Small Test Exact Solution Looped Minimizer

This is the example vignette for function: **snw_vfi_main** from the **PrjOptiSNW Package.** This function solves for policy function fully iteratively using matlab minimizer. Small Solution Analysis.This produces the same result as **snw_vfi_main_bisec_vec,** except slower. The purpose of this function is to confirm that the results from **snw_vfi_main_bisec_vec** is correct.

Test SNW_VFI_MAIN Defaults Small

Call the function with defaults parameters.

```
mp_param = snw_mp_param('default_small');
[V_VFI,ap_VFI,cons_VFI,mp_valpol_more] = snw_vfi_main(mp_param);
```

```
SNW_VFI_MAIN: Finished Age Group:18 of 18
SNW_VFI_MAIN: Finished Age Group:17 of 18
SNW_VFI_MAIN: Finished Age Group:16 of 18
SNW_VFI_MAIN: Finished Age Group:15 of 18
SNW_VFI_MAIN: Finished Age Group:14 of 18
SNW_VFI_MAIN: Finished Age Group:13 of 18
SNW_VFI_MAIN: Finished Age Group:12 of 18
SNW_VFI_MAIN: Finished Age Group:11 of 18
SNW_VFI_MAIN: Finished Age Group:10 of 18
SNW_VFI_MAIN: Finished Age Group:9 of 18
SNW_VFI_MAIN: Finished Age Group:8 of 18
SNW_VFI_MAIN: Finished Age Group:7 of 18
SNW_VFI_MAIN: Finished Age Group:6 of 18
SNW_VFI_MAIN: Finished Age Group:5 of 18
SNW_VFI_MAIN: Finished Age Group:4 of 18
SNW_VFI_MAIN: Finished Age Group:3 of 18
SNW_VFI_MAIN: Finished Age Group:2 of 18
SNW_VFI_MAIN: Finished Age Group:1 of 18
Elapsed time is 515.239525 seconds.
Completed SNW_VFI_MAIN;SNW_MP_PARAM=default_small;SNW_MP_CONTROL=default_base
```

Small Param Results Define Frames

Define the matrix dimensions names and dimension vector values. Policy and Value Functions share the same ND dimensional structure.

```
% Grids:
age_grid = [19, 22:5:97, 100];
agrid = mp_param('agrid')';
eta_H_grid = mp_param('eta_H_grid')';
eta_S_grid = mp_param('eta_S_grid')';
ar_st_eta_HS_grid = string(cellstr([num2str(eta_H_grid', 'hz=%3.2f;'), num2str(eta_S_grid', 'wz
edu_grid = [0,1];
marry_grid = [0,1];
kids_grid = (1:1:mp_param('n_kidsgrid'))';
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
cl_mp_datasetdesc = {};
cl_mp_datasetdesc{1} = containers.Map({'name', 'labval'}, {'age', age_grid});
cl_mp_datasetdesc{2} = containers.Map({'name', 'labval'}, {'savings', agrid});
cl_mp_datasetdesc{3} = containers.Map({'name', 'labval'}, {'eta', 1:length(eta_H_grid)});
cl_mp_datasetdesc{4} = containers.Map({'name', 'labval'}, {'edu', edu_grid});
```

```
cl_mp_datasetdesc{5} = containers.Map({'name', 'labval'}, {'marry', marry_grid});
cl_mp_datasetdesc{6} = containers.Map({'name', 'labval'}, {'kids', kids_grid});
```

Analyze Savings and Shocks

First, analyze Savings Levels and Shocks, Aggregate Over All Others, and do various other calculations.

```
% Generate some Data
mp_support_graph = containers.Map('KeyType', 'char', 'ValueType', 'any');
mp_support_graph('cl_st_xtitle') = {'Savings States, a'};
mp_support_graph('st_legend_loc') = 'best';
mp_support_graph('bl_graph_logy') = true; % do not log
```

MEAN(VAL(A,Z)), MEAN(AP(A,Z)), MEAN(C(A,Z))

Tabulate value and policies along savings and shocks:

```
% Set
% NaN(n_jgrid,n_agrid,n_etagrid,n_educgrid,n_marriedgrid,n_kidsgrid);
ar_permute = [1,4,5,6,3,2];
% Value Function
tb_az_v = ff_summ_nd_array("MEAN(VAL(A,Z))", V_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, ar
```

```
xxx  MEAN(VAL(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5
1	0	-17.721	-9.4697	-4.6571	-1.7924	-0.23581
2	0.0097656	-17.284	-9.3219	-4.5706	-1.7215	-0.16909
3	0.078125	-15.196	-8.4993	-4.1004	-1.3514	0.17326
4	0.26367	-11.907	-7.0394	-3.3075	-0.78441	0.67194
5	0.625	-8.4194	-5.2786	-2.3615	-0.19487	1.1461
6	1.2207	-5.393	-3.5129	-1.3918	0.35026	1.5392
7	2.1094	-3.0352	-1.9483	-0.50577	0.84352	1.8533
8	3.3496	-1.2918	-0.66899	0.26902	1.2874	2.1081
9	5	-0.030416	0.32906	0.92609	1.6707	2.3215
10	7.1191	0.87699	1.0879	1.4656	1.9934	2.5052
11	9.7656	1.5329	1.6594	1.8992	2.267	2.6661
12	12.998	2.0119	2.0896	2.2435	2.4983	2.8056
13	16.875	2.366	2.4149	2.5152	2.6918	2.9253
14	21.455	2.6312	2.6629	2.7294	2.8524	3.0281
15	26.797	2.8329	2.8539	2.8987	2.9852	3.1169
16	32.959	2.9883	3.0025	3.0334	3.0948	3.1935
17	40	3.1097	3.1195	3.1411	3.1853	3.2595
18	47.979	3.2056	3.2125	3.2279	3.2601	3.3161
19	56.953	3.2822	3.2872	3.2984	3.322	3.3647
20	66.982	3.3441	3.3478	3.356	3.3736	3.4063
21	78.125	3.3947	3.3974	3.4035	3.4168	3.442
22	90.439	3.4363	3.4383	3.4429	3.4531	3.4727
23	103.98	3.4708	3.4724	3.4759	3.4837	3.4991
24	118.82	3.4997	3.5009	3.5036	3.5097	3.5218
25	135	3.5241	3.525	3.5271	3.5319	3.5415

