_{t-1}$ pairs that can be interpolated between in order to yield $\grave{c}_{\sim(t-1)}(m)$ at virtually zero computational cost![8]

One might worry about whether the $\{m, c\}$ points obtained in this way will provide

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



parecTm1AD}

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

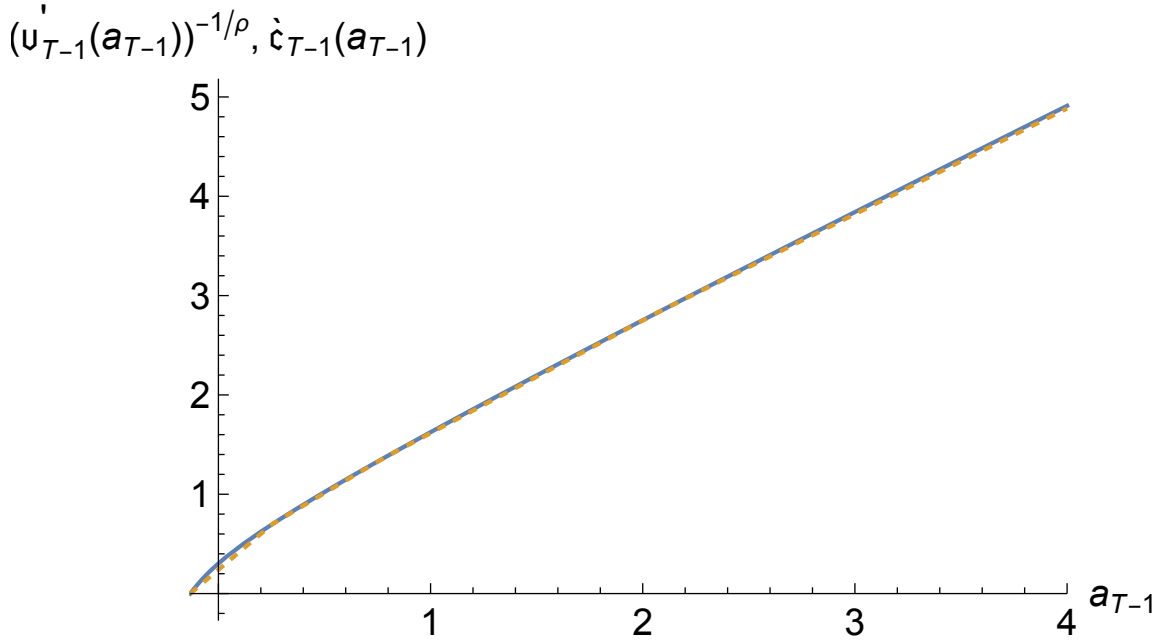
a good representation of the consumption function as a whole, but in practice there are good reasons why they work well (basically, this procedure generates a set of gridpoints that is naturally dense right around the parts of the function with the greatest non-linearity). Figure 11 plots the actual consumption function c_{t-1} and the approximated consumption function $\grave{c}_{t-1}$ derived by the method of endogenous grid points. Compared to the approximate consumption functions illustrated in Figure 8, $\grave{c}_{t-1}$ is quite close to the actual consumption function.

1.10 Improving the a Grid

the-a-grid}

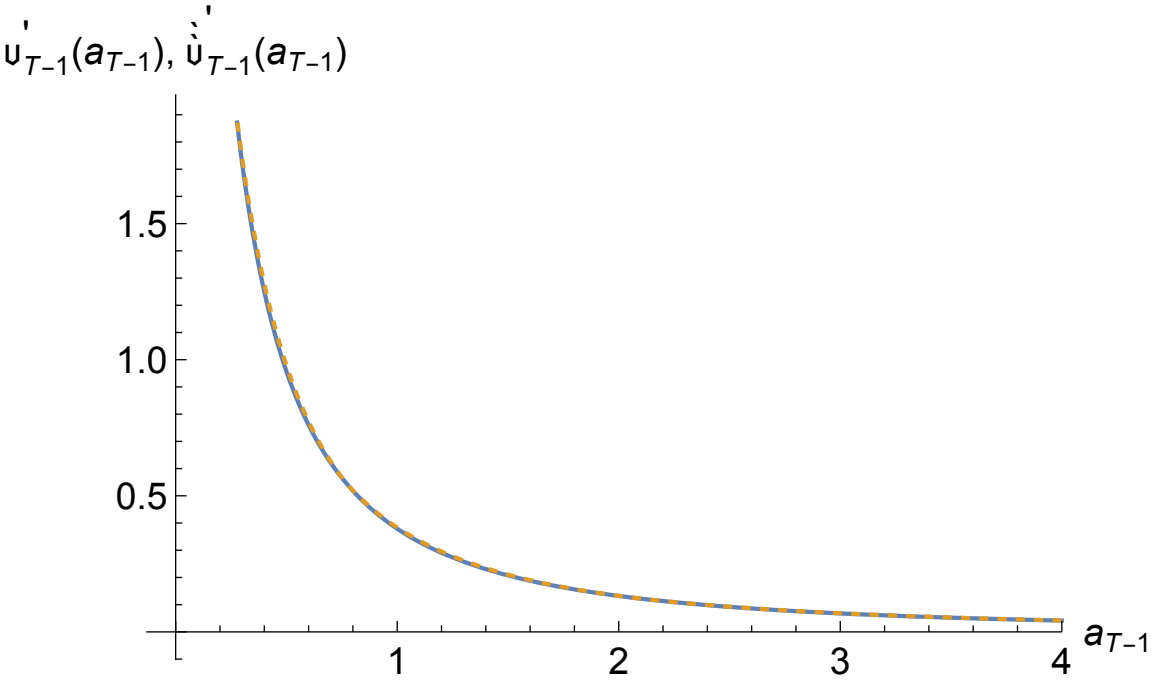
Thus far, we have arbitrarily used a gridpoints of $\{0., 1., 2., 3., 4.\}$ (augmented in the last subsection by $\underline{a}_{t-1}$). But it has been obvious from the figures that the approximated $\grave{c}_{\succ(t-1)}$ function tends to be farthest from its true value at low values of a. Combining 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).



