FF_VFI_AZ_BISEC_VEC Dynamic Savings Problem Vectorized Continuous Exact

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 a vectorized code, it is much faster for larger state-space problems then looped code.

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.

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_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 0.341348 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      15835    22.621    13.367    0.59091      0     47.273

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

    r1          0          0          0          0          0    0.38021    1.4609
    r2    0.19477    0.18872    0.19731    0.24709    0.41492    0.79311    1.8893
    r3    0.54595    0.54109    0.55664    0.62239    0.81173     1.2132    2.3195
    r4     1.0101     1.0101     1.0101     1.0189     1.2217     1.6363    2.7464
    r5     1.4388     1.4362      1.459     1.5151     1.6354     2.0602    3.1804
    r96    43.225     43.246       43.3     43.422     43.632     44.155    45.413
    r97     43.69      43.71     43.765     43.887     44.096     44.618    45.879
    r98    44.154     44.174     44.228     44.352     44.559     45.083    46.344
    r99    44.618     44.638     44.693     44.815     45.024     45.548    46.809
    r100    45.08     45.101     45.156      45.28     45.487     46.012    47.273
```

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 0.188450 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 12.017243 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 22.719622 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') = false;
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
```

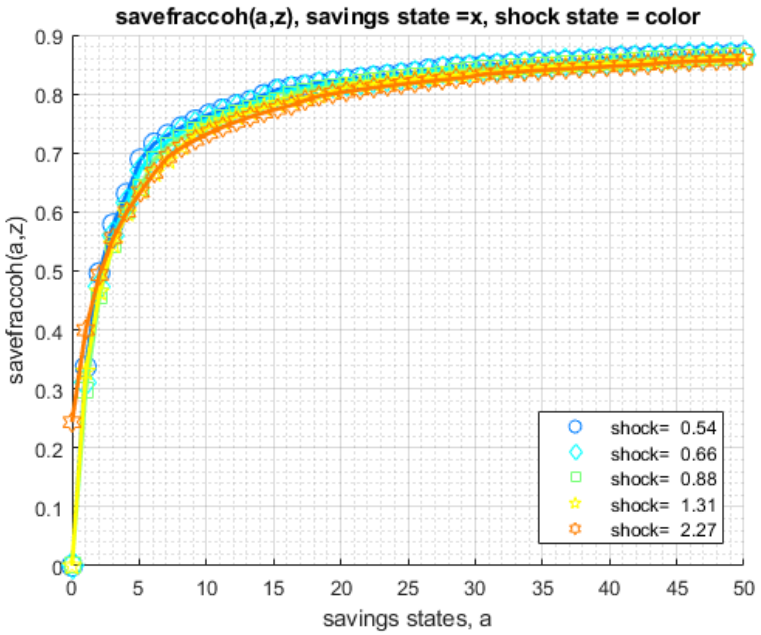
Run the function and show policy function for savings choice. For ls_ffcmd, ls_ffsna, ls_ffgrh, can include these: 'v', 'ap', 'c', 'y', 'coh', 'savefraccoh'. These are value, aprime savings choice, consumption, income, cash on hand, and savings fraction as cash-on-hand.

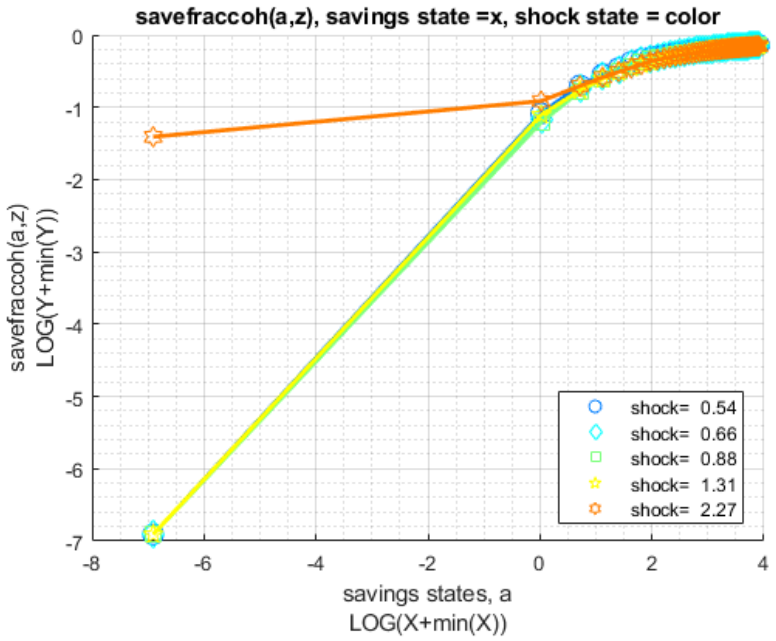
```
mp_support = containers.Map('KeyType','char', 'ValueType','any');
mp_support('bl_print_params') = false;
mp_support('bl_print_iterinfo') = false;
% ls_ffcmd: summary print which outcomes
mp_support('ls_ffcmd') = {};
% ls_ffsna: detail print which outcomes
mp_support('ls_ffsna') = {'savefraccoh'};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

