Conventions for this document:

1. A 'stage' has at most three 'moments':
   a) `EOP` is the end of the stage, when (e.g.) $\bar{v}$ =`v_of_a` is defined
   b) `MOP` is middle of stage, when (e.g.) $v(m)$ =`v_of_m` is defined
   c) `BOP` is beginning of stage, when (e.g.) $\underline{v}(k)$ =`v_of_k` is defined

   and at

1 The Problem

We start by assuming we are solving a problem exactly like the one in the BufferStock-Theory (in the 'normalized' section).

$$
\begin{aligned}
\mathrm{v}_t(m_t) = \max_{\{\mathrm{c}\}_t^T} \; & \mathrm{u}(\mathrm{c}_t) + \beta\, \mathbb{E}_t[G_{t+1}^{1-\gamma} \mathrm{v}_{t+1}(m_{t+1})] \\
\text{s.t.} & \\
a_t &= m_t - c_t \\
k_{t+1} &= a_t / G_{t+1} \\
b_{t+1} &= k_{t+1} R = (R/G_{t+1}) a_t \;\; = \;\; \tilde{R}_{t+1} a_t \\
m_{t+1} &= b_{t+1} + \xi_{t+1},
\end{aligned}
$$

where $\tilde{R}_{t+1} \equiv (R/G_{t+1})$ is a 'permanent-income-growth-normalized' return factor.

2 Details of the Solution Procedure

We will be assuming that the key elements we need in order to solve the model are laid out in Dol[o/ARK] model files.

`terminal.yaml`–	defines $\underline{v}(k) =$ `terminal.v_of_k` $= 0$
`stage_opt_cns.yaml`–	defines full solution to consumption problem
`stage_exp_val.yaml`–	defines full solution to consumption problem

and that we can retrieve elements of those files at will. For example, the assumption (likely incorrect at present) is that we can retrieve the reward function from `terminal` using the syntax `terminal.reward`.

The solution can be built backwards as follows. (The first column is the 1-indexed number of steps back from from the end of the problem. That is, the 1 in line 1 says that the first step sets the value function to $\bar{v}(a) = 0$, because we are assuming there is no bequest motive. For reasons that will become evident later, it will be useful to break the problem down into two sub-stages:

`term_v_of_m.yaml`– defines $\underline{v}(k) =$ `terminal.v_of_m` $= 0$

	date	Equation	`step_type`	origin
1	T	$\bar{v}(a) = 0$	`terminal.v_of_a`	defined
2	T	$u(c) =$	`terminal.reward`	defined
3	T	$\mathrm{c}(m) = m$	`terminal.decision`	constructed
4	T	$v(m) = u(\mathrm{c}(m)) + \beta\bar{v}(a)$	`terminal.v_of_m`	defined

`term_v_of_k.yaml`– defines $\underline{v}(k) =$ `terminal.v_of_m` $= 0$

	date	Equation	`step_type`	origin
5	T	shocks:$\{\psi, \xi, \mathbf{R}\}$	`terminal.exogenous`	defined
6	T	$\underline{v}(k) = \mathbb{E}[v(m)]$	`terminal.v_of_k`	constructed

The transition between period $T-1$ and period T is

	date(s)	Equation	`step_type`
7	$T-1 \leftrightarrow T$	$k_{t+1} = a_t$	`transition_state`
8	$T-1 \leftrightarrow T$	$\bar{v}_t(a_t) = \underline{v}_{t+1}(k_{t+1})$	`transition_value`

We would again define the problem in period $T-1$ as having two substages:
`cstage_v_of_m`:

	date(s)	Equation	`step_type`	comment
9	$T-1$	$\beta =$	`stage_opt_cns.DiscFac`	
10	$T-1$	$u(c) =$	`stage_opt_cns.reward`	
11	$T-1$	$\mathfrak{c}(a) =$	`stage_opt_cns.EGM`	consumed
12	$T-1$	$\mathrm{c}(m)$	`stage_opt_cns.decision`	constructed
13	$T-1$	$v(m) = u(\mathrm{c}(m)) + \beta\bar{v}(m - \mathrm{c}(m))$	`stage_opt_cns.v_of_m`	

`cstage_v_of_k`:

	date(s)	Equation	`step_type`	comment
14	$T-1$	shocks	`stage_opt_cns.exogenous`	

15	$T-1$	$\underline{v}(k) = \mathbb{E}[v(m)]$	`stage_opt_cns.expect`

This schema illustrates several points.

1. The choice of whether to identify the 'shocks' as becoming known instantaneously after the beginning of period T or instantaneously before the end of $T-1$ is mathematically and computationally arbitrary.
   a) 'arbitrary' in the sense that is just a question of labeling; the computations are identical whichever scheme is chosen
   b) The scheme above is my preferred, new way of doing it, because we ran into confusions in the old way of doing things where the expectation now taken in step 6 was taken at the end of the prior period (resulting in the infamous Gothic $\mathfrak{v}$).
   c) The old scheme has the advantage (which is also its disadvantage) that the k_{t+1} variable need not be defined. We could just redefine step 8 as $\bar{v}_t(a_t) = \mathbb{E}_t[v_{t+1}(a_{t+1}\mathcal{R}_{t+1} + \theta_{t+1})]$ (where the new notation of $\bar{v}$ is equivalent to the old $\mathfrak{v}(a)$, and $\mathcal{R} = R/(\psi G)$, and eliminate step 7 as superflous.

