FF_OPTIM_MZOOM_SAVEZRONE Derivative Multisection

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

This is the example vignette for function: **ff_optim_mzoom_savezrone** from the **MEconTools Package**. This functions solves for optimal savings/borrowing level given an anonymous function that provides the utility (not derivative) of a intertemporal savings problem. This is a vectorized function solves for multiple state-space elements at the same time. The function allows for controls of iteration counts, the number of evaluations per iteration, and how much to "zoom-in" for each iteration around the last iteration's maximum/optimal choice.

Note that if first order conditions are available this method should not be used, but **ff_optim_mlsec_savezrone** should be used. **ff_optim_mlsec_savezrone** relies on bisection. In the first example below more _it_mzoom_jnt_pnts_ values are needed to achieve the same precision than under **ff_optim_mlsec_savezrone**. However, increasing _it_mzoom_jnt_pnts_ might not expensive given vectorization, should increase time cost linearly in generally. MZOOM is much more robust than bisection based methods. And by increasing the number of points evaluated per iteration, in limited number of iterations, the approximately exact optimal savings choice can be found.

The vectorized zooming savings problem rely on this function to solve for optimal savings choices:

- States Grid + Approximate Continuous Exact Savings (zoom) as Share of Cash-on-Hand **_Vectorized_**: **ff_vfi_az_zoom_vec**, precision and speed

Test FF_OPTIM_MZOOM_SAVEZRONE One Individual

Bisection for savings choice at one state:

```
% Generate the state-space and function
[fl_z1, fl_z2, fl_r, fl_beta] = deal(0.4730, 0.6252, 0.0839, 0.7365);
% ffi_intertemporal_max is a function in ff_optim_mlsec_savezrone for testing
fc_util = @(x) ffi_intertemporal_util(x, fl_z1, fl_z2, fl_r, fl_beta);
% Call Function
bl_verbose = false;
bl_timer = true;
% optimally borrowing given the parameters here
mp_mzoom_ctrlinfo = containers.Map('KeyType','char', 'ValueType','any');
mp_mzoom_ctrlinfo('it_mzoom_jnt_pnts') = 15;
mp_mzoom_ctrlinfo('it_mzoom_max_iter') = 10;
mp_mzoom_ctrlinfo('it_mzoom_zm_ratio') = 0.25;
[fl_opti_save_frac, fl_opti_save_level] = ...
    ff_optim_mzoom_savezrone(fc_util, bl_verbose, bl_timer, mp_mzoom_ctrlinfo)
```

```
Elapsed time is 0.011586 seconds.
fl_opti_save_frac = 0.4241
fl_opti_save_level = -0.1316
```

