FF_VFI_AZ_BISEC_VEC Savings Vectorized Exact (FOC) Examples

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

This is the example vignette for function: **ff_vfi_az_bisec_vec** 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 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.

The code uses **continuous choices**, solved with bi(multi)section. The state-space is on a grid, but choice grids are in terms of percentage of resources available, which is individual specific, to save and solved exactly up to ((1/(2)^16)*100=0.001525878) percentage of cash on hand. The **ff_vfi_az_vec** from the **MEconTools Package** solves the same problem using vectorized common grid code where the choice set and state space share the same grid. The common grid function is faster, but less precise for the same number of asset grid points.

Links to Other Code:

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

- Common Choice and States Grid ***Loop***: **ff_vfi_az_loop**
- Common Choice and States Grid ***Vectorized***: **ff_vfi_az_vec**
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand, rely on FOC, ***Loop***: **ff_vfi_az_bisec_loop**
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand, rely on FOC ***Vectorized***: **ff_vfi_az_bisec_vec**
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand, VALUE comparison, ***Loop***: **ff_vfi_az_mzoom_loop**
- States Grid + Continuous Exact Savings as Share of Cash-on-Hand, VALUE comparison, ***Vectorized***: **ff_vfi_az_mzoom_vec**

Test FF_VFI_AZ_BISEC_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;
% call function
ff_vfi_az_bisec_vec(mp_params);
```

```
Elapsed time is 1.762201 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	9863.4	14.091	14.388	1.0211	0	50.117

```
xxx TABLE:ap xxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7
r1	0	0	0	0.053491	0.25574	0.60604	1.1157
r2	0	0	0	0.053998	0.25571	0.6066	1.1163
r3	0	0	0	0.056449	0.25576	0.60907	1.1187
r4	0	0	0	0.061799	0.26016	0.6109	1.1239
r5	0	0	0	0.066463	0.26897	0.61141	1.1327
r96	43.388	43.52	43.701	43.925	44.222	44.68	45.228
r97	44.566	44.695	44.878	45.103	45.398	45.856	46.403
r98	45.761	45.892	46.072	46.298	46.592	47.05	47.597
r99	46.973	47.107	47.286	47.514	47.806	48.263	48.815
r100	48.206	48.338	48.519	48.746	49.037	49.497	50.117

Test FF_VFI_AZ_BISEC_VEC Speed Tests

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

```
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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 0.792541 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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 43.095190 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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 80.139775 seconds.
```

Test FF_VFI_AZ_BISEC_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') = true;
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
mp_support('ls_ffcmd') = {};
ff_vfi_az_vec(mp_params, mp_support);
```

```
Elapsed time is 0.029901 seconds.
```

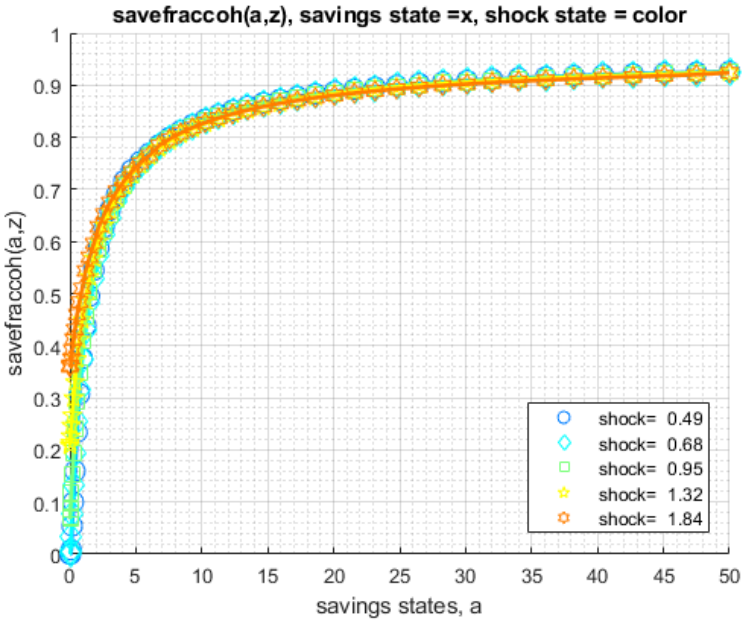
Run the function and show policy function for savings choice. For ls_ffcmd, ls_ffsna, ls_ffgrh, can include these: 'v', 'ap', 'c', 'y', 'coh', 'savefraccoh'. These are value, aprime savings choice, consumption, income, cash on hand, and savings fraction as cash-on-hand.

