FF_VFI_AZ_VEC Dynamic Savings Problem Vectorized Common Grid

back to Fan's Intro Math for Econ, Matlab Examples, or Dynamic Asset Repositories

This is the example vignette for function: **ff_vfi_az_vec** from the **MEconTools Package.** This function solves (vectorized) the dynamic programming problem for a (a,z) model. Households can save a, and face AR(1) shock z. The problem is solved over the infinite horizon.

The code uses common grid, with the same state space and choice space grids. **ff_vfi_az_bisec_vec** from the **MEconTools Package** solves the same problem but using continuous exact percentage asset choices, which is more precise than the solution here, and perhaps a little bit slower.

This is the vectorized code, its speed is much faster than the looped code. The function is designed to have small memory footprint and requires low computing resources, yet is fast.

Links to Four Code:

Four Core Savings/Borrowing Dynamic Programming Solution Functions that are functions in the **MEconTools Package.** :

- Common Choice and States Grid ***Loop***: **ff_vfi_az_loop**, slow should use for testing new models
- Common Choice and States Grid ***Vectorized***: **ff_vfi_az_vec**, fast good for many purposes
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand ***Loop***: **ff_vfi_az_bisec_loop**, high precision even with small grid
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand ***Vectorized***: **ff_vfi_az_bisec_vec**, precision and speed

Test FF_VFI_AZ_VEC Defaults

Call the function with defaults. By default, shows the asset policy function summary. Model parameters can be changed by the mp_params.

```
%mp_params
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('fl_crra') = 1.5;
mp_params('fl_beta') = 0.94;
ff_vfi_az_vec(mp_params);
```

```
Elapsed time is 0.128918 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
          i    idx    ndim    numel    rowN    colN     sum      mean      std     coefvari    min    max
          _    ___    ____    _____    ____    ____    _____    ______    _____    ________    ___    ___

    ap    1     1      2       700     100      7      16864    24.091    14.08    0.58446      0     50

xxx TABLE:ap xxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5         c6         c7
          _______    _______    _______    _______    _______    _______    _____
```

r1	0	0	0	0	0	0.50505	2.0202
r2	0	0	0	0.50505	0.50505	1.0101	2.5253
r3	0.50505	0.50505	0.50505	0.50505	1.0101	1.5152	3.0303
r4	1.0101	1.0101	1.0101	1.0101	1.5152	2.0202	3.5354
r5	1.5152	1.5152	1.5152	1.5152	2.0202	2.5253	4.0404
r96	45.455	45.455	45.96	45.96	45.96	46.97	48.485
r97	45.96	45.96	45.96	46.465	46.465	47.475	48.99
r98	46.465	46.465	46.465	46.97	46.97	47.98	48.99
r99	46.97	46.97	46.97	47.475	47.475	48.485	49.495
r100	47.475	47.475	47.475	47.98	47.98	48.99	50

Test FF_VFI_AZ_BISEC_VEC Speed Tests

Call the function with different a and z grid size, print out speed:

```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_timer') = true;
mp_support('ls_ffcmd') = {};
```

A grid 200, shock grid 9:

```
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 200;
mp_params('it_z_n') = 9;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.220867 seconds.
```

A grid 750, shock grid 15:

```
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 750;
mp_params('it_z_n') = 15;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 3.573648 seconds.
```

A grid 600, shock grid 45:

```
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 600;
mp_params('it_z_n') = 45;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 8.398580 seconds.
```

Test FF_VFI_AZ_VEC Control Outputs

Run the function first without any outputs;

```
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 50;
mp_params('it_z_n') = 5;
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_timer') = false;
mp_support('bl_print_params') = false;
```

```
mp_support('bl_print_iterinfo') = false;
```

