FF_VFI_AZ_VEC Dynamic Programming Asset Problem with Shocks Vectorized

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 dynamica 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 vectorized code, its speed is much faster than the looped code.

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 0.380397 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       2100     300      7      50584    24.088    13.973    0.58008      0      50

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

    r1            0          0          0          0    0.16722     0.6689    2.0067
    r2            0          0          0    0.16722    0.33445    0.83612    2.1739
    r3      0.16722    0.16722    0.16722    0.16722    0.50167     1.0033    2.3411
    r4      0.33445    0.33445    0.33445    0.33445     0.6689     1.1706    2.5084
    r5      0.33445    0.33445    0.50167    0.50167    0.83612     1.3378    2.5084
    r296     46.823      46.99      46.99      47.157     47.492     48.161    49.498
    r297      46.99     47.157     47.157     47.324     47.659     48.328    49.666
    r298     47.157     47.324     47.324     47.492     47.826     48.495    49.833
    r299     47.324     47.492     47.492     47.659     47.993     48.662        50
    r300     47.492     47.659     47.659     47.826     48.161     48.829        50
```

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;
```

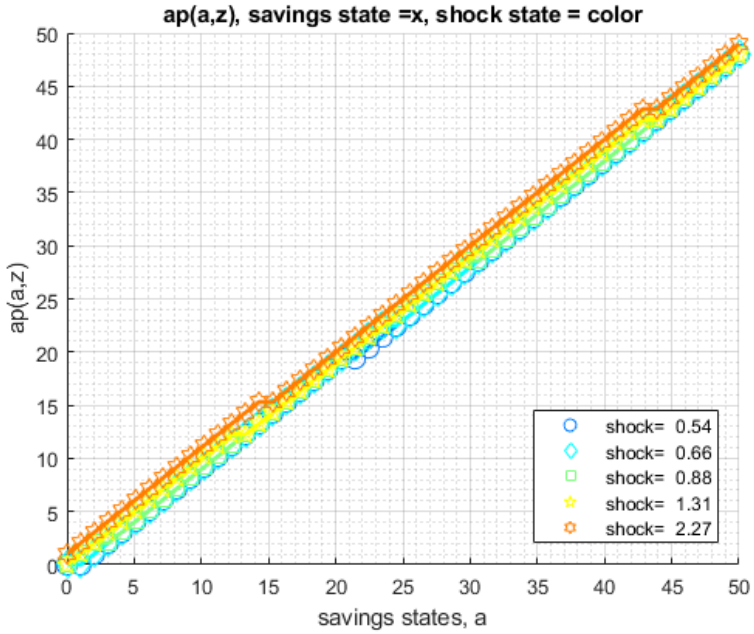
Run the function and show policy function for savings choice. For ls_ffcmd, ls_ffsna, ls_ffgrh, can include these: 'v', 'ap', 'c', 'y', 'coh', 'savefraccoh'. These are value, aprime savings choice, consumption, income, cash on hand, and savings fraction as cash-on-hand.

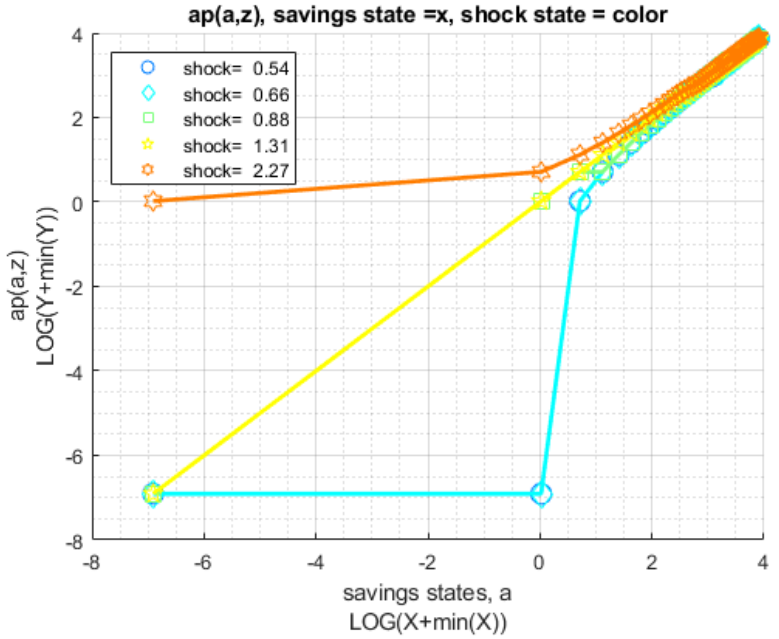
```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
% ls_ffcmd: summary print which outcomes
mp_support('ls_ffcmd') = {};
% ls_ffsna: detail print which outcomes
mp_support('ls_ffsna') = {'ap'};
% ls_ffgrh: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

```
Elapsed time is 0.016286 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_vec(mp_params, mp_support);
```

