The solution can be built backwards as follows. (The first column is the 1-indexed number of steps back from from the

There is a clean separation between the 'transition' phase (lines -, which connects adjacent periods, and the remaining steps
No subscripts are needed for variables used in `stage_opt_cns-solve` because the whole point of declaring this to be period
What we mainly care about is the consumption function, which is constructed in step . The exactly identical numerical con

0 $\beta =$ `stage_opt_cns.DiscFac`
1 $u(c) =$ `stage_opt_cns.reward`
2 $c(a) =$ `stage_opt_cns.EGM` *consumed*
3 (m)`stage_opt_cns.decision` *constructed*
4 $v(m) = u((m)) + \beta\bar{v}(m - (m))$`stage_opt_cns.v_of_m`
5 *shocks*`stage_opt_cns.exogenous`
6 `portfolio`
7 $v(k) = [v(m)]$`stage_opt_cns.expect`

The beauty of this scheme is that we can now add a portfolio choice wherever we want: 3 date(s) `stage_type`

T-3$\leftrightarrow T-4$ `transtage`
$T-4$ `stage_opt_cns-with-portfolio-solve`
$T-4 \leftrightarrow T-5$ `transtage`
$T-5$ `stage_opt_cns-with-portfolio-solve`
$T-5 \leftrightarrow T-6$ `transtage`
$T-6$ `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 th

Finally, notice that if we were to say that the job of the user of the toolkit is to provide an algorithm for the constructi

Notice further how easy it is to add a discrete choice component to this. Suppose the problem is one of durable good a

3 date(s) `stage_type`

T-5$\leftrightarrow T-6$ `transtage`
$T-6$ $\{$`stagemove`, `stagestay`$\}$
$T-6$ `choose-move-or-stay`

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

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