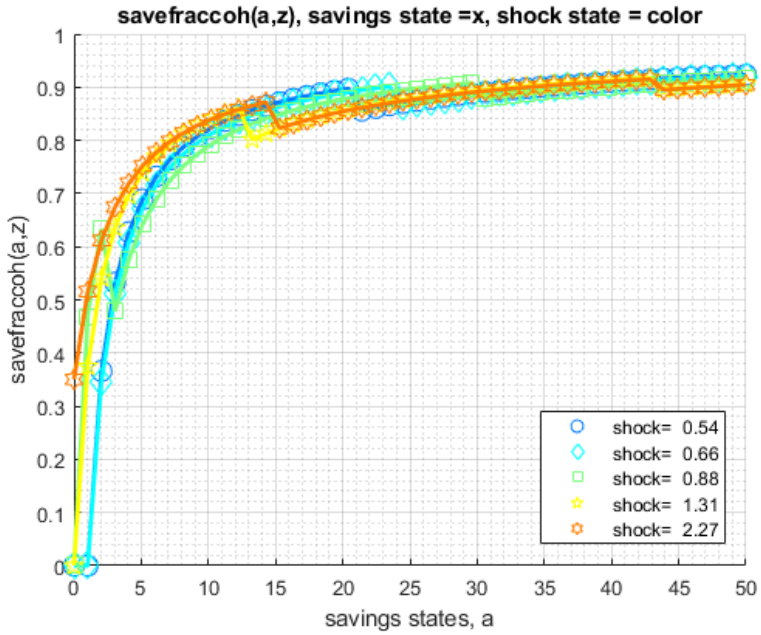
Run the function and show policy function for savings choice. For ls_ffcmd, ls_ffsna, ls_ffgrh, can include these: 'v', 'ap', 'c', 'y', 'coh', 'savefraccoh'. These are value, aprime savings choice, consumption, income, cash on hand, and savings fraction as cash-on-hand.