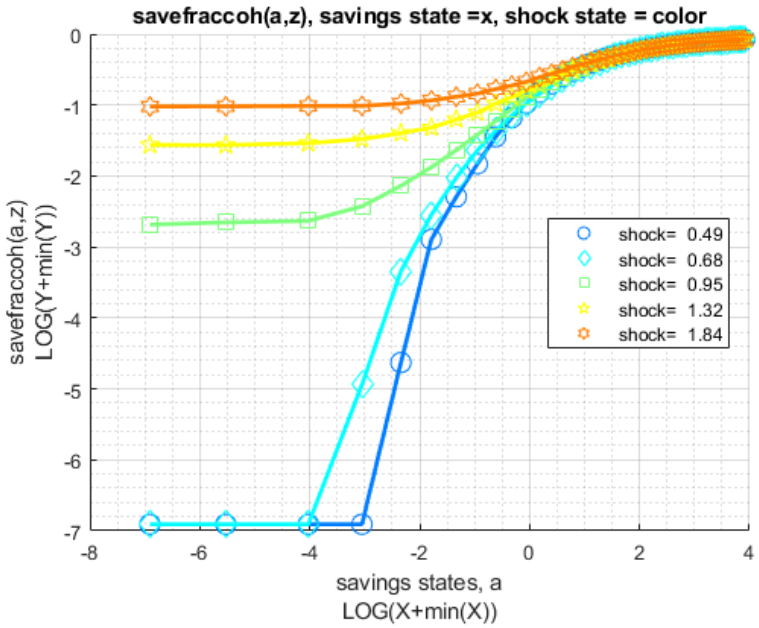
```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
% ls_ffcmd: summary print which outcomes
mp_support('ls_ffcmd') = {};
% ls_ffsna: detail print which outcomes
mp_support('ls_ffsna') = {'savefraccoh'};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

```
Elapsed time is 0.494900 seconds.
xxx  ff_vfi_az_vec, outcome=savefraccoh  xxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	a	mean_z_0_4858	mean_z_0_67798	mean_z_0_9462	mean_z_1_3205	mean_z_1_8429
1	0	0	0	0.067239	0.20859	0.35953
2	0.002975	0	0	0.069375	0.20829	0.36032
3	0.016829	0	0	0.070901	0.2139	0.36215
4	0.046375	0	0.0061439	0.087319	0.2266	0.36264
5	0.095198	0.0087684	0.034403	0.1168	0.2468	0.37473
6	0.1663	0.054361	0.077248	0.1522	0.26639	0.39151
7	0.26234	0.099892	0.13132	0.19388	0.29929	0.41281
8	0.38568	0.15958	0.19309	0.24112	0.33017	0.43088
9	0.53852	0.23417	0.25553	0.29215	0.37436	0.45969
10	0.72291	0.3071	0.31656	0.34812	0.41153	0.48386
11	0.94076	0.37595	0.37503	0.40842	0.44925	0.50992
12	1.1939	0.43881	0.42941	0.45755	0.48697	0.54367
13	1.484	0.49509	0.48129	0.50381	0.53262	0.56979
14	1.8128	0.54489	0.53018	0.54642	0.56778	0.59634
15	2.1817	0.58871	0.57382	0.58548	0.60055	0.6282
16	2.5924	0.62716	0.61258	0.62076	0.63101	0.65249
17	3.0463	0.66079	0.64682	0.65243	0.65884	0.6752
18	3.5449	0.69027	0.67709	0.68069	0.68423	0.69638
19	4.0894	0.71621	0.70376	0.70596	0.70724	0.71591
20	4.6813	0.73703	0.72732	0.72848	0.72799	0.73385
21	5.3218	0.75326	0.74813	0.7485	0.74673	0.75021
22	6.0121	0.76913	0.76657	0.76632	0.76364	0.76535
23	6.7536	0.78536	0.78286	0.78231	0.77889	0.7842
24	7.5474	0.79983	0.79745	0.79653	0.79269	0.79678
25	8.3948	0.81271	0.81039	0.80929	0.80514	0.80831