```
Elapsed time is 0.160923 seconds.
xxx  ff_vfi_az_vec, outcome=savefraccoh  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

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	0.24358
2	1.0204	0.33724	0.31063	0.29793	0.32952	0.3998
3	2.0408	0.49626	0.47337	0.4572	0.46446	0.49318
4	3.0612	0.57912	0.56065	0.5457	0.54484	0.55559
5	4.0816	0.63096	0.61577	0.60252	0.59846	0.60036
6	5.102	0.68907	0.67137	0.64222	0.63694	0.63377
7	6.1224	0.71595	0.7043	0.6891	0.66563	0.66642
8	7.1429	0.73066	0.72084	0.71144	0.68766	0.69337
9	8.1633	0.74391	0.73503	0.72618	0.71119	0.70921
10	9.1837	0.75538	0.74739	0.73918	0.73335	0.72236
11	10.204	0.7653	0.75798	0.75032	0.74412	0.73381
12	11.224	0.77394	0.76719	0.75999	0.75367	0.74403
13	12.245	0.78147	0.77525	0.76847	0.76231	0.75306
14	13.265	0.78816	0.78233	0.77598	0.77006	0.76115
15	14.286	0.79841	0.79035	0.78266	0.77699	0.76838
16	15.306	0.80723	0.80201	0.7888	0.78321	0.77488
17	16.327	0.81135	0.8065	0.79972	0.78883	0.78077
18	17.347	0.81474	0.81031	0.80534	0.79386	0.78904
19	18.367	0.81815	0.81388	0.80918	0.79841	0.79634
20	19.388	0.82121	0.81715	0.8126	0.805	0.80027
21	20.408	0.82414	0.82026	0.81596	0.81172	0.80393
22	21.429	0.82685	0.82313	0.81898	0.81492	0.80732
23	22.449	0.82938	0.82584	0.82182	0.81776	0.81053
24	23.469	0.83177	0.82838	0.8245	0.8205	0.81352
25	24.49	0.83399	0.83073	0.827	0.8231	0.81635
26	25.51	0.83634	0.83296	0.82935	0.82554	0.81901
27	26.531	0.84156	0.83689	0.83155	0.82786	0.82151
28	27.551	0.84394	0.84098	0.8339	0.83003	0.82389
29	28.571	0.84553	0.84266	0.83875	0.8321	0.82612
30	29.592	0.84693	0.84425	0.84107	0.83405	0.82877
31	30.612	0.84821	0.84562	0.84266	0.83589	0.83326

32	31.633	0.84956	0.84699	0.84409	0.83787	0.83527
33	32.653	0.85084	0.84837	0.84547	0.84199	0.83686
34	33.673	0.852	0.84962	0.84684	0.84391	0.83842
35	34.694	0.85316	0.85081	0.84815	0.84528	0.83988
36	35.714	0.85429	0.852	0.84934	0.8465	0.84129
37	36.735	0.85532	0.85313	0.85053	0.84773	0.84266
38	37.755	0.85633	0.8542	0.85169	0.84895	0.84397
39	38.776	0.85795	0.85523	0.85279	0.85008	0.84522
40	39.796	0.86091	0.85767	0.85383	0.85114	0.84641
41	40.816	0.86176	0.85975	0.85499	0.85221	0.8476
42	41.837	0.86256	0.8606	0.85786	0.85325	0.8487
43	42.857	0.86332	0.86143	0.85917	0.85423	0.8498
44	43.878	0.86399	0.86216	0.85999	0.85517	0.85236
45	44.898	0.86463	0.86283	0.86079	0.85609	0.85401
46	45.918	0.86533	0.86356	0.86149	0.85831	0.85493
47	46.939	0.86601	0.86427	0.86219	0.85996	0.85578
48	47.959	0.86665	0.86494	0.86292	0.86073	0.85658
49	48.98	0.86723	0.86558	0.86362	0.86146	0.85737
50	50	0.86781	0.86619	0.86427	0.86216	0.85813