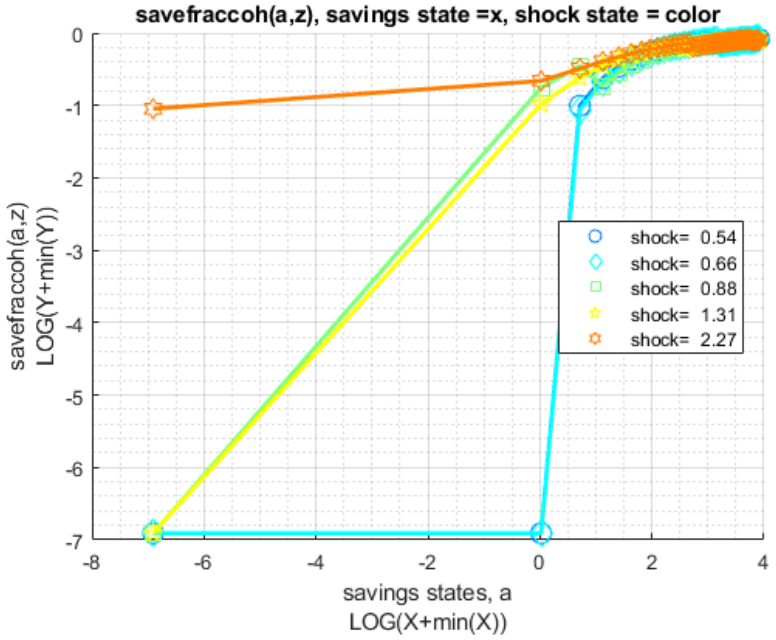
```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
% ls_ffcmd: summary print which outcomes
mp_support('ls_ffcmd') = {};
% ls_ffsna: detail print which outcomes
mp_support('ls_ffsna') = {'savefraccoh'};
% ls_ffgrh: graphical print which outcomes
mp_support('ls_ffgrh') = {'savefraccoh'};
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.014484 seconds.
xxx  ff_vfi_az_vec, outcome=savefraccoh  xxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	a	mean_z_0_54195	mean_z_0_66401	mean_z_0_88162	mean_z_1_3095	mean_z_2_2745
1	0	0	0	0	0	0.3505
2	1.0204	0	0	0.46928	0.37487	0.51572
3	2.0408	0.36632	0.34687	0.63373	0.54163	0.61186
4	3.0612	0.53265	0.51178	0.47837	0.63592	0.67476
5	4.0816	0.62764	0.60816	0.57627	0.69655	0.71911
6	5.102	0.68908	0.67137	0.64196	0.73882	0.75206
7	6.1224	0.73208	0.71603	0.68909	0.76996	0.77751
8	7.1429	0.76386	0.74926	0.72456	0.79387	0.79776
9	8.1633	0.7883	0.77494	0.75221	0.81279	0.81425
10	9.1837	0.80769	0.79539	0.77438	0.82815	0.82795
11	10.204	0.82343	0.81206	0.79254	0.84086	0.8395
12	11.224	0.83648	0.82591	0.8077	0.85155	0.84937
13	12.245	0.84747	0.83759	0.82053	0.86067	0.85791
14	13.265	0.85685	0.84758	0.83155	0.80173	0.86537
15	14.286	0.86495	0.85622	0.8411	0.81288	0.87194
16	15.306	0.87201	0.86377	0.84947	0.82268	0.82291
17	16.327	0.87823	0.87043	0.85685	0.83137	0.83104
18	17.347	0.88374	0.87633	0.86342	0.83912	0.83834
19	18.367	0.88866	0.88161	0.8693	0.84608	0.84495
20	19.388	0.89309	0.88635	0.8746	0.85237	0.85095
21	20.408	0.89708	0.89064	0.87939	0.85807	0.85642
22	21.429	0.85567	0.89454	0.88375	0.86327	0.86143
23	22.449	0.86096	0.89809	0.88773	0.86803	0.86604
24	23.469	0.86581	0.90135	0.89138	0.87241	0.87029
25	24.49	0.87026	0.86502	0.89474	0.87644	0.87422
26	25.51	0.87436	0.8693	0.89784	0.88017	0.87787
27	26.531	0.87816	0.87327	0.90071	0.88362	0.88126
28	27.551	0.88168	0.87695	0.90338	0.88684	0.88443
29	28.571	0.88496	0.88037	0.90586	0.88984	0.88739
30	29.592	0.88802	0.88357	0.90818	0.89264	0.89017
31	30.612	0.89087	0.88655	0.87896	0.89527	0.89278
32	31.633	0.89355	0.88935	0.88197	0.89773	0.89523
33	32.653	0.89606	0.89198	0.8848	0.90005	0.89754
34	33.673	0.89843	0.89446	0.88747	0.90223	0.89972
35	34.694	0.90065	0.89679	0.88998	0.90429	0.90178
36	35.714	0.90275	0.89899	0.89235	0.90624	0.90374
37	36.735	0.90474	0.90107	0.8946	0.90809	0.90559
38	37.755	0.90662	0.90304	0.89673	0.90984	0.90735
39	38.776	0.90841	0.90491	0.89874	0.9115	0.90902

40	39.796	0.9101	0.90669	0.90066	0.91308	0.91062
41	40.816	0.91171	0.90838	0.90249	0.91458	0.91214
42	41.837	0.91325	0.90998	0.90422	0.91601	0.91359
43	42.857	0.91471	0.91152	0.90588	0.91738	0.91497
44	43.878	0.9161	0.91298	0.90746	0.89681	0.89499
45	44.898	0.91743	0.91438	0.90897	0.89854	0.89671
46	45.918	0.91871	0.91571	0.91042	0.90019	0.89836
47	46.939	0.91993	0.91699	0.91181	0.90178	0.89994
48	47.959	0.9211	0.91822	0.91313	0.9033	0.90146
49	48.98	0.92222	0.91939	0.91441	0.90475	0.90292
50	50	0.92329	0.92052	0.91563	0.90615	0.90433





Run the function and show summaries for savings and fraction of coh saved:

```
mp_params('it_a_n') = 100;
mp_params('it_z_n') = 9;
mp_support('ls_ffcmd') = {'ap', 'savefraccoh'};
mp_support('ls_ffsna') = {};
```

```
mp_support('ls_ffgrh') = {};
mp_support('bl_vfi_store_all') = true; % store c(a,z), y(a,z)
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.127807 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min
ap	1	1	2	900	100	9	21825	24.25	14.089	0.581	0
savefraccoh	2	2	2	900	100	9	752.38	0.83597	0.13497	0.16145	0

xxx TABLE:ap xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0	0.50505	1.5152	3.0303
r2	0	0	0	0	0.50505	0.50505	1.0101	1.5152	3.5354
r3	0.50505	0.50505	0.50505	0.50505	0.50505	1.0101	1.5152	2.0202	4.0404
r4	1.0101	1.0101	1.0101	1.0101	1.0101	1.5152	2.0202	2.5253	4.5455
r5	1.5152	1.5152	1.5152	1.5152	1.5152	2.0202	2.5253	3.0303	5.0505
r96	45.455	45.455	45.455	45.96	45.96	45.96	46.465	47.475	49.495
r97	45.96	45.96	45.96	46.465	46.465	46.465	46.97	47.98	49.495
r98	46.465	46.465	46.465	46.465	46.97	46.97	47.475	48.485	50
r99	46.97	46.97	46.97	46.97	47.475	47.475	47.98	48.99	50
r100	47.475	47.475	47.475	47.475	47.98	47.98	48.485	49.495	50

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0	0.24587	0.48182	0.56208
r2	0	0	0	0	0.3075	0.25444	0.39276	0.41371	0.59831
r3	0.30679	0.29486	0.27938	0.25939	0.2338	0.40362	0.49043	0.4833	0.6287
r4	0.4668	0.45285	0.43438	0.40981	0.37721	0.50166	0.56006	0.53755	0.65456
r5	0.56502	0.55132	0.53293	0.50802	0.47415	0.57101	0.61221	0.58103	0.67683
r96	0.91292	0.9117	0.90997	0.91752	0.91364	0.90746	0.90692	0.90732	0.90699
r97	0.91357	0.91236	0.91064	0.91812	0.91427	0.90815	0.90761	0.90799	0.89847
r98	0.9142	0.913	0.9113	0.90882	0.91489	0.90882	0.90828	0.90865	0.89919
r99	0.91482	0.91363	0.91195	0.90949	0.91549	0.90949	0.90894	0.90929	0.89089
r100	0.91543	0.91425	0.91258	0.91014	0.91609	0.91013	0.90959	0.90992	0.88275

Test FF_VFI_AZ_VEC Change Interest Rate and Discount

Show only save fraction of cash on hand:

```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
mp_support('ls_ffcmd') = {'savefraccoh'};
mp_support('ls_ffsna') = {};
mp_support('ls_ffgrh') = {};
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 750;
mp_params('it_z_n') = 9;
mp_params('fl_a_max') = 50;
mp_params('st_grid_type') = 'grid_powerspace';
```