26	9.2967	0.82418	0.82198	0.82076	0.81637	0.81875
27	10.254	0.8345	0.83242	0.83114	0.82656	0.82833
28	11.269	0.84377	0.84176	0.84042	0.83584	0.83706
29	12.342	0.85214	0.85024	0.84884	0.8442	0.84499
30	13.473	0.85964	0.85781	0.85647	0.85183	0.85232
31	14.665	0.86648	0.86471	0.86337	0.85879	0.85897
32	15.918	0.87264	0.87099	0.86965	0.86507	0.86507
33	17.233	0.87826	0.87667	0.87533	0.87161	0.87063
34	18.611	0.88338	0.88186	0.88052	0.87771	0.87582
35	20.053	0.88802	0.88656	0.88528	0.88326	0.88052
36	21.56	0.8923	0.89089	0.88967	0.88833	0.88485
37	23.133	0.89614	0.89486	0.89364	0.8926	0.88888
38	24.773	0.89974	0.89852	0.8973	0.89626	0.8926
39	26.481	0.90304	0.90182	0.90072	0.89968	0.89608
40	28.258	0.90603	0.90493	0.90383	0.90279	0.89925
41	30.104	0.90884	0.90774	0.9067	0.90572	0.90218
42	32.021	0.9114	0.91036	0.90932	0.90841	0.90493
43	34.01	0.91378	0.9128	0.91183	0.91091	0.90749
44	36.07	0.91598	0.91506	0.91408	0.91317	0.90987
45	38.204	0.91805	0.91714	0.91622	0.91537	0.91207
46	40.412	0.91994	0.91909	0.91817	0.91732	0.91415
47	42.695	0.92171	0.92086	0.92001	0.91921	0.9161
48	45.053	0.92336	0.92257	0.92171	0.92092	0.91799
49	47.488	0.92489	0.92409	0.92336	0.92257	0.92025
50	50	0.92629	0.92562	0.92489	0.92428	0.92403





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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 1.164186 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      12926     14.362     14.544      1.0127      0
    savefraccoh   2     2      2       900     100      9      621.24    0.69027    0.26896     0.38965     0

xxx TABLE:ap xxxxxxxxxxxxxxxxxx
              c1         c2         c3          c4          c5          c6          c7          c8          c9
          _______    _______    _______    _________    ________    ________    ________    _____    _____

    r1          0          0          0            0    0.087442     0.27778     0.58243    1.0038    1.5724
    r2          0          0          0            0    0.087962     0.27828     0.58297    1.0044    1.5731
    r3          0          0          0            0    0.090477     0.28074     0.58547    1.0069    1.5755
    r4          0          0          0   0.00055771     0.09279     0.28605      0.5907    1.0122    1.5808
    r5          0          0          0    0.0059496     0.09602     0.29477     0.59952    1.0209    1.5895
    r96    43.845     43.923     44.022       44.198      44.428      44.722      45.103    45.546    46.186
    r97    45.031     45.101     45.208       45.384      45.613       45.91      46.293    46.735    47.382
    r98    46.237     46.297     46.411        46.59      46.818      47.115      47.501    47.948    48.605
    r99     47.46     47.512     47.635       47.812      48.041       48.34      48.726    49.191    49.869
    r100   48.703     48.746     48.878       49.055      49.283      49.586      49.978    50.495    51.171

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

r1	0	0	0	0	0.066018	0.16569	0.27445	0.37369	0.46243
r2	0	0	0	0	0.066384	0.16593	0.27463	0.37381	0.46256
r3	0	0	0	0	0.068154	0.16715	0.27549	0.37442	0.46292
r4	0	0	0	0.00052879	0.069619	0.16978	0.27726	0.37564	0.46378
r5	0	0	0	0.0055946	0.071572	0.17405	0.28025	0.37766	0.46512
r96	0.92458	0.92354	0.92226	0.92171	0.92116	0.92055	0.91994	0.91842	0.91811
r97	0.92531	0.92416	0.92306	0.92251	0.92196	0.92141	0.92086	0.91933	0.91915
r98	0.92605	0.9247	0.92379	0.9233	0.92275	0.9222	0.92171	0.92031	0.92031
r99	0.92672	0.92525	0.92452	0.92403	0.92348	0.923	0.92251	0.92147	0.92184
r100	0.92739	0.9258	0.92525	0.92477	0.92422	0.92379	0.92342	0.92336	0.92367