vVSGothCEE}

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



GothCInvEE}

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

Section "Improve the *agrid*" 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.

1.11 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.

In the terminology of section 4.3, each iteration of this backward loop is an invocation of the backward builder $\mathtt{B}^{\leftarrow}_{\text{prd}}$: it creates the Connector (inserted into `P` between the new and existing period solutions), uses it and the already-solved next period to perform the creation of $\mathrm{v}_{\succ(t)}$, and solves for the optimal consumption rule. The dictionary `cFunc_life` is the computational embodiment of `P`—the growing structure of solved periods and Connectors between them assembled by backward induction.

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.

1.12 Results

The code creates the relevant $\grave{\mathrm{c}}_t(m)$ functions for any period in the horizon, at the given values of m. Figure 14 shows $\grave{\mathrm{c}}_{T-n}(m)$ 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.

The construction of $\mathtt{B}^{\leftarrow}_{\text{prd}}$ in this single-stage case uses the builders and connectors of subsection 4.3.

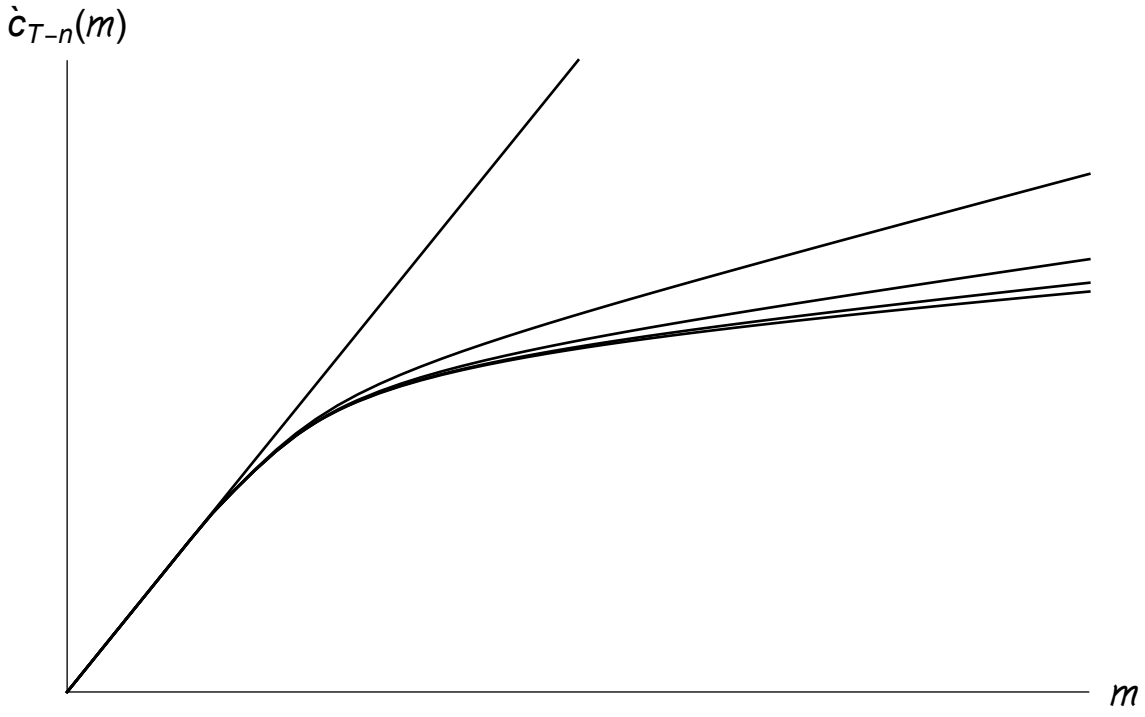


Figure 14 Converging $\grave{c}_{T-n}(m)$ Functions as n Increases