Test FF_OPTIM_MZOOM_SAVEZRONE 4 Individuals 3 Iterations 50 Points Per Iteration

5 grid points per iteration, and 5 iterations.

```matlab
% Generate the state-space and function
rng(123);
it_draws = 4; % must be even number
ar_z1 = exp(rand([it_draws,1])*3-1.5);
ar_z2 = exp(rand([it_draws,1])*3-1.5);
ar_r = (rand(it_draws,1)*10.0);
ar_beta = [rand(round(it_draws/2),1)*0.9+0.1; rand(round(it_draws/2),1)*0.9+1];
fc_util = @(x) ffi_intertemporal_util(x, ar_z1, ar_z2, ar_r, ar_beta);
% Call Function
bl_verbose = true;
bl_timer = true;
mp_mzoom_ctrlinfo = containers.Map('KeyType','char', 'ValueType','any');
mp_mzoom_ctrlinfo('it_mzoom_jnt_pnts') = 50;
mp_mzoom_ctrlinfo('it_mzoom_max_iter') = 3;
mp_mzoom_ctrlinfo('it_mzoom_zm_ratio') = 0;
[fl_opti_save_frac, fl_opti_save_level] = ...
    ff_optim_mzoom_savezrone(fc_util, bl_verbose, bl_timer, mp_mzoom_ctrlinfo);
```
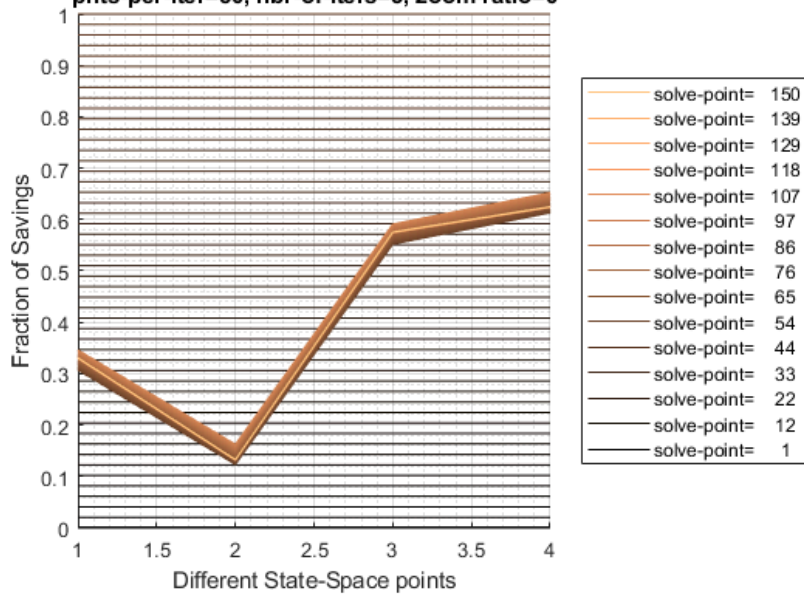
iter	cl_row_names_a	Var1	Var2	Var3	Var4
1	"point=1"	1e-05	1e-05	1e-05	1e-05
1	"point=2"	0.020418	0.020418	0.020418	0.020418
1	"point=3"	0.040826	0.040826	0.040826	0.040826
1	"point=4"	0.061233	0.061233	0.061233	0.061233
1	"point=5"	0.081641	0.081641	0.081641	0.081641
1	"point=6"	0.10205	0.10205	0.10205	0.10205
1	"point=7"	0.12246	0.12246	0.12246	0.12246
1	"point=8"	0.14286	0.14286	0.14286	0.14286
1	"point=9"	0.16327	0.16327	0.16327	0.16327
1	"point=10"	0.18368	0.18368	0.18368	0.18368
1	"point=11"	0.20409	0.20409	0.20409	0.20409
1	"point=12"	0.2245	0.2245	0.2245	0.2245
1	"point=13"	0.2449	0.2449	0.2449	0.2449
1	"point=14"	0.26531	0.26531	0.26531	0.26531
1	"point=15"	0.28572	0.28572	0.28572	0.28572
1	"point=16"	0.30613	0.30613	0.30613	0.30613
1	"point=17"	0.32653	0.32653	0.32653	0.32653
1	"point=18"	0.34694	0.34694	0.34694	0.34694
1	"point=19"	0.36735	0.36735	0.36735	0.36735
1	"point=20"	0.38776	0.38776	0.38776	0.38776
1	"point=21"	0.40817	0.40817	0.40817	0.40817
1	"point=22"	0.42857	0.42857	0.42857	0.42857
1	"point=23"	0.44898	0.44898	0.44898	0.44898
1	"point=24"	0.46939	0.46939	0.46939	0.46939
1	"point=25"	0.4898	0.4898	0.4898	0.4898
1	"point=26"	0.5102	0.5102	0.5102	0.5102
1	"point=27"	0.53061	0.53061	0.53061	0.53061
1	"point=28"	0.55102	0.55102	0.55102	0.55102
1	"point=29"	0.57143	0.57143	0.57143	0.57143
1	"point=30"	0.59183	0.59183	0.59183	0.59183
1	"point=31"	0.61224	0.61224	0.61224	0.61224
1	"point=32"	0.63265	0.63265	0.63265	0.63265
1	"point=33"	0.65306	0.65306	0.65306	0.65306
1	"point=34"	0.67347	0.67347	0.67347	0.67347
1	"point=35"	0.69387	0.69387	0.69387	0.69387
1	"point=36"	0.71428	0.71428	0.71428	0.71428
1	"point=37"	0.73469	0.73469	0.73469	0.73469
1	"point=38"	0.7551	0.7551	0.7551	0.7551
1	"point=39"	0.7755	0.7755	0.7755	0.7755
1	"point=40"	0.79591	0.79591	0.79591	0.79591