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 0.443544 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	20493	22.77	13.386	0.5879	0
savefraccoh	2	2	2	900	100	9	701.94	0.77994	0.13136	0.16842	0

xxx TABLE:ap xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0	0.20716	0.89208	2.4759
r2	0.19971	0.19144	0.18896	0.2007	0.24755	0.38215	0.61592	1.3126	2.9098
r3	0.55145	0.54262	0.54255	0.5618	0.62321	0.77699	1.0303	1.7326	3.3477
r4	1.0101	1.0101	1.0101	1.0101	1.0198	1.1844	1.5151	2.1613	3.7899
r5	1.4445	1.436	1.4393	1.4657	1.5152	1.5944	1.9615	2.5895	4.2337
r96	43.226	43.233	43.257	43.313	43.424	43.584	43.951	44.764	46.479
r97	43.69	43.697	43.722	43.776	43.888	44.048	44.444	45.227	46.97
r98	44.155	44.161	44.186	44.241	44.352	44.512	44.933	45.692	47.461
r99	44.619	44.626	44.65	44.707	44.817	44.976	45.398	46.156	47.927
r100	45.081	45.088	45.114	45.169	45.28	45.454	45.861	46.621	48.391

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9

r1	0	0	0	0	0	0	0.10085	0.28368	0.45924
r2	0.17696	0.16018	0.14648	0.1404	0.15072	0.19253	0.23949	0.35842	0.49245
r3	0.33498	0.31679	0.30013	0.28853	0.2885	0.31047	0.33348	0.41451	0.52092
r4	0.46678	0.45284	0.43437	0.40981	0.38082	0.39214	0.42003	0.46007	0.54576
r5	0.53868	0.52254	0.50624	0.49144	0.47417	0.45067	0.47554	0.49651	0.56737
r96	0.86817	0.86713	0.86597	0.86469	0.86323	0.86054	0.85786	0.85551	0.85172
r97	0.86845	0.86744	0.86631	0.865	0.86356	0.86091	0.8588	0.8559	0.85264
r98	0.86875	0.86774	0.86662	0.86533	0.8639	0.86128	0.85966	0.8563	0.85352
r99	0.86903	0.86805	0.86692	0.86567	0.86424	0.86161	0.86002	0.8567	0.85395
r100	0.86927	0.86829	0.8672	0.86594	0.86454	0.86222	0.86036	0.85709	0.85435

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

```
Elapsed time is 2.064615 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	2573.6	0.38127	0.24694	0.64767	0	0

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8	c9
	_____	_____	_____	_____	_____	_____	_____	_____	_____
r1	0	0	0	0	0	0	0	0.014734	0.11626
r2	0	0	0	0	0	0	0	0.014734	0.11626
r3	0	0	0	0	0	0	0	0.014734	0.11626
r4	0	0	0	0	0	0	0	0.014734	0.11626
r5	0	0	0	0	0	0	0	0.014734	0.11626
r746	0.68623	0.68354	0.68095	0.67686	0.67308	0.66722	0.66044	0.65098	0.63987
r747	0.68663	0.68364	0.68119	0.67698	0.6732	0.66734	0.66063	0.65117	0.64009
r748	0.68675	0.6837	0.68129	0.67711	0.67332	0.66749	0.66078	0.65138	0.64027
r749	0.68681	0.68379	0.68141	0.6772	0.67344	0.66764	0.66096	0.65184	0.64048
r750	0.6869	0.68385	0.6815	0.67759	0.67357	0.66777	0.66111	0.65233	0.6407