```
% Aprime Choice
tb_az_ap = ff_summ_nd_array("MEAN(AP(A,Z))", ap_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, a
```

```
xxx  MEAN(AP(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5

1	0	3.2168e-05	0.0034996	0.049878	0.24382	0.89299
2	0.0097656	0.00055444	0.0053208	0.053281	0.24787	0.89865
3	0.078125	0.021863	0.029684	0.083029	0.28062	0.93971
4	0.26367	0.13322	0.14773	0.20012	0.3888	1.0591
5	0.625	0.39134	0.41043	0.45332	0.64573	1.3087
6	1.2207	0.84131	0.86393	0.91226	1.0928	1.745
7	2.1094	1.5303	1.5542	1.6156	1.7559	2.3963
8	3.3496	2.4876	2.5118	2.573	2.6876	3.3398
9	5	3.7642	3.7887	3.8498	3.9922	4.592
10	7.1191	5.4275	5.4525	5.5145	5.6929	6.1933
11	9.7656	7.4794	7.5043	7.5679	7.7532	8.1877
12	12.998	9.9124	9.9329	9.9956	10.186	10.627
13	16.875	12.928	12.95	13.005	13.196	13.715
14	21.455	16.529	16.548	16.604	16.783	17.374
15	26.797	20.601	20.618	20.668	20.837	21.462
16	32.959	25.307	25.325	25.37	25.525	26.151
17	40	30.667	30.69	30.742	30.886	31.487
18	47.979	36.761	36.782	36.841	36.999	37.562
19	56.953	43.773	43.795	43.847	44.012	44.56
20	66.982	51.605	51.628	51.688	51.85	52.403
21	78.125	59.955	59.978	60.038	60.211	60.768
22	90.439	69.267	69.29	69.352	69.528	70.097
23	103.98	79.753	79.774	79.834	80.008	80.586
24	118.82	91.116	91.14	91.201	91.367	91.942
25	135	103.47	103.49	103.55	103.72	104.29

```
% Consumption Choices
tb_az_c = ff_summ_nd_array("MEAN(C(A,Z))", cons_VFI, true, ["mean"], 4, 1, cl_mp_datasetdesc, a
```

```
xxx  MEAN(C(A,Z))  xxxxxxxxxxxxxxxxxxxxxxxxxx
```

group	savings	mean_eta_1	mean_eta_2	mean_eta_3	mean_eta_4	mean_eta_5
1	0	0.30273	0.43104	0.68779	1.2165	2.3367
2	0.0097656	0.31374	0.44069	0.69581	1.2239	2.3424
3	0.078125	0.37308	0.49662	0.74605	1.2709	2.3811
4	0.26367	0.48039	0.59638	0.846	1.3793	2.478
5	0.625	0.64735	0.75736	1.0152	1.5439	2.6496
6	1.2207	0.89648	1.0013	1.2519	1.7913	2.9071
7	2.1094	1.2479	1.3498	1.5854	2.1634	3.2903
8	3.3496	1.7393	1.8394	2.0734	2.6754	3.7896
9	5	2.3872	2.4859	2.7182	3.2909	4.4564
10	7.1191	3.1917	3.289	3.5191	4.0542	5.3181
11	9.7656	4.2188	4.3155	4.543	5.07	6.3986
12	12.998	5.5439	5.6447	5.8722	6.3933	7.7142
13	16.875	7.0334	7.133	7.3676	7.8866	9.1285
14	21.455	8.754	8.8551	9.0887	9.6188	10.789
15	26.797	10.886	10.989	11.228	11.768	12.903
16	32.959	13.336	13.438	13.682	14.235	15.368
17	40	16.151	16.249	16.485	17.049	18.207
18	47.979	19.321	19.42	19.649	20.2	21.394
19	56.953	22.728	22.827	23.062	23.605	24.816
20	66.982	26.539	26.636	26.864	27.41	28.615
21	78.125	31.124	31.221	31.45	31.985	33.186
22	90.439	36.108	36.205	36.431	36.963	38.152
23	103.98	41.345	41.444	41.673	42.206	43.386
24	118.82	47.202	47.298	47.525	48.066	49.248
25	135	53.632	53.731	53.962	54.496	55.685