1	"point=41"	0.81632	0.81632	0.81632	0.81632
1	"point=42"	0.83673	0.83673	0.83673	0.83673
1	"point=43"	0.85714	0.85714	0.85714	0.85714
1	"point=44"	0.87754	0.87754	0.87754	0.87754
1	"point=45"	0.89795	0.89795	0.89795	0.89795
1	"point=46"	0.91836	0.91836	0.91836	0.91836
1	"point=47"	0.93877	0.93877	0.93877	0.93877
1	"point=48"	0.95917	0.95917	0.95917	0.95917
1	"point=49"	0.97958	0.97958	0.97958	0.97958
1	"point=50"	0.99999	0.99999	0.99999	0.99999
2	"point=1"	0.30693	0.12326	0.55182	0.61304
2	"point=2"	0.30773	0.12406	0.55262	0.61384
2	"point=3"	0.30853	0.12486	0.55342	0.61464
2	"point=4"	0.30933	0.12566	0.55422	0.61544
2	"point=5"	0.31013	0.12646	0.55502	0.61624
2	"point=6"	0.31093	0.12726	0.55582	0.61704
2	"point=7"	0.31173	0.12806	0.55662	0.61784
2	"point=8"	0.31253	0.12886	0.55742	0.61865
2	"point=9"	0.31333	0.12966	0.55822	0.61945
2	"point=10"	0.31413	0.13046	0.55902	0.62025
2	"point=11"	0.31493	0.13126	0.55982	0.62105
2	"point=12"	0.31573	0.13206	0.56062	0.62185
2	"point=13"	0.31653	0.13286	0.56142	0.62265
2	"point=14"	0.31733	0.13366	0.56222	0.62345
2	"point=15"	0.31813	0.13446	0.56302	0.62425
2	"point=16"	0.31893	0.13526	0.56382	0.62505
2	"point=17"	0.31973	0.13606	0.56462	0.62585
2	"point=18"	0.32053	0.13686	0.56542	0.62665
2	"point=19"	0.32133	0.13766	0.56623	0.62745
2	"point=20"	0.32213	0.13846	0.56703	0.62825
2	"point=21"	0.32293	0.13926	0.56783	0.62905
2	"point=22"	0.32373	0.14006	0.56863	0.62985
2	"point=23"	0.32453	0.14086	0.56943	0.63065
2	"point=24"	0.32533	0.14166	0.57023	0.63145
2	"point=25"	0.32613	0.14246	0.57103	0.63225
2	"point=26"	0.32693	0.14326	0.57183	0.63305
2	"point=27"	0.32773	0.14406	0.57263	0.63385
2	"point=28"	0.32853	0.14487	0.57343	0.63465
2	"point=29"	0.32934	0.14567	0.57423	0.63545
2	"point=30"	0.33014	0.14647	0.57503	0.63625
2	"point=31"	0.33094	0.14727	0.57583	0.63705
2	"point=32"	0.33174	0.14807	0.57663	0.63785
2	"point=33"	0.33254	0.14887	0.57743	0.63865
2	"point=34"	0.33334	0.14967	0.57823	0.63945
2	"point=35"	0.33414	0.15047	0.57903	0.64025
2	"point=36"	0.33494	0.15127	0.57983	0.64105
2	"point=37"	0.33574	0.15207	0.58063	0.64185
2	"point=38"	0.33654	0.15287	0.58143	0.64265
2	"point=39"	0.33734	0.15367	0.58223	0.64345
2	"point=40"	0.33814	0.15447	0.58303	0.64425
2	"point=41"	0.33894	0.15527	0.58383	0.64506
2	"point=42"	0.33974	0.15607	0.58463	0.64586
2	"point=43"	0.34054	0.15687	0.58543	0.64666
2	"point=44"	0.34134	0.15767	0.58623	0.64746
2	"point=45"	0.34214	0.15847	0.58703	0.64826
2	"point=46"	0.34294	0.15927	0.58783	0.64906
2	"point=47"	0.34374	0.16007	0.58863	0.64986
2	"point=48"	0.34454	0.16087	0.58943	0.65066
2	"point=49"	0.34534	0.16167	0.59023	0.65146
2	"point=50"	0.34614	0.16247	0.59103	0.65226
3	"point=1"	0.32937	0.13129	0.57426	0.62348
3	"point=2"	0.3294	0.13132	0.57429	0.62351
3	"point=3"	0.32943	0.13135	0.57432	0.62354
3	"point=4"	0.32946	0.13139	0.57435	0.62357
3	"point=5"	0.32949	0.13142	0.57439	0.6236

3	"point=6"	0.32952	0.13145	0.57442	0.62364
3	"point=7"	0.32955	0.13148	0.57445	0.62367
3	"point=8"	0.32959	0.13151	0.57448	0.6237
3	"point=9"	0.32962	0.13154	0.57451	0.62373
3	"point=10"	0.32965	0.13157	0.57454	0.62376
3	"point=11"	0.32968	0.13161	0.57457	0.62379
3	"point=12"	0.32971	0.13164	0.5746	0.62382
3	"point=13"	0.32974	0.13167	0.57464	0.62385
3	"point=14"	0.32977	0.1317	0.57467	0.62389
3	"point=15"	0.32981	0.13173	0.5747	0.62392
3	"point=16"	0.32984	0.13176	0.57473	0.62395
3	"point=17"	0.32987	0.13179	0.57476	0.62398
3	"point=18"	0.3299	0.13182	0.57479	0.62401
3	"point=19"	0.32993	0.13186	0.57482	0.62404
3	"point=20"	0.32996	0.13189	0.57486	0.62407
3	"point=21"	0.32999	0.13192	0.57489	0.62411
3	"point=22"	0.33003	0.13195	0.57492	0.62414
3	"point=23"	0.33006	0.13198	0.57495	0.62417
3	"point=24"	0.33009	0.13201	0.57498	0.6242
3	"point=25"	0.33012	0.13204	0.57501	0.62423
3	"point=26"	0.33015	0.13208	0.57504	0.62426
3	"point=27"	0.33018	0.13211	0.57508	0.62429
3	"point=28"	0.33021	0.13214	0.57511	0.62433
3	"point=29"	0.33025	0.13217	0.57514	0.62436
3	"point=30"	0.33028	0.1322	0.57517	0.62439
3	"point=31"	0.33031	0.13223	0.5752	0.62442
3	"point=32"	0.33034	0.13226	0.57523	0.62445
3	"point=33"	0.33037	0.1323	0.57526	0.62448
3	"point=34"	0.3304	0.13233	0.5753	0.62451
3	"point=35"	0.33043	0.13236	0.57533	0.62455
3	"point=36"	0.33046	0.13239	0.57536	0.62458
3	"point=37"	0.3305	0.13242	0.57539	0.62461
3	"point=38"	0.33053	0.13245	0.57542	0.62464
3	"point=39"	0.33056	0.13248	0.57545	0.62467
3	"point=40"	0.33059	0.13252	0.57548	0.6247
3	"point=41"	0.33062	0.13255	0.57551	0.62473
3	"point=42"	0.33065	0.13258	0.57555	0.62477
3	"point=43"	0.33068	0.13261	0.57558	0.6248
3	"point=44"	0.33072	0.13264	0.57561	0.62483
3	"point=45"	0.33075	0.13267	0.57564	0.62486
3	"point=46"	0.33078	0.1327	0.57567	0.62489
3	"point=47"	0.33081	0.13273	0.5757	0.62492
3	"point=48"	0.33084	0.13277	0.57573	0.62495
3	"point=49"	0.33087	0.1328	0.57577	0.62498
3	"point=50"	0.3309	0.13283	0.5758	0.62502