Test FF_VFI_AZ_BISEC_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') = 100;
mp_params('it_z_n') = 7;
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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 0.271658 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	700	100	7	357.85	0.51122	0.27528	0.53848	0	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7
r1	0	0	0	0	0	0.00022362	0.041544
r2	0	0	0	0	0	0.00022362	0.041544
r3	0	0	0	0	0	0.0011391	0.041544
r4	0	0	0	0	0	0.0016884	0.041483
r5	0	0	0	0	0	0.0034584	0.04136
r96	0.79586	0.79275	0.78945	0.78591	0.78225	0.77853	0.77059
r97	0.79684	0.79379	0.79055	0.78713	0.78359	0.77993	0.77212
r98	0.79782	0.79482	0.79171	0.78835	0.78488	0.78127	0.77365
r99	0.79873	0.79586	0.79275	0.78951	0.7861	0.78262	0.77548
r100	0.79965	0.79684	0.79385	0.79061	0.78732	0.7839	0.7781

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

```
Elapsed time is 0.971218 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       700      100      7      481.37    0.68768    0.27118    0.39435      0
```

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

    r1            0          0          0    0.065774    0.18076    0.30655    0.41654
    r2            0          0          0    0.066201    0.18101    0.30674     0.4166
    r3            0          0          0     0.06791    0.18223    0.30747    0.41709
    r4            0          0          0    0.069619    0.18467    0.30759    0.41812
    r5            0          0          0    0.071694    0.18876    0.30838    0.41983
    r96     0.92428    0.92245    0.92178     0.92116    0.92049    0.91872    0.91824
    r97     0.92501    0.92324    0.92257     0.92196    0.92129    0.91958    0.91921
    r98     0.92574    0.92397    0.92336     0.92275    0.92208    0.92049    0.92025
    r99     0.92647     0.9247    0.92409     0.92348    0.92287    0.92147    0.92159
    r100    0.92702    0.92544    0.92483     0.92422    0.92373    0.92336    0.92348
```

Test FF_VFI_AZ_BISEC_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') = 100;