```
Elapsed time is 0.120582 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 xxxxxxxxxxxxxxxxxx
```

	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;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.745276 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      3291.4    0.48762    0.27804    0.57021      0

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

    r1            0          0          0          0          0          0    0.01987    0.12517    0.29012
    r2            0          0          0          0          0          0    0.01987    0.12517    0.29012
    r3            0          0          0          0          0          0    0.01987    0.12517    0.29012
    r4            0          0          0          0          0          0    0.01987    0.12517    0.29012
    r5            0          0          0          0          0          0    0.01987    0.12517    0.29012
    r746    0.80538    0.80084    0.79932     0.7971    0.79372    0.79177    0.78608    0.77969    0.77353
    r747    0.80218    0.80112     0.7996    0.79739    0.79402    0.79208    0.78643    0.78008      0.774
    r748    0.80245    0.80139    0.79988    0.79767    0.79432     0.7924    0.78677    0.78046    0.77447
    r749    0.80272    0.80167    0.80016    0.79796    0.79462    0.79271    0.78711    0.78085    0.77493
    r750    0.80299    0.80194    0.80044    0.79825    0.79492    0.79303    0.78745    0.78124     0.7754
```

```
% 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 2.419509 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:

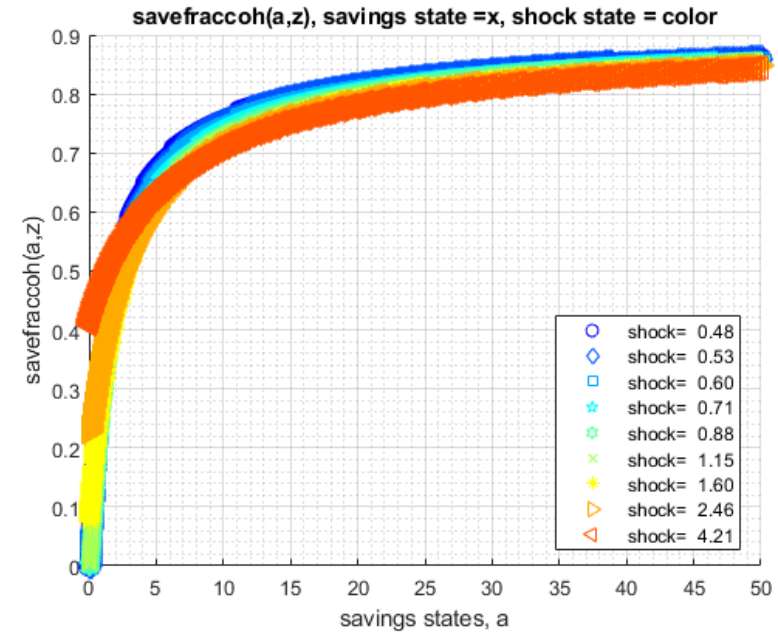
```
% Lower Risk Aversion
mp_params('fl_crra') = 0.5;
ff_vfi_az_vec(mp_params, mp_support);
```

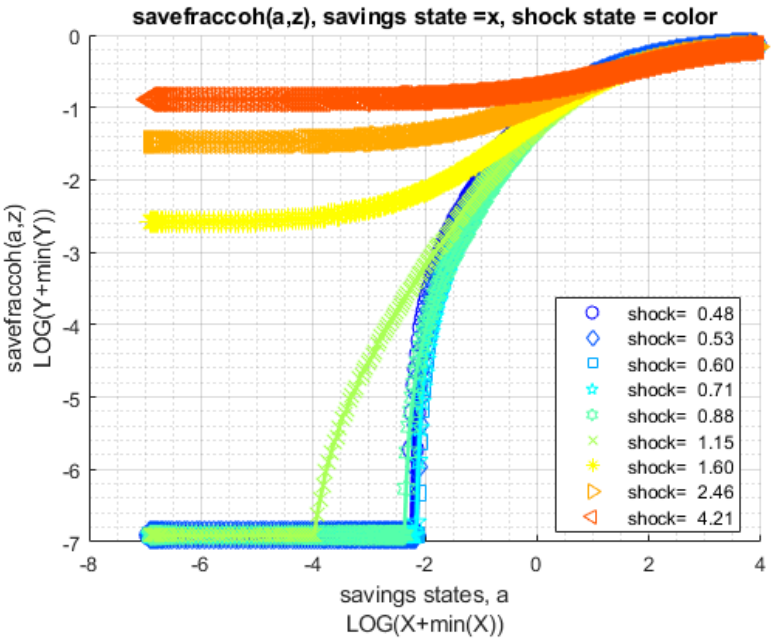
```
Elapsed time is 1.950318 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	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	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.