Vectorized Exact Zooming Optimization, Savings Fractions

pnts-per-iter=50, nbr-of-iters=3, zoom-ratio=0

iter	cl_row_names_a	Var1	Var2	Var3	Var4
1	"point=1"	-3.6912	-1.9565	-12.83	-14.789
1	"point=2"	0.058694	-0.80561	-2.4984	-2.1254
1	"point=3"	0.38043	-0.72015	-1.5784	-0.99337
1	"point=4"	0.55947	-0.67935	-1.0493	-0.34024
1	"point=5"	0.67979	-0.65711	-0.68055	0.11647
1	"point=6"	0.7677	-0.64529	-0.39997	0.46531
1	"point=7"	0.8349	-0.64026	-0.17534	0.74571
1	"point=8"	0.88763	-0.6401	0.010483	0.9787
1	"point=9"	0.92959	-0.64367	0.16774	1.1768
1	"point=10"	0.96316	-0.65026	0.30302	1.3481
1	"point=11"	0.98996	-0.65938	0.4208	1.4981
1	"point=12"	1.0111	-0.67071	0.52427	1.6308
1	"point=13"	1.0275	-0.684	0.61578	1.7489
1	"point=14"	1.0397	-0.6991	0.69709	1.8547
1	"point=15"	1.0482	-0.71588	0.76958	1.9499
1	"point=16"	1.0533	-0.73426	0.83429	2.0357
1	"point=17"	1.0554	-0.75419	0.8921	2.1132
1	"point=18"	1.0546	-0.77564	0.94367	2.1833
1	"point=19"	1.0512	-0.79861	0.98955	2.2467
1	"point=20"	1.0451	-0.82309	1.0302	2.3039
1	"point=21"	1.0366	-0.8491	1.066	2.3554
1	"point=22"	1.0256	-0.87669	1.0971	2.4015
1	"point=23"	1.0123	-0.90591	1.124	2.4425
1	"point=24"	0.99654	-0.93682	1.1466	2.4788
1	"point=25"	0.97838	-0.9695	1.1652	2.5104
1	"point=26"	0.95775	-1.004	1.1798	2.5375
1	"point=27"	0.93459	-1.0406	1.1905	2.5602
1	"point=28"	0.90881	-1.0792	1.1973	2.5785
1	"point=29"	0.88029	-1.1202	1.2002	2.5925
1	"point=30"	0.84886	-1.1635	1.1991	2.6022
1	"point=31"	0.81434	-1.2096	1.1938	2.6073
1	"point=32"	0.77649	-1.2587	1.1843	2.6078
1	"point=33"	0.73504	-1.3109	1.1703	2.6035
1	"point=34"	0.68964	-1.3668	1.1514	2.594
1	"point=35"	0.63987	-1.4268	1.1274	2.5792
1	"point=36"	0.58522	-1.4913	1.0978	2.5584
1	"point=37"	0.52505	-1.5611	1.062	2.5312
1	"point=38"	0.45857	-1.6369	1.0192	2.4968
1	"point=39"	0.38475	-1.7198	0.96837	2.4541