mp_params('it_z_n') = 7;
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:

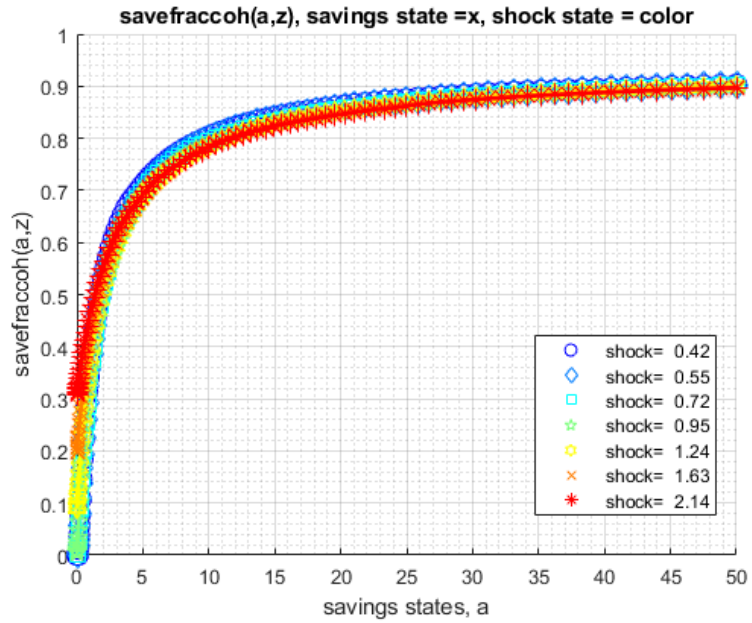
```
% Lower Risk Aversion
mp_params('fl_crra') = 0.5;
ff_vfi_az_bisec_vec(mp_params, mp_support);
```

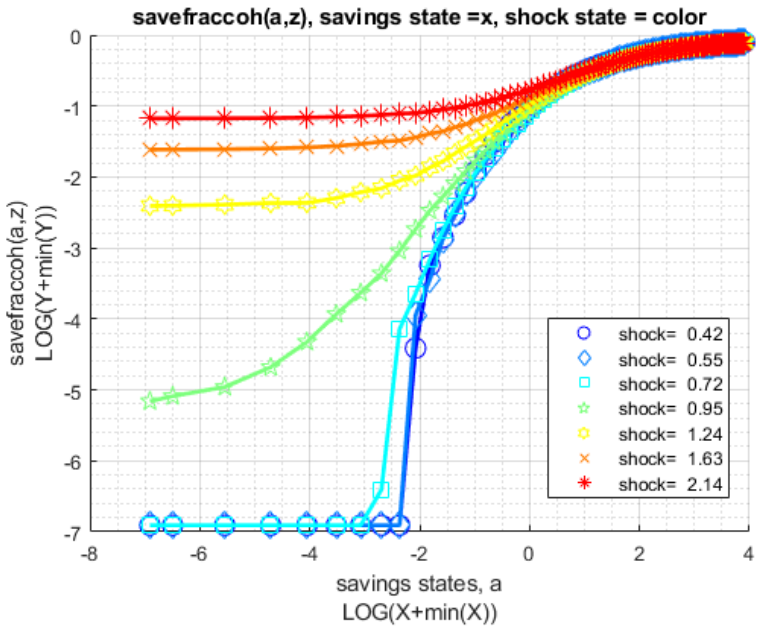
```
Elapsed time is 0.873752 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN     sum       mean       std       coefvari    min
```

	_	___	____	_____	____	____	______	______	_______	________	___	_
savefraccoh	1	1	2	700	100	7	452.13	0.6459	0.28031	0.43398	0	0.

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7
r1	0	0	0	0.0047401	0.089089	0.19822	0.30783
r2	0	0	0	0.0051674	0.089394	0.1984	0.30796
r3	0	0	0	0.0060218	0.090676	0.19926	0.30851
r4	0	0	0	0.0082801	0.092812	0.20115	0.30973
r5	0	0	0	0.012247	0.092995	0.2042	0.31174
r96	0.90047	0.89925	0.89828	0.8973	0.89632	0.89376	0.89297
r97	0.90127	0.90017	0.89919	0.89828	0.8973	0.8948	0.89394
r98	0.90206	0.90102	0.90011	0.89919	0.89828	0.89577	0.89498
r99	0.90279	0.90188	0.90102	0.90011	0.89919	0.89681	0.8959
r100	0.90359	0.90273	0.90188	0.90096	0.90011	0.89803	0.89687





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

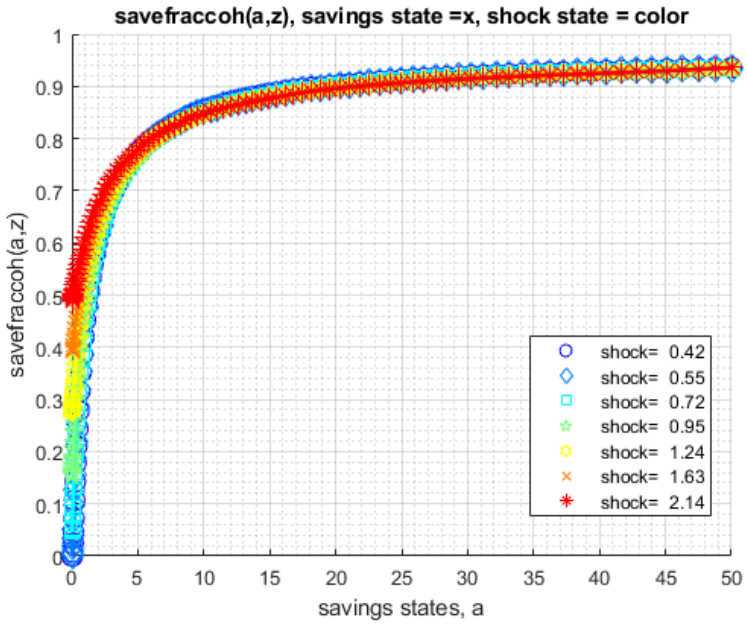
```
% Higher Risk Aversion
mp_params('fl_crra') = 5;
ff_vfi_az_bisec_vec(mp_params, mp_support);
```

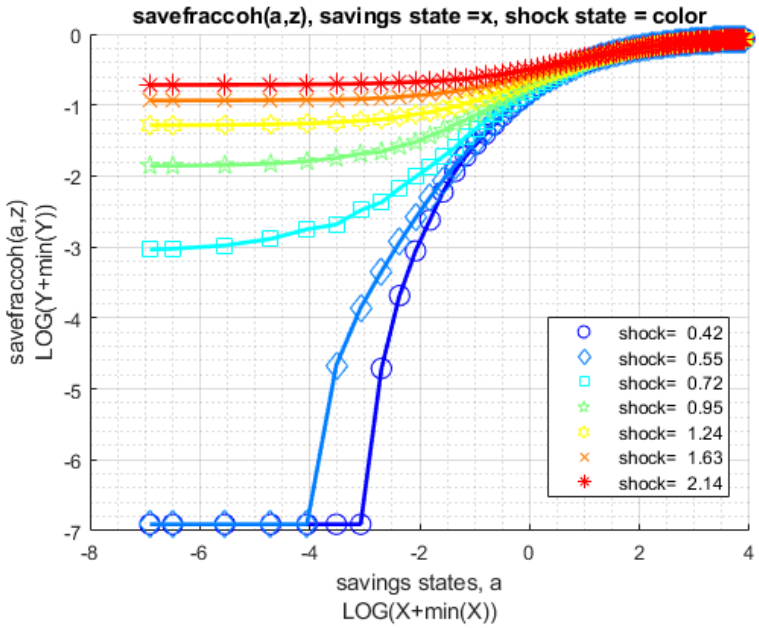
```
Elapsed time is 0.970314 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min
savefraccoh	1	1	2	700	100	7	502.71	0.71816	0.25437	0.3542	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7
r1	0	0	0.047037	0.15537	0.27573	0.3909	0.48782
r2	0	0	0.047525	0.15531	0.27591	0.39102	0.48795
r3	0	0	0.049844	0.1569	0.27695	0.3917	0.48837
r4	0	0	0.054788	0.16025	0.27915	0.3931	0.48929
r5	0	0	0.062905	0.16569	0.28275	0.39542	0.49075
r96	0.93307	0.93258	0.93203	0.93154	0.9302	0.92995	0.92971
r97	0.93374	0.93325	0.93276	0.93227	0.93111	0.93105	0.93117
r98	0.93441	0.93398	0.93349	0.93307	0.93209	0.93227	0.9327
r99	0.93508	0.93465	0.93423	0.93392	0.93331	0.93368	0.93435
r100	0.93575	0.93539	0.93508	0.9349	0.93496	0.93526	0.93587





