FF_VFI_AZ_LOOP Dynamic Savings Problem Loop 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_loop** from the **MEconTools Package.** This function solves 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. This is the looped code, it is slow for larger state-space problems. The code uses common grid, with the same state space and choice space grids.

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

The four sample codes are written for the standard dynamic savings problem with AR(1) shock that is one of the core problems introduced in first sessions of graduate Economics courses. The code can be easily adapted to accomand multiple assets, savings and borrowing, discrete and continuous choice, etc. A large proportion of dynamic economic models are based on the underlying structure of solving a model with endogenous states and exogenous shocks, and that is what the (a,z) model does. In general, one should write looped code first to make sure the economics is correct, then vectorized code can be adopted to increase speed.

Test FF_VFI_AZ_LOOP 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;
% call function
ff_vfi_az_loop(mp_params);
```

```
Elapsed time is 1.291175 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 xxxxxxxxxxxxxxxxxx
              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_LOOP 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 50, shock grid 5:

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

```
Elapsed time is 0.223217 seconds.
```

A grid 100, shock grid 7:

```
mp_params = containers.Map('KeyType','char', 'ValueType','any');
mp_params('it_a_n') = 100;
mp_params('it_z_n') = 7;
ff_vfi_az_loop(mp_params, mp_support);
```

```
Elapsed time is 1.284511 seconds.
```

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_loop(mp_params, mp_support);
```

```
Elapsed time is 6.325330 seconds.
```

Test FF_VFI_AZ_LOOP 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;
```

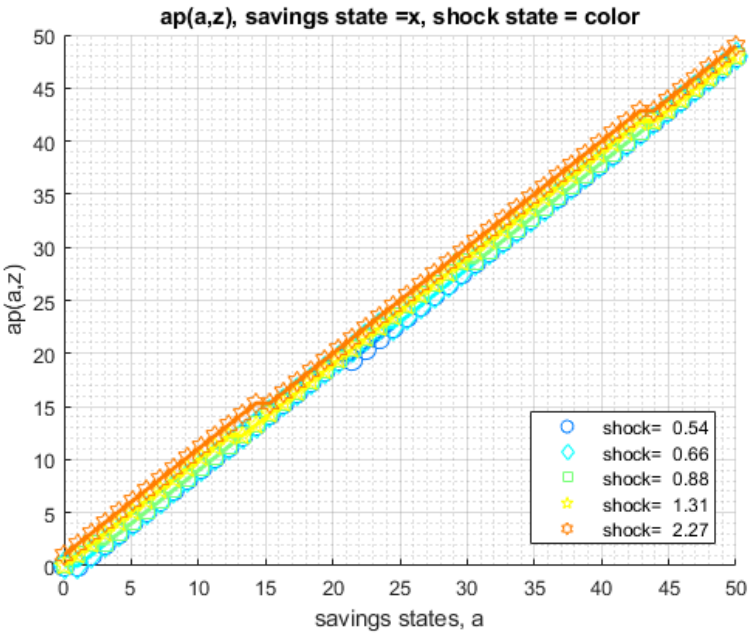
Run the function and show policy function for savings choice. For ls_ffcmd, ls_ffsna, ls_ffgrh, can include these: 'v', 'ap', 'c', 'y', 'coh', 'savefraccoh'. These are value, aprime savings choice, consumption, income, cash on hand, and savings fraction as cash-on-hand.