Solve the model with several different interest rates and discount factor:

```
% Lower Savings Incentives
mp_params('fl_beta') = 0.80;
mp_params('fl_r') = 0.01;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.771613 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN     sum      mean       std      coefvari    min
                  _    ___    ____    _____    ____    ____    ______    _______    _______    ________    ___

    savefraccoh   1     1      2       6750     750      9      3318.8    0.49167    0.27768    0.56477      0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5         c6         c7         c8         c9
          _______    _______    _______    _______    _______    _______    _______    _______    _______

    r1          0          0          0          0          0          0    0.023475    0.13289    0.29755
    r2          0          0          0          0          0          0    0.023475    0.13289    0.29755
    r3          0          0          0          0          0          0    0.023475    0.13289    0.29755
    r4          0          0          0          0          0          0    0.023475    0.13289    0.29755
    r5          0          0          0          0          0          0    0.023475    0.13289    0.29755
    r746   0.8044    0.80333    0.80182    0.79961    0.79626    0.79093     0.7887     0.7824    0.77641
    r747  0.80465    0.80359    0.80209    0.79989    0.79655    0.79124    0.78903    0.78277    0.77686
    r748  0.80491    0.80385    0.80235    0.80016    0.79683    0.79154    0.78936    0.78315    0.77731
    r749  0.80517    0.80411    0.80262    0.80043    0.79712    0.79185    0.78969    0.78352    0.77776
    r750  0.80542    0.80437    0.80288    0.80071     0.7974    0.79215    0.79002    0.78389    0.77821
```

```
% Higher Savings Incentives
mp_params('fl_beta') = 0.95;
mp_params('fl_r') = 0.04;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 2.484993 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN     sum      mean       std      coefvari    min
                  _    ___    ____    _____    ____    ____    ______    _______    _______    ________    ___

    savefraccoh   1     1      2       6750     750      9      4491.9    0.66547    0.28771    0.43234      0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
             c1         c2         c3         c4         c5         c6         c7         c8         c9
          _______    _______    _______    _______    _______    _______    _______    _______    _______

    r1          0          0          0          0    0.031818    0.14726    0.31047    0.48484    0.64489
    r2          0          0          0          0    0.031818    0.14726    0.31047    0.48484    0.64489
    r3          0          0          0          0    0.031818    0.14726    0.31047    0.48484    0.64489
    r4          0          0          0          0    0.031818    0.14726    0.31047    0.48484    0.64489
    r5          0          0          0          0    0.031818    0.14726    0.31047    0.48484    0.64489
    r746  0.92742       0.93     0.9283    0.92581     0.92578    0.92349    0.92443    0.91686    0.88398
    r747   0.9275    0.93007    0.92838     0.9259     0.92588    0.92361    0.92457    0.91706    0.88076
    r748  0.92757    0.93014    0.92846    0.92599     0.92598    0.92373    0.92472    0.91359    0.87757
    r749  0.92764    0.93022    0.92854    0.92608     0.92608    0.92384    0.92115    0.91014    0.87438
    r750  0.92772    0.93029    0.92862    0.92617     0.92618    0.92396     0.9213    0.90671    0.87121
```

Test FF_VFI_AZ_VEC Changing Risk Aversion

Here, again, show fraction of coh saved in summary tabular form, but also show it graphically.

```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
mp_support('ls_ffcmd') = {'savefraccoh'};
mp_support('ls_ffsna') = {};
mp_support('ls_ffgrh') = {'savefraccoh'};
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 750;
mp_params('it_z_n') = 9;
mp_params('fl_a_max') = 50;
mp_params('st_grid_type') = 'grid_powerspace';
```

Solve the model with different risk aversion levels, higher preferences for risk:

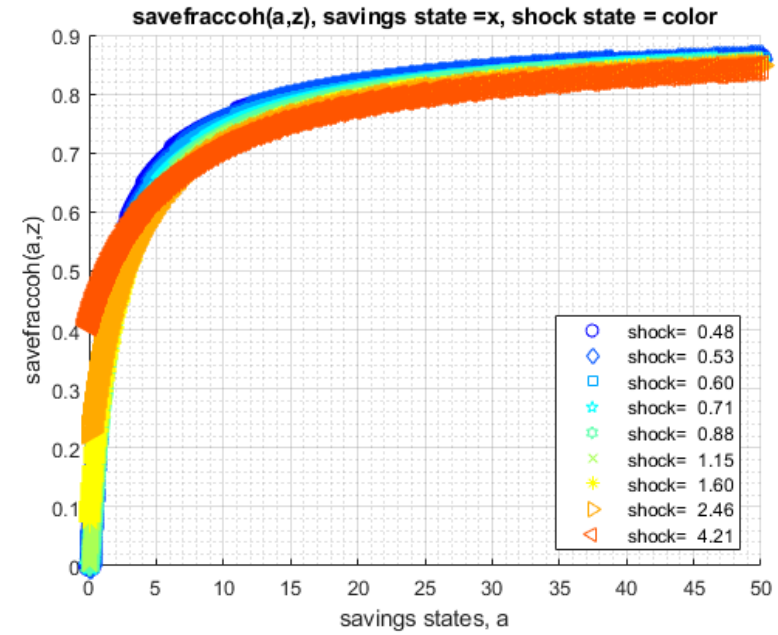
```
% Lower Risk Aversion
mp_params('fl_crra') = 0.5;
ff_vfi_az_vec(mp_params, mp_support);
```