Test FF_VFI_AZ_BISEC_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') = 150;
mp_params('it_z_n') = 15;
mp_params('fl_a_max') = 50;
mp_params('st_grid_type') = 'grid_powerspace';
% graph color spectrum
mp_params('cl_colors') = 'copper';
```

Lower standard deviation of shock:

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

```
Elapsed time is 2.595920 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	2250	150	15	1507.5	0.67001	0.28668	0.42788	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c11	c12	c13	c14	c15
r1	0	0	0	0	0	0.13847	0.18485	0.23026	0.27378	0.317
r2	0	0	0	0	0	0.13853	0.18491	0.23032	0.27384	0.317
r3	0	0	0	0	0	0.13895	0.18528	0.23063	0.27408	0.31
r4	0	0	0	0	0	0.13987	0.18607	0.2313	0.27469	0.318
r5	0	0	0	0	0	0.14011	0.18735	0.2324	0.27567	0.318
r146	0.92373	0.92354	0.9233	0.92312	0.92287	0.92086	0.92068	0.92049	0.91952	0.919
r147	0.92422	0.92403	0.92385	0.92361	0.92342	0.92141	0.92123	0.92098	0.92007	0.919
r148	0.9247	0.92452	0.92434	0.92409	0.92391	0.9219	0.92171	0.92153	0.92062	0.920
r149	0.92519	0.92501	0.92483	0.92458	0.9244	0.92245	0.92226	0.92208	0.92116	0.92
r150	0.92568	0.9255	0.92531	0.92507	0.92489	0.92293	0.92275	0.92257	0.92245	0.922

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_bisec_vec(mp_params, mp_support);
```

```
Elapsed time is 2.805227 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	2250	150	15	1685.6	0.74914	0.22909	0.3058	0	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c11	c12	c13	c14	c15
r1	0	0	0	0	0	0.5264	0.61264	0.68271	0.73922	0.784
r2	0	0	0	0	0	0.52646	0.61264	0.68271	0.73922	0.784
r3	0	0	0	0	0	0.52658	0.6127	0.68271	0.73922	0.784
r4	0	0	0	0	0	0.52682	0.61288	0.68283	0.73928	0.784
r5	0	0	0	0	0	0.52731	0.61313	0.68295	0.73934	0.784
r146	0.92983	0.92928	0.92873	0.92806	0.92739	0.92269	0.92354	0.9258	0.92904	0.933
r147	0.9302	0.92971	0.9291	0.92849	0.92788	0.92361	0.92477	0.9269	0.93001	0.934
r148	0.93056	0.93008	0.92953	0.92892	0.92831	0.92458	0.92593	0.928	0.93105	0.935
r149	0.93093	0.93044	0.92995	0.92934	0.92873	0.9258	0.92702	0.9291	0.93203	0.9
r150	0.9313	0.93087	0.93032	0.92977	0.92916	0.92696	0.92818	0.93014	0.93294	0.936