1	"point=40"	0.3023	-1.8111	0.90834	2.4021
1	"point=41"	0.20947	-1.9126	0.83737	2.3388
1	"point=42"	0.10391	-2.0266	0.75313	2.2622
1	"point=43"	-0.017693	-2.1564	0.65234	2.1687
1	"point=44"	-0.16019	-2.3069	0.53016	2.0538
1	"point=45"	-0.33112	-2.4857	0.37908	1.9097
1	"point=46"	-0.54312	-2.7054	0.18649	1.724
1	"point=47"	-0.81989	-2.9896	-0.071303	1.4729
1	"point=48"	-1.2146	-3.3917	-0.44748	1.1033
1	"point=49"	-1.8971	-4.0814	-1.1118	0.44547
1	"point=50"	-9.5085	-11.7	-8.7054	-7.1418
2	"point=1"	1.0535	-0.64017	1.1975	2.6074
2	"point=2"	1.0536	-0.64009	1.1977	2.6075
2	"point=3"	1.0537	-0.64001	1.1979	2.6076
2	"point=4"	1.0539	-0.63995	1.198	2.6077
2	"point=5"	1.054	-0.63989	1.1982	2.6077
2	"point=6"	1.0541	-0.63983	1.1984	2.6078
2	"point=7"	1.0542	-0.63979	1.1985	2.6079
2	"point=8"	1.0543	-0.63975	1.1986	2.6079
2	"point=9"	1.0544	-0.63971	1.1988	2.608
2	"point=10"	1.0545	-0.63969	1.1989	2.608
2	"point=11"	1.0546	-0.63967	1.199	2.6081
2	"point=12"	1.0547	-0.63966	1.1992	2.6081
2	"point=13"	1.0548	-0.63965	1.1993	2.6081
2	"point=14"	1.0548	-0.63965	1.1994	2.6081
2	"point=15"	1.0549	-0.63966	1.1995	2.6081
2	"point=16"	1.055	-0.63967	1.1996	2.6081
2	"point=17"	1.0551	-0.63969	1.1997	2.6081
2	"point=18"	1.0551	-0.63971	1.1998	2.6081
2	"point=19"	1.0552	-0.63975	1.1998	2.6081
2	"point=20"	1.0552	-0.63978	1.1999	2.6081
2	"point=21"	1.0553	-0.63983	1.2	2.608
2	"point=22"	1.0553	-0.63988	1.2	2.608
2	"point=23"	1.0553	-0.63993	1.2001	2.6079
2	"point=24"	1.0554	-0.63999	1.2001	2.6079
2	"point=25"	1.0554	-0.64006	1.2002	2.6078
2	"point=26"	1.0554	-0.64013	1.2002	2.6077
2	"point=27"	1.0555	-0.64021	1.2002	2.6077
2	"point=28"	1.0555	-0.64029	1.2003	2.6076
2	"point=29"	1.0555	-0.64038	1.2003	2.6075
2	"point=30"	1.0555	-0.64048	1.2003	2.6074
2	"point=31"	1.0555	-0.64058	1.2003	2.6073
2	"point=32"	1.0555	-0.64069	1.2003	2.6071
2	"point=33"	1.0555	-0.6408	1.2003	2.607
2	"point=34"	1.0555	-0.64091	1.2003	2.6069
2	"point=35"	1.0555	-0.64104	1.2002	2.6067
2	"point=36"	1.0554	-0.64116	1.2002	2.6066
2	"point=37"	1.0554	-0.64129	1.2002	2.6064
2	"point=38"	1.0554	-0.64143	1.2001	2.6063
2	"point=39"	1.0554	-0.64157	1.2001	2.6061
2	"point=40"	1.0553	-0.64172	1.2001	2.6059
2	"point=41"	1.0553	-0.64188	1.2	2.6057
2	"point=42"	1.0552	-0.64203	1.1999	2.6056
2	"point=43"	1.0552	-0.6422	1.1999	2.6053
2	"point=44"	1.0551	-0.64236	1.1998	2.6051
2	"point=45"	1.0551	-0.64254	1.1997	2.6049
2	"point=46"	1.055	-0.64271	1.1996	2.6047
2	"point=47"	1.0549	-0.64289	1.1995	2.6045
2	"point=48"	1.0549	-0.64308	1.1994	2.6042
2	"point=49"	1.0548	-0.64327	1.1993	2.604
2	"point=50"	1.0547	-0.64347	1.1992	2.6037
3	"point=1"	1.0555	-0.63967	1.2003	2.6081
3	"point=2"	1.0555	-0.63967	1.2003	2.6081
3	"point=3"	1.0555	-0.63967	1.2003	2.6081
3	"point=4"	1.0555	-0.63967	1.2003	2.6081

3	"point=5"	1.0555	-0.63967	1.2003	2.6081
3	"point=6"	1.0555	-0.63967	1.2003	2.6081
3	"point=7"	1.0555	-0.63967	1.2003	2.6081
3	"point=8"	1.0555	-0.63966	1.2003	2.6081
3	"point=9"	1.0555	-0.63966	1.2003	2.6081
3	"point=10"	1.0555	-0.63966	1.2003	2.6081
3	"point=11"	1.0555	-0.63966	1.2003	2.6081
3	"point=12"	1.0555	-0.63966	1.2003	2.6081
3	"point=13"	1.0555	-0.63966	1.2003	2.6081
3	"point=14"	1.0555	-0.63966	1.2003	2.6081
3	"point=15"	1.0555	-0.63966	1.2003	2.6081
3	"point=16"	1.0555	-0.63966	1.2003	2.6081
3	"point=17"	1.0555	-0.63966	1.2003	2.6081
3	"point=18"	1.0555	-0.63966	1.2003	2.6081
3	"point=19"	1.0555	-0.63966	1.2003	2.6081
3	"point=20"	1.0555	-0.63966	1.2003	2.6081
3	"point=21"	1.0555	-0.63966	1.2003	2.6081
3	"point=22"	1.0555	-0.63966	1.2003	2.6081
3	"point=23"	1.0555	-0.63966	1.2003	2.6081
3	"point=24"	1.0555	-0.63966	1.2003	2.6081
3	"point=25"	1.0555	-0.63966	1.2003	2.6081
3	"point=26"	1.0555	-0.63966	1.2003	2.6081
3	"point=27"	1.0555	-0.63966	1.2003	2.6081
3	"point=28"	1.0555	-0.63966	1.2003	2.6081
3	"point=29"	1.0555	-0.63966	1.2003	2.6081
3	"point=30"	1.0555	-0.63966	1.2003	2.6081
3	"point=31"	1.0555	-0.63965	1.2003	2.6081
3	"point=32"	1.0555	-0.63965	1.2003	2.6081
3	"point=33"	1.0555	-0.63965	1.2003	2.6081
3	"point=34"	1.0555	-0.63965	1.2003	2.6081
3	"point=35"	1.0555	-0.63965	1.2003	2.6081
3	"point=36"	1.0555	-0.63965	1.2003	2.6081
3	"point=37"	1.0555	-0.63965	1.2003	2.6081
3	"point=38"	1.0555	-0.63965	1.2003	2.6081
3	"point=39"	1.0555	-0.63965	1.2003	2.6081
3	"point=40"	1.0555	-0.63965	1.2003	2.6081
3	"point=41"	1.0555	-0.63965	1.2003	2.6081
3	"point=42"	1.0555	-0.63965	1.2003	2.6081
3	"point=43"	1.0555	-0.63965	1.2003	2.6081
3	"point=44"	1.0555	-0.63965	1.2003	2.6081
3	"point=45"	1.0555	-0.63965	1.2003	2.6081
3	"point=46"	1.0555	-0.63965	1.2003	2.6081
3	"point=47"	1.0555	-0.63965	1.2003	2.6081
3	"point=48"	1.0555	-0.63965	1.2003	2.6081
3	"point=49"	1.0555	-0.63965	1.2003	2.6081
3	"point=50"	1.0555	-0.63965	1.2003	2.6081