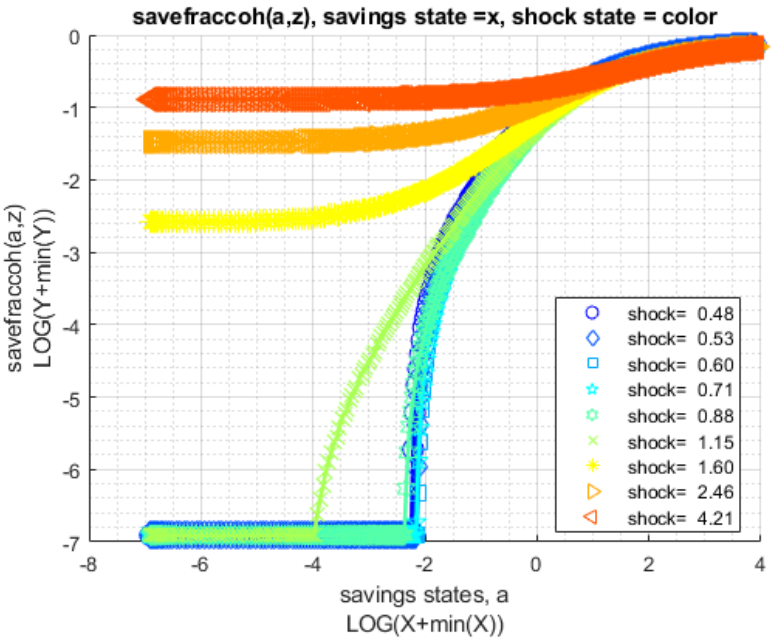
```
Elapsed time is 1.991475 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN     sum       mean       std       coefvari    min
                  _    ___    ____    _____    ____    ____    ______    _______    ______    ________    ___

    savefraccoh   1     1       2      6750     750      9      3735.9    0.55347    0.2897    0.52343      0
```

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
              c1         c2         c3         c4         c5         c6         c7         c8         c9
            _______    _______    _______    _______    _______    _______    ________    _______    _______