```
% 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 8.355503 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      4047.5    0.59963    0.28766    0.47974      0

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

    r1          0          0          0          0          0    0.046381    0.17205    0.33791    0.51164
    r2          0          0          0          0          0    0.046381    0.17205    0.33791    0.51164
    r3          0          0          0          0          0    0.046381    0.17205    0.33791    0.51164
    r4          0          0          0          0          0    0.046381    0.17205    0.33791    0.51164
    r5          0          0          0          0          0    0.046381    0.17205    0.33791    0.51164
    r746    0.88633    0.88548    0.88435    0.88337    0.88194     0.88041    0.87852    0.87629    0.87345
    r747    0.88645     0.8856    0.88447    0.88349    0.88206     0.88053    0.87867    0.87644    0.87373
    r748    0.88657    0.88575    0.88459    0.88361    0.88221     0.88068    0.87882    0.87659    0.87406
    r749     0.8867    0.88587    0.88474    0.88377    0.88233     0.88084    0.87897    0.87675    0.87437
    r750    0.88682    0.88599    0.88486    0.88389    0.88248     0.88096     0.8791     0.8769    0.87482
```

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') = 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:

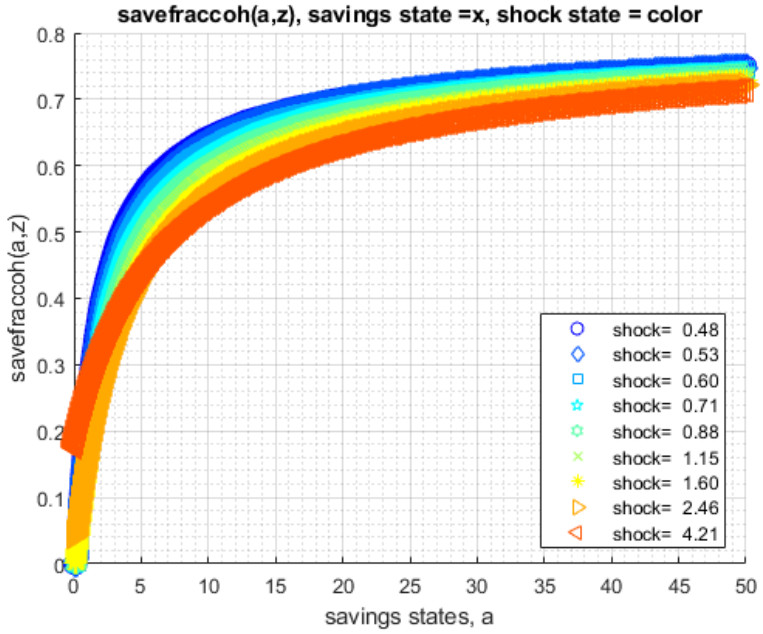
```
% Lower Risk Aversion