Vectorized Exact Zooming Optimization, U(save)
pnts-per-iter=50, nbr-of-iters=3, zoom-ratio=0

Legend:
- solve-point= 150
- solve-point= 139
- solve-point= 129
- solve-point= 118
- solve-point= 107
- solve-point= 97
- solve-point= 86
- solve-point= 76
- solve-point= 65
- solve-point= 54
- solve-point= 44
- solve-point= 33
- solve-point= 22
- solve-point= 12
- solve-point= 1

Y-axis: Utility at Current Savings Fractions
X-axis: Different State-Space points

```
Elapsed time is 1.304303 seconds.
-----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari
	_	___	____	_____	____	____	_____	_____	_____	_____
ar_opti_foc_obj	1	1	2	4	1	4	4.2243	1.0561	1.3298	1.2592
ar_opti_save_frac	2	2	2	4	4	1	1.664	0.416	0.2284	0.54904
ar_opti_save_level	3	3	2	4	1	4	0.80247	0.20062	0.37807	1.8845

```
xxx TABLE:ar_opti_foc_obj xxxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4
	_____	_____	_____	_____
r1	1.0555	-0.63965	1.2003	2.6081

```
xxx TABLE:ar_opti_save_frac xxxxxxxxxxxxxxxxxxx
```

	c1
	_____
r1	0.33086
r2	0.13278
r3	0.57575
r4	0.62461

```
xxx TABLE:ar_opti_save_level xxxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4
	_____	_____	_____	_____
r1	0.37401	-0.070015	-0.15125	0.64972

Test FF_OPTIM_MZOOM_SAVEZRONE 8 Individuals 3 Iterations 10 Points Per Iteration, 0.25 zoom in ratio

10 grid points per iteration, and 3 iterations.