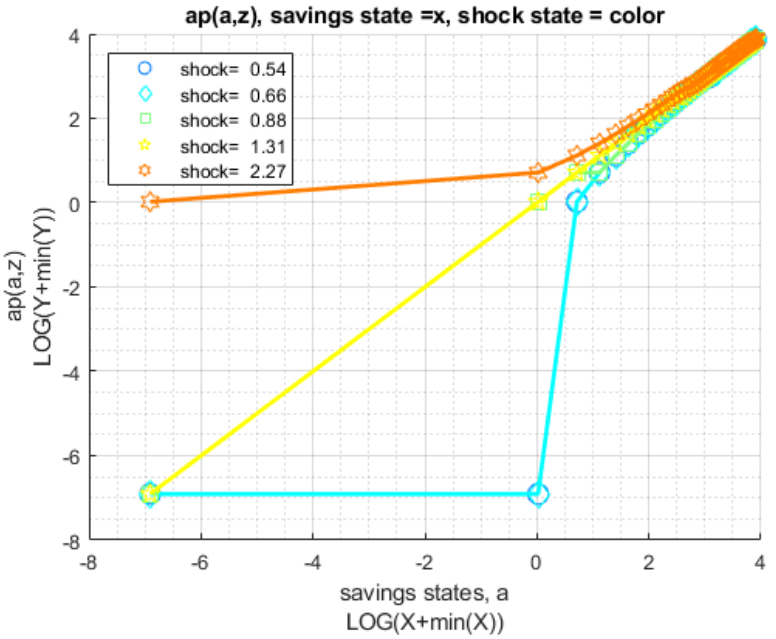
```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
% ls_ffcmd: summary print which outcomes
mp_support('ls_ffcmd') = {};
% ls_ffsna: detail print which outcomes
mp_support('ls_ffsna') = {'ap'};
% ls_ffgrh: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

```
Elapsed time is 0.313830 seconds.
xxx  ff_vfi_az_vec, outcome=ap  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	1.0204
2	1.0204	0	0	1.0204	1.0204	2.0408
3	2.0408	1.0204	1.0204	2.0408	2.0408	3.0612
4	3.0612	2.0408	2.0408	2.0408	3.0612	4.0816
5	4.0816	3.0612	3.0612	3.0612	4.0816	5.102
6	5.102	4.0816	4.0816	4.0816	5.102	6.1224
7	6.1224	5.102	5.102	5.102	6.1224	7.1429
8	7.1429	6.1224	6.1224	6.1224	7.1429	8.1633
9	8.1633	7.1429	7.1429	7.1429	8.1633	9.1837
10	9.1837	8.1633	8.1633	8.1633	9.1837	10.204
11	10.204	9.1837	9.1837	9.1837	10.204	11.224
12	11.224	10.204	10.204	10.204	11.224	12.245
13	12.245	11.224	11.224	11.224	12.245	13.265
14	13.265	12.245	12.245	12.245	12.245	14.286
15	14.286	13.265	13.265	13.265	13.265	15.306
16	15.306	14.286	14.286	14.286	14.286	15.306
17	16.327	15.306	15.306	15.306	15.306	16.327
18	17.347	16.327	16.327	16.327	16.327	17.347
19	18.367	17.347	17.347	17.347	17.347	18.367
20	19.388	18.367	18.367	18.367	18.367	19.388
21	20.408	19.388	19.388	19.388	19.388	20.408
22	21.429	19.388	20.408	20.408	20.408	21.429
23	22.449	20.408	21.429	21.429	21.429	22.449
24	23.469	21.429	22.449	22.449	22.449	23.469
25	24.49	22.449	22.449	23.469	23.469	24.49
26	25.51	23.469	23.469	24.49	24.49	25.51
27	26.531	24.49	24.49	25.51	25.51	26.531
28	27.551	25.51	25.51	26.531	26.531	27.551
29	28.571	26.531	26.531	27.551	27.551	28.571
30	29.592	27.551	27.551	28.571	28.571	29.592
31	30.612	28.571	28.571	28.571	29.592	30.612
32	31.633	29.592	29.592	29.592	30.612	31.633
33	32.653	30.612	30.612	30.612	31.633	32.653
34	33.673	31.633	31.633	31.633	32.653	33.673
35	34.694	32.653	32.653	32.653	33.673	34.694
36	35.714	33.673	33.673	33.673	34.694	35.714
37	36.735	34.694	34.694	34.694	35.714	36.735

38	37.755	35.714	35.714	35.714	36.735	37.755
39	38.776	36.735	36.735	36.735	37.755	38.776
40	39.796	37.755	37.755	37.755	38.776	39.796
41	40.816	38.776	38.776	38.776	39.796	40.816
42	41.837	39.796	39.796	39.796	40.816	41.837
43	42.857	40.816	40.816	40.816	41.837	42.857
44	43.878	41.837	41.837	41.837	41.837	42.857
45	44.898	42.857	42.857	42.857	42.857	43.878
46	45.918	43.878	43.878	43.878	43.878	44.898
47	46.939	44.898	44.898	44.898	44.898	45.918
48	47.959	45.918	45.918	45.918	45.918	46.939
49	48.98	46.939	46.939	46.939	46.939	47.959
50	50	47.959	47.959	47.959	47.959	48.98