mp_params('fl_crra') = 0.5;
ff_vfi_az_bisec_vec(mp_params, mp_support);
```

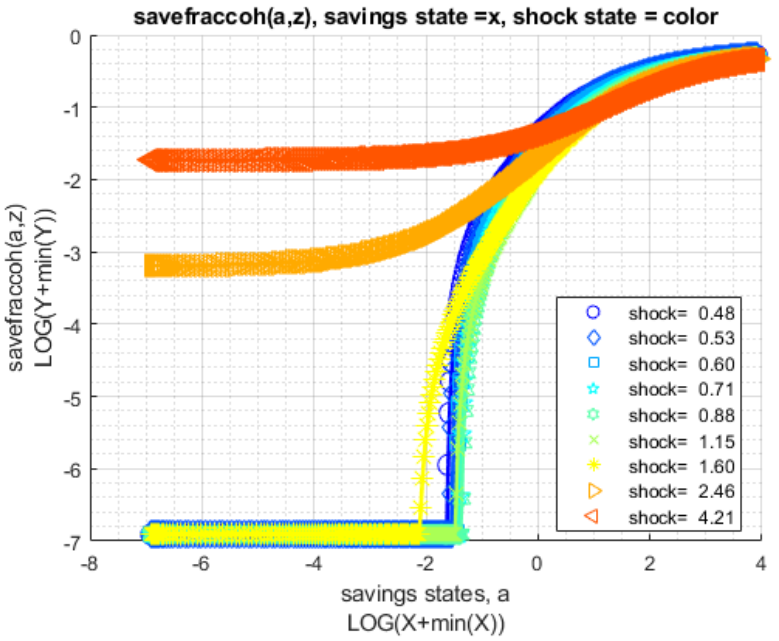
```
Elapsed time is 6.947134 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
                  i    idx    ndim    numel    rowN    colN     sum       mean        std       coefvari    min
```

	_	__	___	____	___	___	_____	_____	_____	_____	__	_
savefraccoh	1	1	2	6750	750	9	2940.8	0.43567	0.26675	0.61228	0	0

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0	0	0.040155	0.17657
r2	0	0	0	0	0	0	0	0.040155	0.17657
r3	0	0	0	0	0	0	0	0.040155	0.17657
r4	0	0	0	0	0	0	0	0.040155	0.17657
r5	0	0	0	0	0	0	0	0.040155	0.17657
r746	0.74928	0.74699	0.74427	0.74165	0.73826	0.73371	0.72828	0.72074	0.71244
r747	0.74949	0.74711	0.7444	0.74195	0.73844	0.73405	0.72847	0.72096	0.71266
r748	0.74958	0.74723	0.74452	0.74226	0.7386	0.73432	0.72865	0.72117	0.71287
r749	0.74971	0.74736	0.74467	0.74241	0.73875	0.73451	0.72883	0.72139	0.71308
r750	0.74983	0.74748	0.74491	0.74253	0.7389	0.73466	0.72905	0.72178	0.7133





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

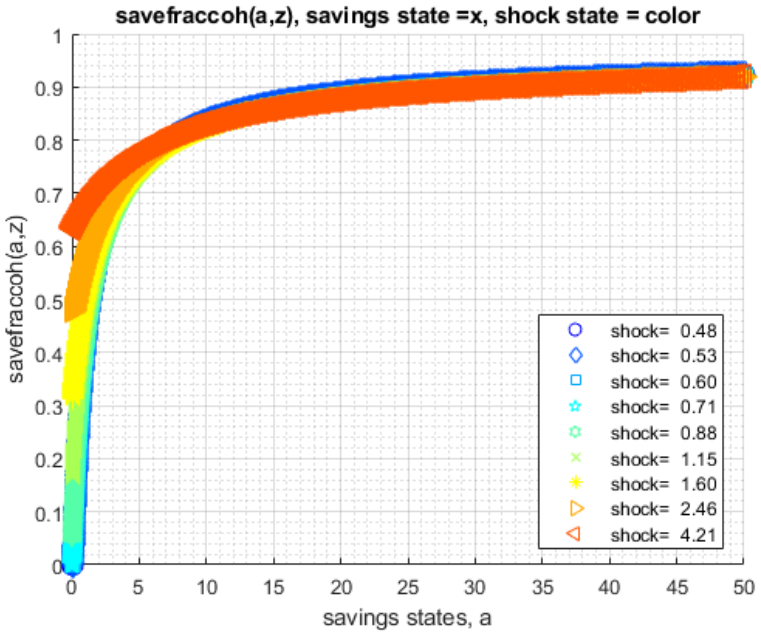
```
% Higher Risk Aversion
mp_params('fl_crra') = 5;
ff_vfi_az_bisec_vec(mp_params, mp_support);
```

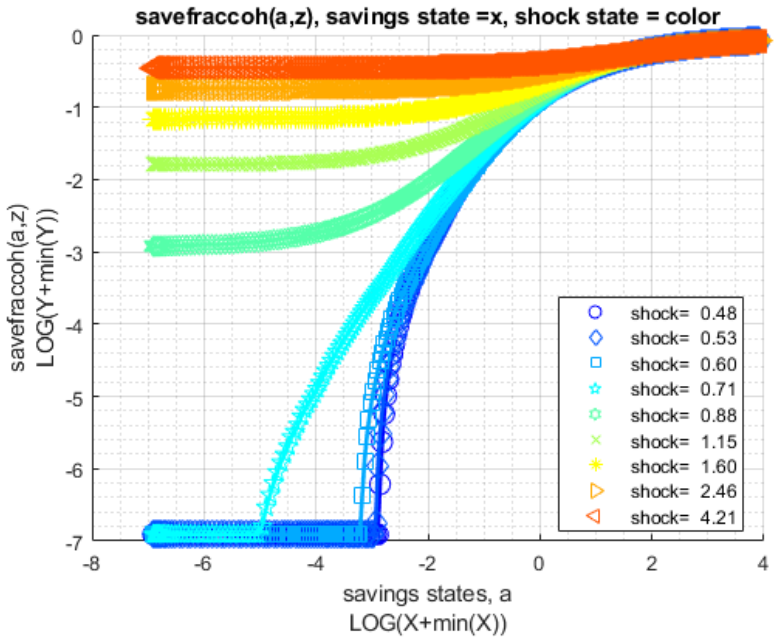
```
Elapsed time is 6.425400 seconds.
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_ffcmd ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari	min	ma
savefraccoh	1	1	2	6750	750	9	4449	0.65911	0.2826	0.42876	0	0.92

xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0.05282	0.16466	0.31347	0.47728	0.63304
r2	0	0	0	0	0.05282	0.16466	0.31347	0.47728	0.63304
r3	0	0	0	0	0.05282	0.16466	0.31347	0.47728	0.63304
r4	0	0	0	0	0.05282	0.16466	0.31347	0.47728	0.63304
r5	0	0	0	0	0.05282	0.16466	0.31347	0.47728	0.63304
r746	0.92341	0.92298	0.92249	0.92176	0.92097	0.9202	0.9191	0.91825	0.91926
r747	0.92353	0.9231	0.92261	0.92188	0.92109	0.92033	0.91923	0.9184	0.91956
r748	0.92365	0.92319	0.92271	0.922	0.92121	0.92045	0.91935	0.91852	0.91987
r749	0.92377	0.92332	0.92283	0.92213	0.92133	0.92057	0.9195	0.91868	0.92014
r750	0.92387	0.92344	0.92295	0.92225	0.92145	0.92069	0.91962	0.9188	0.92045





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

```
Elapsed time is 6.784360 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    3617.7    0.53596    0.31083    0.57996       0

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

r1	0	0	0	0	0	0	0	0.034876	0.095147
r2	0	0	0	0	0	0	0	0.034876	0.095147
r3	0	0	0	0	0	0	0	0.034876	0.095147
r4	0	0	0	0	0	0	0	0.034876	0.095147
r5	0	0	0	0	0	0	0	0.034876	0.095147
r746	0.8642	0.86359	0.86295	0.86192	0.86124	0.8603	0.85944	0.85835	0.85694
r747	0.86436	0.86375	0.86314	0.8621	0.8614	0.86048	0.8596	0.85853	0.85712
r748	0.86451	0.8639	0.86329	0.86225	0.86158	0.8607	0.85978	0.85871	0.85731
r749	0.86466	0.86408	0.86344	0.86243	0.86173	0.86091	0.85996	0.85886	0.85755
r750	0.86482	0.86424	0.86359	0.86259	0.86192	0.86112	0.86012	0.85905	0.85783

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 7.804664 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	4755.4	0.7045	0.26237	0.37241	0	0.

```
xxx TABLE:savefraccoh xxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8	c9
r1	0	0	0	0	0	0.152	0.44643	0.71928	0.92841
r2	0	0	0	0	0	0.152	0.44643	0.71928	0.92841
r3	0	0	0	0	0	0.152	0.44643	0.71928	0.92841
r4	0	0	0	0	0	0.152	0.44643	0.71928	0.92841
r5	0	0	0	0	0	0.152	0.44643	0.71928	0.92841
r746	0.8914	0.89054	0.88944	0.88798	0.88599	0.88279	0.87788	0.87836	0.95118
r747	0.89146	0.8906	0.8895	0.88807	0.88609	0.88288	0.878	0.87879	0.95124
r748	0.89152	0.89066	0.88956	0.88813	0.88615	0.88297	0.87812	0.87919	0.9513
r749	0.89158	0.89072	0.88963	0.88819	0.88624	0.88306	0.87824	0.87962	0.95136
r750	0.89164	0.89079	0.88972	0.88828	0.8863	0.88316	0.87833	0.88001	0.95142