```
% Generate the state-space and function
rng(123);
```

```
it_draws = 8; % must be even number
ar_z1 = exp(rand([it_draws,1])*3-1.5);
ar_z2 = exp(rand([it_draws,1])*3-1.5);
ar_r = (rand(it_draws,1)*10.0);
ar_beta = [rand(round(it_draws/2),1)*0.9+0.1; rand(round(it_draws/2),1)*0.9+1];
fc_util = @(x) ffi_intertemporal_util(x, ar_z1, ar_z2, ar_r, ar_beta);
% Call Function
bl_verbose = true;
bl_timer = true;
mp_mzoom_ctrlinfo = containers.Map('KeyType','char', 'ValueType','any');
mp_mzoom_ctrlinfo('it_mzoom_jnt_pnts') = 10;
mp_mzoom_ctrlinfo('it_mzoom_max_iter') = 3;
mp_mzoom_ctrlinfo('it_mzoom_zm_ratio') = 0.25;
[fl_opti_save_frac, fl_opti_save_level] = ...
    ff_optim_mzoom_savezrone(fc_util, bl_verbose, bl_timer, mp_mzoom_ctrlinfo);
```
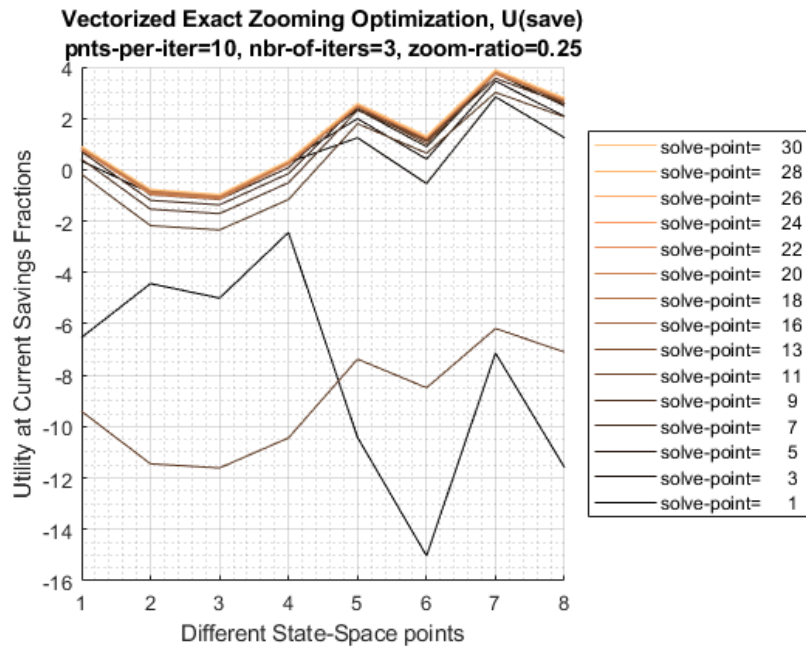
iter	cl_row_names_a	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8
1	"point=1"	1e-05	1e-05	1e-05	1e-05	1e-05	1e-05	1e-05	1e-05
1	"point=2"	0.11112	0.11112	0.11112	0.11112	0.11112	0.11112	0.11112	0.11112
1	"point=3"	0.22223	0.22223	0.22223	0.22223	0.22223	0.22223	0.22223	0.22223
1	"point=4"	0.33334	0.33334	0.33334	0.33334	0.33334	0.33334	0.33334	0.33334
1	"point=5"	0.44445	0.44445	0.44445	0.44445	0.44445	0.44445	0.44445	0.44445
1	"point=6"	0.55555	0.55555	0.55555	0.55555	0.55555	0.55555	0.55555	0.55555
1	"point=7"	0.66666	0.66666	0.66666	0.66666	0.66666	0.66666	0.66666	0.66666
1	"point=8"	0.77777	0.77777	0.77777	0.77777	0.77777	0.77777	0.77777	0.77777
1	"point=9"	0.88888	0.88888	0.88888	0.88888	0.88888	0.88888	0.88888	0.88888
1	"point=10"	0.99999	0.99999	0.99999	0.99999	0.99999	0.99999	0.99999	0.99999
2	"point=1"	0.28788	0.20455	0.20455	0.12122	0.37121	0.37121	0.37121	0.37121
2	"point=2"	0.32576	0.24243	0.24243	0.1591	0.40909	0.40909	0.40909	0.40909
2	"point=3"	0.36364	0.28031	0.28031	0.19698	0.44697	0.44697	0.44697	0.44697
2	"point=4"	0.40152	0.31819	0.31819	0.23485	0.48485	0.48485	0.48485	0.48485
2	"point=5"	0.4394	0.35606	0.35606	0.27273	0.52273	0.52273	0.52273	0.52273
2	"point=6"	0.47727	0.39394	0.39394	0.31061	0.5606	0.5606	0.5606	0.5606
2	"point=7"	0.51515	0.43182	0.43182	0.34849	0.59848	0.59848	0.59848	0.59848
2	"point=8"	0.55303	0.4697	0.4697	0.38637	0.63636	0.63636	0.63636	0.63636
2	"point=9"	0.59091	0.50758	0.50758	0.42424	0.67424	0.67424	0.67424	0.67424
2	"point=10"	0.62879	0.54545	0.54545	0.46212	0.71212	0.71212	0.71212	0.71212
3	"point=1"	0.34987	0.20972	0.20972	0.15479	0.46161	0.49001	0.4332	0.49001
3	"point=2"	0.3645	0.22435	0.22435	0.16943	0.47624	0.50465	0.44783	0.50465
3	"point=3"	0.37913	0.23899	0.23899	0.18406	0.49087	0.51928	0.46247	0.51928
3	"point=4"	0.39377	0.25362	0.25362	0.1987	0.50551	0.53392	0.4771	0.53392
3	"point=5"	0.4084	0.26826	0.26826	0.21333	0.52014	0.54855	0.49174	0.54855
3	"point=6"	0.42304	0.28289	0.28289	0.22797	0.53478	0.56319	0.50637	0.56319
3	"point=7"	0.43767	0.29752	0.29752	0.2426	0.54941	0.57782	0.52101	0.57782
3	"point=8"	0.45231	0.31216	0.31216	0.25724	0.56405	0.59246	0.53564	0.59246
3	"point=9"	0.46694	0.32679	0.32679	0.27187	0.57868	0.60709	0.55027	0.60709