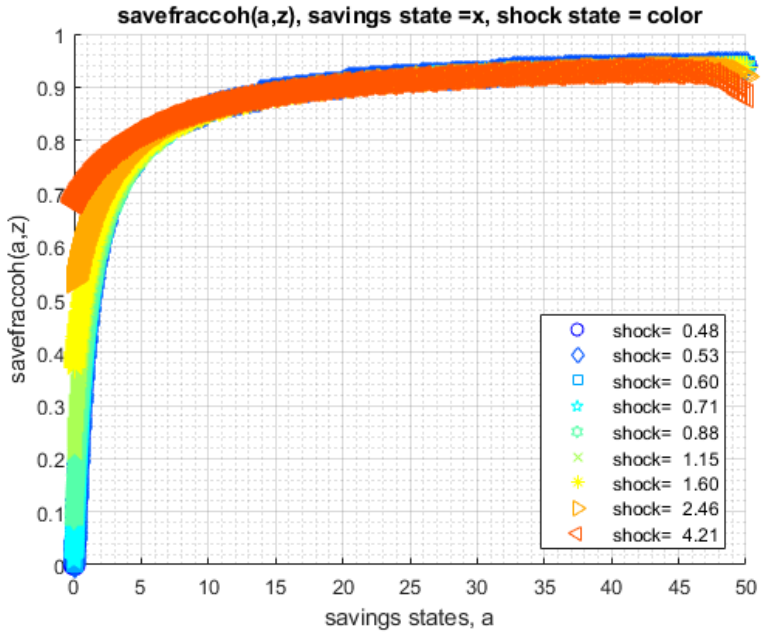
```
% Higher Risk Aversion
mp_params('fl_crra') = 5;
ff_vfi_az_vec(mp_params, mp_support);
```

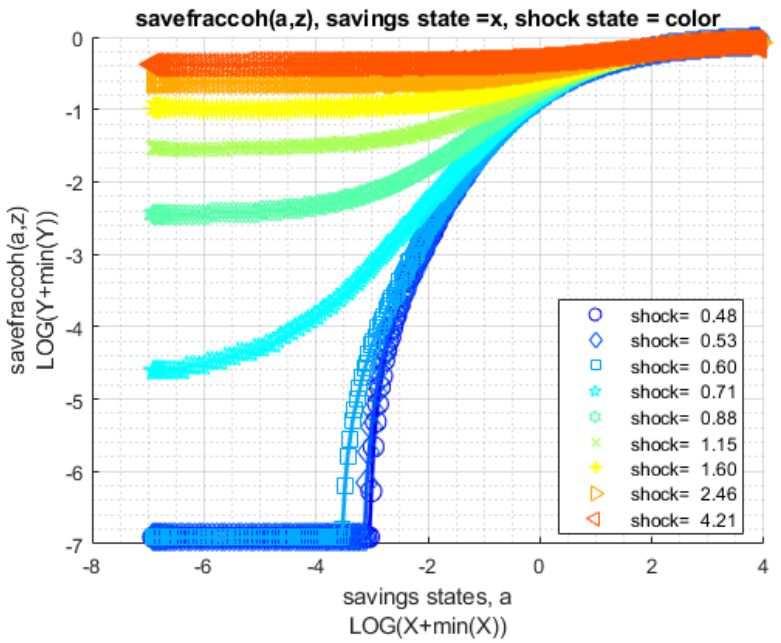
```
Elapsed time is 2.480325 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	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') = 0.05;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 1.980327 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	3935.8	0.58309	0.32813	0.56274	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0	0.0035419	0.022183	0.050593
r2	0	0	0	0	0	0	0.0035419	0.022183	0.050593
r3	0	0	0	0	0	0	0.0035419	0.022183	0.050593
r4	0	0	0	0	0	0	0.0035419	0.022182	0.050593
r5	0	0	0	0	0	0	0.0035419	0.022182	0.050593
r746	0.91062	0.90972	0.91245	0.91134	0.91009	0.91241	0.91083	0.90905	0.91074
r747	0.91075	0.90986	0.91259	0.91148	0.91024	0.91256	0.91099	0.90921	0.9109
r748	0.91088	0.91	0.91272	0.91162	0.91038	0.9127	0.91114	0.90937	0.91106
r749	0.91102	0.91013	0.91286	0.91176	0.91053	0.91285	0.91129	0.90952	0.91122
r750	0.91115	0.91027	0.91299	0.9119	0.91067	0.90929	0.91144	0.90968	0.91138

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') = 0.25;
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 1.941686 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	4429.3	0.65619	0.28387	0.43261	0	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0.011319	0.12886	0.32464	0.53487	0.72181
r2	0	0	0	0	0.011319	0.12886	0.32464	0.53487	0.72181
r3	0	0	0	0	0.011319	0.12886	0.32464	0.53487	0.72181
r4	0	0	0	0	0.011319	0.12886	0.32464	0.53487	0.72181
r5	0	0	0	0	0.011319	0.12886	0.32464	0.53487	0.72181
r746	0.91612	0.91885	0.9173	0.91484	0.91448	0.91454	0.91098	0.90731	0.83143
r747	0.91622	0.91896	0.91741	0.91496	0.9146	0.91469	0.91117	0.90394	0.82863
r748	0.91633	0.91906	0.91751	0.91507	0.91473	0.91483	0.91136	0.90422	0.82584
r749	0.91643	0.91916	0.91762	0.91519	0.91486	0.91498	0.91154	0.90449	0.82306
r750	0.91653	0.91926	0.91773	0.91531	0.91498	0.91512	0.91173	0.90115	0.82029