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_loop(mp_params, mp_support);
```

```
Elapsed time is 1.867278 seconds.
---------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	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	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_LOOP 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') = 50;
mp_params('it_z_n') = 5;
```

```
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_loop(mp_params, mp_support);
```

```
Elapsed time is 0.113265 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       250      50      5      118.68    0.47472    0.2843    0.59887      0     0.

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
            c1         c2         c3         c4         c5
          _______    _______    _______    _______    _______

    r1          0          0          0          0    0.10642
    r2          0          0          0          0     0.1064
    r3          0          0          0          0    0.10629
    r4          0          0          0          0      0.106
    r5          0          0          0          0    0.10543
    r46   0.79096    0.78787    0.78241    0.77191    0.74922
    r47   0.79553    0.79262    0.78747    0.77755    0.75606
    r48    0.7999    0.79715    0.79229     0.7829    0.76254
    r49   0.80407    0.80147    0.79687    0.78799    0.76868
    r50   0.80805    0.80559    0.80125    0.79284     0.7745
```

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

```
Elapsed time is 0.327279 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       250      50      5      160.99    0.64394    0.29947    0.46506      0     0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
            c1         c2          c3         c4         c5
          _______    _______    ________    _______    _______

    r1          0          0     0.024103    0.18484    0.40057
    r2          0          0     0.024094     0.1848    0.40051
    r3          0          0     0.024028    0.18446    0.40008
    r4          0          0     0.046583    0.18354    0.39894
    r5          0          0     0.045925    0.24935    0.39672
    r46   0.94526    0.94167      0.93533    0.92312    0.89672
    r47   0.94628    0.94291      0.93696    0.92548    0.90059
    r48   0.94722    0.94405      0.93846    0.92766    0.90418
```

r49	0.94808	0.94511	0.93984	0.92966	0.90749
r50	0.94888	0.94608	0.94111	0.93151	0.91056

Test FF_VFI_AZ_LOOP 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') = 100;
mp_params('it_z_n') = 5;
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:

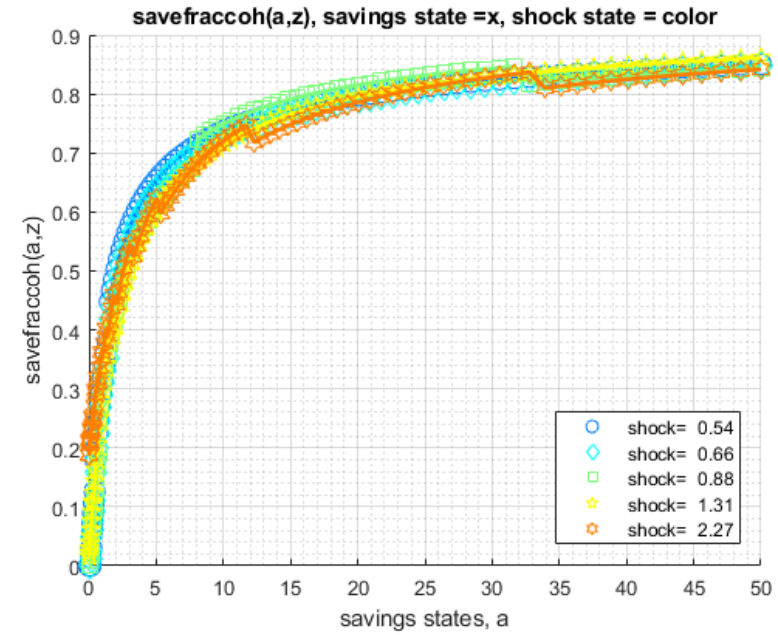
```
% Lower Risk Aversion
mp_params('fl_crra') = 0.5;
ff_vfi_az_loop(mp_params, mp_support);
```

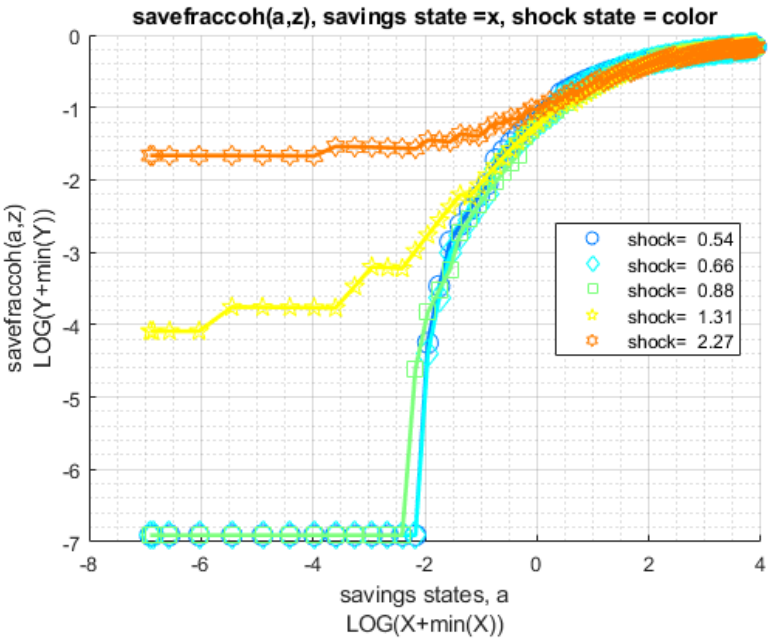
```
Elapsed time is 0.581794 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	
savefraccoh	1	1	2	500	100	5	268.82	0.53764	0.29852	0.55524	0	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5
r1	0	0	0	0.015741	0.18847
r2	0	0	0	0.01574	0.18847
r3	0	0	0	0.015737	0.18844
r4	0	0	0	0.015728	0.18838
r5	0	0	0	0.022367	0.18825
r96	0.84455	0.84169	0.83664	0.85445	0.83255
r97	0.84611	0.84333	0.83842	0.85626	0.83496
r98	0.84763	0.84493	0.84016	0.85803	0.83729
r99	0.84911	0.84648	0.84185	0.85974	0.83956
r100	0.85055	0.848	0.84349	0.86141	0.84176