    r1            0          0          0          0          0          0    0.075021    0.22812    0.41075
    r2            0          0          0          0          0          0    0.075021    0.22812    0.41075
    r3            0          0          0          0          0          0    0.075021    0.22812    0.41075
    r4            0          0          0          0          0          0    0.075021    0.22812    0.41075
    r5            0          0          0          0          0          0    0.075021    0.22812    0.41075
    r746    0.85928    0.85816    0.85657    0.85425    0.85428     0.8522     0.84972    0.84635    0.84292
    r747    0.85946    0.85834    0.85676    0.85444    0.85449    0.85242     0.84997    0.84665     0.8433
    r748    0.85963    0.85852    0.85694    0.85464    0.85469    0.85264     0.85021    0.84694    0.84368
    r749    0.85981     0.8587    0.85713    0.85483    0.85489    0.85286     0.85046    0.84723    0.84405
    r750    0.85998    0.85888    0.85731    0.85502    0.85509    0.85307      0.8507    0.84752    0.84442
```





When risk aversion increases, at every state-space point, the household wants to save more.

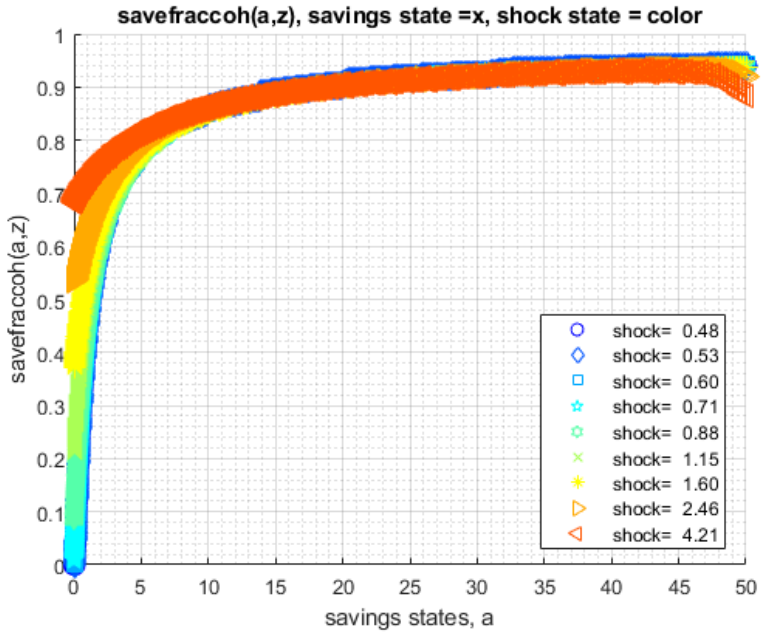
```
% Higher Risk Aversion
mp_params('fl_crra') = 5;
ff_vfi_az_vec(mp_params, mp_support);
```

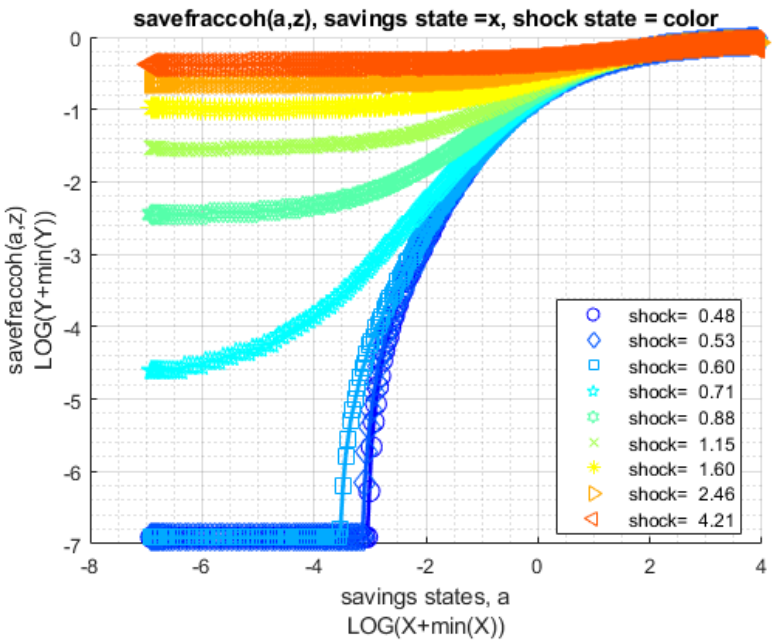
```
Elapsed time is 2.026442 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	m
savefraccoh	1	1	2	6750	750	9	4639.3	0.6873	0.28204	0.41036	0	0.