2. There is a clean separation between the 'transition' phase (lines 7-8, which connects adjacent periods, and the remaining steps (9-15), which build the solution to the problem by executing a series of steps in order. Let's call this collection of steps `stage_opt_cns-solve`.

3. No subscripts are needed for variables used in `stage_opt_cns-solve` because the whole point of declaring this to be period T is that no variable in that namespace can retrieve anything from any other period.

4. What we mainly care about is the consumption function, which is constructed in step 12. The exactly identical numerical consumption function is created regardless of which period we do the expectation calculations in.

Now, if we define `transtage` as comprising steps 7-8, the remainder of the problem is constructed by iterating the two elements: Then the problem in period $T-1$ is like this:

	date(s)	`stage_type`
16	$T-2 \leftrightarrow T-1$	`transtage`
17	$T-2$	`stage_opt_cns-solve`
18	$T-3 \leftrightarrow T-2$	`transtage`
19	$T-3$	`stage_opt_cns-solve`

It would consist of steps like this:

	date(s)	`stage_type`
20	$T-2 \leftrightarrow T-1$	`transtage`
21	$T-2$	`stage_opt_cns-solve`
22	$T-3 \leftrightarrow T-2$	`transtage`
23	$T-3$	`stage_opt_cns-solve`

Now suppose we want to add a portfolio stage to the problem. Specifically, we will assume that right after the beginning of period t, before the shocks are realized, the consumer must make a choice about the proportion ς of capital k to be committed to risky asset that will earn return $\mathbf{R}$ and the proportion $(1-\varsigma)$ that will earn R, for a combined portfolio return of $\mathbb{R} = \varsigma\mathbf{R} + (1-\varsigma)R = R + \varsigma(\mathbf{R} - R)$.

The problem now has an extra stage to it:

$$\underline{v}(k) = \max_{\varsigma} \mathbb{E}[v(\overbrace{(\mathbb{R}/(G\psi))k + \xi}^{\equiv m})]$$

The first order condition for this problem is well-known, and this whole portfolio optimization problem can therefore be bundled up into a `portfolio` stage that takes k as a beginning-of-period input and yields k and ς as outputs.

Thanks to the modularity of the ways in which we have laid out the problem, it will now be possible simply to drop the portfolio stage into the appropriate point in the sequence of steps used earlier to define `stage_opt_cns-solve`. Suppose we call the modified set of steps `stage_opt_cns-with-portfolio-solve`:

backsteps	Equation	step_type	comment
0	$\beta =$	stage_opt_cns.DiscFac	
1	$u(c) =$	stage_opt_cns.reward	
2	$\mathfrak{c}(a) =$	stage_opt_cns.EGM	consumed
3	$\mathrm{c}(m)$	stage_opt_cns.decision	constructed
4	$v(m) = u(\mathrm{c}(m)) + \beta\bar{v}(m - \mathrm{c}(m))$	stage_opt_cns.v_of_m	
5	shocks	stage_opt_cns.exogenous	
6	portfolio		
7	$\underline{v}(k) = \mathbb{E}[v(m)]$	stage_opt_cns.expect	

The beauty of this scheme is that we can now add a portfolio choice wherever we want:

	date(s)	stage_type
24	$T-3 \leftrightarrow T-4$	transtage
25	$T-4$	stage_opt_cns-with-portfolio-solve
26	$T-4 \leftrightarrow T-5$	transtage
27	$T-5$	stage_opt_cns-with-portfolio-solve
28	$T-5 \leftrightarrow T-6$	transtage
29	$T-6$	stage_opt_cns-solve

This sequence would define a problem in which the consumer has no portfolio choice in periods T through $T-3$ but then has a portfolio choice in periods $T-4$ and $T-5$, followed by a $T-6$ with no portfolio choice again. It is easy to seen now how this way of doing things allows us modularly to add as many stages as we like to a particular period. (Each 'stage' should be a simple problem disciplined by the requirements of a Dol[o/ARK] yaml file; but we can string together as may such stages as we like, as long as the requirements of the successive stages are mutually satisfied).

Finally, notice that if we were to say that the job of the user of the toolkit is to provide an algorithm for the construction of the prior period, given the existing length of the problem, we sidestep all the complicated questions about defining in advance what is time varying and what is time invariant, etc. Subject to the constraint of compatibility of the transition process between t and $t-1$ the user has complete freedom to rewrite anything at all as the stage that comes before all the prior stages. They can set the interest factor, the time preference rate, beliefs about the magnitude of stock market fluctuations, to whatever they like. The constructive machinery would record the assumptions that would be made, and those records would define the sequence of values of (potentially) time-varying objects.

Notice further how easy it is to add a discrete choice component to this. Suppose the problem is one of durable good adjustment (buy/sell my car). Call the two options "stay" and "move".

	date(s)	stage_type
30	$T-5 \leftrightarrow T-6$	transtage
31	$T-6$	{stagemove,stagestay}

Where `choose-move-or-stay` just decides, for each configuration of state variables, which option yields the highest value.

This is how we should have done things from the start.