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

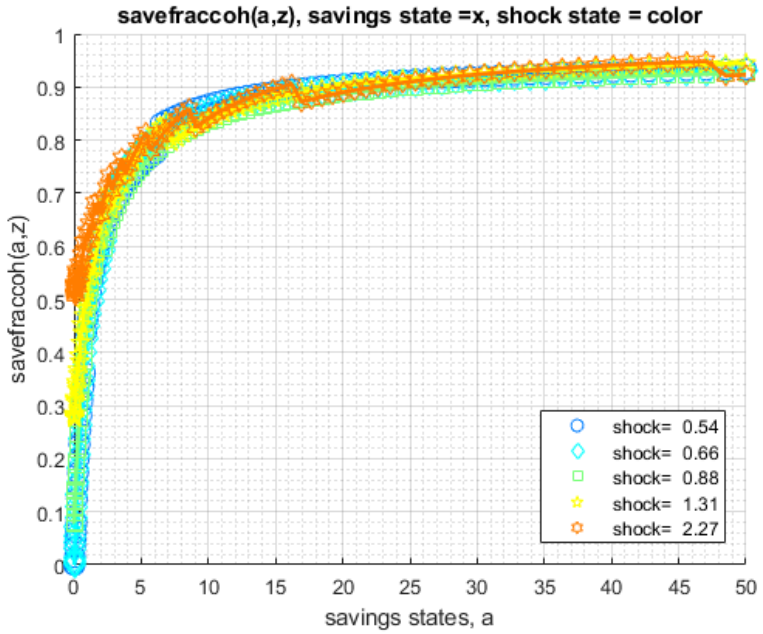
```
% Higher Risk Aversion
mp_params('fl_crra') = 5;
ff_vfi_az_loop(mp_params, mp_support);
```