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0.008995	0.085095	0.21314	0.37277	0.53628	0.68332
r2	0	0	0	0.008995	0.085095	0.21314	0.37277	0.53628	0.68332
r3	0	0	0	0.008995	0.085095	0.21314	0.37277	0.53628	0.68332
r4	0	0	0	0.008995	0.085095	0.21314	0.37277	0.53628	0.68332
r5	0	0	0	0.0089949	0.085094	0.21314	0.37277	0.53628	0.68332
r746	0.94083	0.9396	0.94168	0.93912	0.93904	0.94041	0.93743	0.92949	0.89566
r747	0.94091	0.93969	0.94176	0.93921	0.93914	0.93674	0.93758	0.92969	0.89241
r748	0.94098	0.93977	0.94184	0.93931	0.93924	0.93686	0.93772	0.92618	0.88918
r749	0.94106	0.93985	0.94192	0.9394	0.93934	0.93699	0.93787	0.92269	0.88596
r750	0.94113	0.93993	0.942	0.93949	0.93944	0.93711	0.93801	0.91921	0.88275





Test FF_VFI_AZ_VEC with Higher Uncertainty

Increase the standard deviation of the Shock.

```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
mp_support('ls_ffcmd') = {'savefraccoh'};
mp_support('ls_ffsna') = {};
mp_support('ls_ffgrh') = {};
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 750;
mp_params('it_z_n') = 9;
mp_params('fl_a_max') = 50;
mp_params('st_grid_type') = 'grid_powerspace';
```

Lower standard deviation of shock:

```
% Lower Risk Aversion
mp_params('fl_shk_std') = 0.10;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 2.065989 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	m
savefraccoh	1	1	2	6750	750	9	4026	0.59644	0.31533	0.52869	0	0.9

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0.012569	0.062884	0.13754	0.22274
r2	0	0	0	0	0	0.012569	0.062884	0.13754	0.22274
r3	0	0	0	0	0	0.012569	0.062884	0.13754	0.22274
r4	0	0	0	0	0	0.012569	0.062884	0.13754	0.22274
r5	0	0	0	0	0	0.012569	0.062884	0.13754	0.22274
r746	0.91375	0.91251	0.91101	0.91289	0.91057	0.91138	0.91136	0.91021	0.9075
r747	0.91387	0.91264	0.91114	0.91302	0.91072	0.91153	0.91152	0.91039	0.90769
r748	0.91399	0.91277	0.91127	0.91315	0.91086	0.91168	0.91168	0.91056	0.90788
r749	0.91411	0.91289	0.9114	0.91329	0.911	0.91183	0.91183	0.91073	0.90807
r750	0.91423	0.91302	0.91153	0.91342	0.91114	0.91197	0.91199	0.9109	0.90826

Higher shock standard deviation: low shock high asset save more, high shock more asset save less, high shock low asset save more:

```
% Higher Risk Aversion
mp_params('fl_shk_std') = 0.40;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 2.184888 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	
savefraccoh	1	1	2	6750	750	9	4687.4	0.69442	0.27109	0.39038	0	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0.030619	0.24561	0.55369	0.80189	0.46265
r2	0	0	0	0	0.030619	0.24561	0.55369	0.80189	0.46265
r3	0	0	0	0	0.030619	0.2456	0.55369	0.80189	0.46265
r4	0	0	0	0	0.030619	0.2456	0.55369	0.80189	0.46265
r5	0	0	0	0	0.030618	0.2456	0.55369	0.80189	0.46265
r746	0.93365	0.93335	0.9328	0.93173	0.92941	0.92713	0.92079	0.8402	0.31544
r747	0.93371	0.93341	0.93286	0.9318	0.92949	0.92723	0.92095	0.83734	0.31504
r748	0.93378	0.93348	0.93293	0.93187	0.92957	0.92733	0.92111	0.83449	0.31463
r749	0.93384	0.93354	0.933	0.93194	0.92965	0.92743	0.92127	0.83166	0.31423
r750	0.9339	0.9336	0.93306	0.93201	0.92973	0.92753	0.92143	0.82883	0.31383