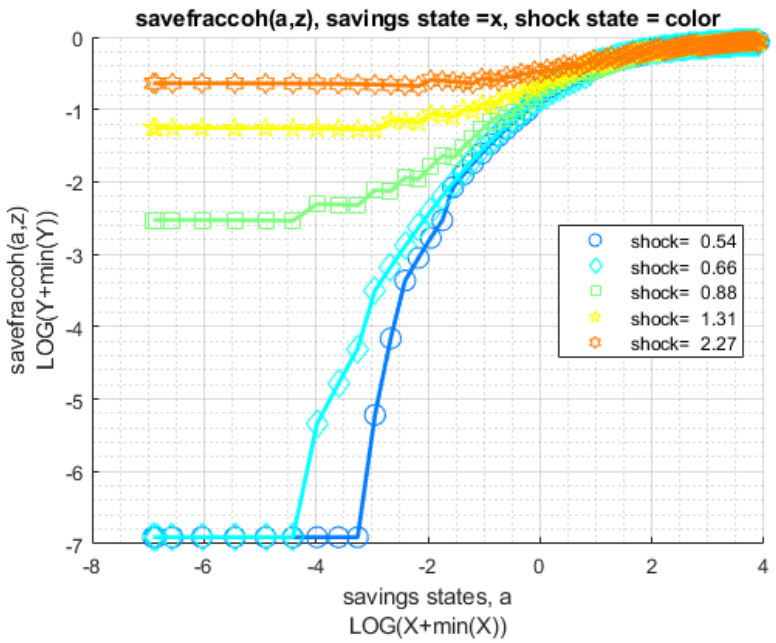
```
Elapsed time is 0.937495 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	
savefraccoh	1	1	2	500	100	5	335.64	0.67129	0.28688	0.42735	0	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0	0	0.078907	0.28472	0.52731
r2	0	0	0.078904	0.28471	0.5273
r3	0	0	0.078878	0.28465	0.52723
r4	0	0	0.078808	0.28448	0.52705
r5	0	0	0.078672	0.28415	0.52669
r96	0.93086	0.92771	0.92215	0.94079	0.94593
r97	0.93161	0.92855	0.92315	0.94183	0.94739
r98	0.93233	0.92936	0.92411	0.94283	0.9488
r99	0.93303	0.93015	0.92505	0.94379	0.92164
r100	0.93371	0.93091	0.92595	0.94471	0.92317





Test FF_VFI_AZ_LOOP 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') = 100;
mp_params('it_z_n') = 5;
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_loop(mp_params, mp_support);
```

```
Elapsed time is 0.957457 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       500     100      5      294.1    0.5882    0.32083    0.54544      0     0.9

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
              c1         c2         c3          c4         c5
            _______    _______    _______    ________    _______

    r1            0          0          0    0.034556    0.11424
    r2            0          0          0    0.034555    0.11424
    r3            0          0          0    0.034546    0.11422
    r4            0          0          0    0.034523    0.11416
    r5            0          0          0    0.034478    0.11404
    r96     0.89673    0.89421    0.91986     0.91499    0.90808
    r97     0.89789    0.89545    0.92093      0.9162    0.90948
    r98     0.89903    0.89665    0.92196     0.91737    0.91084
    r99     0.90013    0.89782    0.92295      0.9185    0.91215
    r100    0.90119    0.89896    0.92392     0.91959    0.91342
```

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_loop(mp_params, mp_support);
```

```
Elapsed time is 0.923630 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	500	100	5	350.37	0.70073	0.26741	0.38162	0	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5
r1	0	0	0.030722	0.36969	0.77072
r2	0	0	0.03072	0.36967	0.77071
r3	0	0	0.0307	0.36958	0.77068
r4	0	0	0.030646	0.36933	0.7706
r5	0	0	0.030543	0.36885	0.77044
r96	0.90975	0.90819	0.9038	0.91513	0.88687
r97	0.91053	0.90902	0.90476	0.91633	0.89076
r98	0.91129	0.90982	0.90569	0.9175	0.86794
r99	0.91204	0.91061	0.9066	0.91862	0.84583
r100	0.91276	0.91138	0.90748	0.91971	0.82439