3	"point=10"	0.48158	0.34143	0.34143	0.28651	0.59332	0.62173	0.56491	0.62173

Vectorized Exact Zooming Optimization, Savings Fractions
pnts-per-iter=10, nbr-of-iters=3, zoom-ratio=0.25
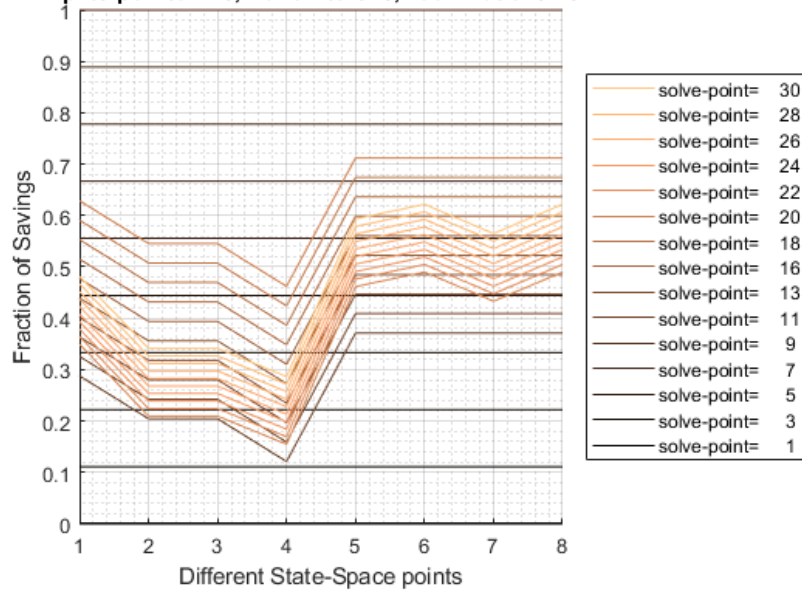
iter	cl_row_names_a	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var
1	"point=1"	-6.5286	-4.4312	-4.9951	-2.4407	-10.415	-15.025	-7.1352	-11.
1	"point=2"	0.34227	-0.90966	-1.148	0.28691	1.2451	-0.53687	2.835	1.
1	"point=3"	0.7287	-0.77242	-0.98657	0.36508	1.9879	0.4163	3.452	2.0
1	"point=4"	0.87872	-0.76818	-0.96816	0.33477	2.3463	0.89785	3.737	2.4
1	"point=5"	0.91222	-0.83811	-1.028	0.24031	2.5277	1.1666	3.8662	2.7
1	"point=6"	0.85648	-0.97408	-1.1562	0.085331	2.5867	1.2933	3.8847	2.7
1	"point=7"	0.70558	-1.1905	-1.3663	-0.14666	2.5296	1.2915	3.7944	2.7
1	"point=8"	0.41577	-1.5358	-1.7061	-0.50502	2.319	1.1277	3.5559	2.5
1	"point=9"	-0.17716	-2.1767	-2.3424	-1.1573	1.7947	0.64395	3.0074	2.0
1	"point=10"	-9.4046	-11.446	-11.608	-10.437	-7.3721	-8.4872	-6.1808	-7.0
2	"point=1"	0.8347	-0.78233	-0.99938	0.30205	2.4239	1.0081	3.795	2.5
2	"point=2"	0.87277	-0.76475	-0.97586	0.34105	2.4846	1.0983	3.8381	2.6
2	"point=3"	0.89748	-0.75933	-0.96536	0.36018	2.5303	1.1709	3.8677	2.7
2	"point=4"	0.91044	-0.76388	-0.96549	0.36559	2.5622	1.2275	3.8849	2.7
2	"point=5"	0.91269	-0.7771	-0.97477	0.36049	2.581	1.269	3.89	2.
2	"point=6"	0.90477	-0.79823	-0.99237	0.34672	2.5867	1.296	3.883	2.7
2	"point=7"	0.88684	-0.8269	-1.0178	0.32535	2.5793	1.3084	3.8637	2.7
2	"point=8"	0.85872	-0.86304	-1.051	0.29697	2.5578	1.3055	3.831	2.7
2	"point=9"	0.81987	-0.90685	-1.0921	0.26182	2.5209	1.2862	3.7837	2.
2	"point=10"	0.76932	-0.95877	-1.1415	0.21989	2.4664	1.2483	3.7192	2.7
3	"point=1"	0.88992	-0.7791	-0.99528	0.33777	2.5443	1.234	3.8584	2.7
3	"point=2"	0.8979	-0.77144	-0.98526	0.3479	2.5562	1.251	3.8683	2.7
3	"point=3"	0.90413	-0.7658	-0.97741	0.35543	2.5661	1.2659	3.8762	2.
3	"point=4"	0.90869	-0.762	-0.97154	0.3607	2.5741	1.2785	3.8824	2.7
3	"point=5"	0.91163	-0.75989	-0.96746	0.36397	2.5801	1.289	3.8867	2.7
3	"point=6"	0.91299	-0.75934	-0.96506	0.36546	2.5842	1.2974	3.8892	2.7
3	"point=7"	0.91281	-0.76025	-0.96421	0.36532	2.5864	1.3035	3.89	2.7
3	"point=8"	0.91112	-0.76255	-0.96482	0.3637	2.5866	1.3074	3.889	2.7
3	"point=9"	0.90792	-0.76615	-0.96683	0.3607	2.5849	1.3091	3.8861	2.7
3	"point=10"	0.90324	-0.77102	-0.97016	0.35641	2.5811	1.3085	3.8815	2.7

Vectorized Exact Zooming Optimization, U(save)
pnts-per-iter=10, nbr-of-iters=3, zoom-ratio=0.25


```
Elapsed time is 0.997040 seconds.
-----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std	coefvari
	_	___	____	_____	____	____	_____	_____	_____	_____
ar_opti_foc_obj	1	1	2	8	1	8	10.125	1.2656	1.731	1.3677
ar_opti_save_frac	2	2	2	8	8	1	3.3843	0.42304	0.15074	0.35632
ar_opti_save_level	3	3	2	8	1	8	4.965	0.62062	0.71498	1.152

```
xxx TABLE:ar_opti_foc_obj xxxxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8
	_____	_____	_____	_____	_____	_____	_____	_____
r1	0.91163	-0.75989	-0.96506	0.36397	2.5864	1.3074	3.8892	2.7911

```
xxx TABLE:ar_opti_save_frac xxxxxxxxxxxxxxxxxxxx
```

	c1
	_____
r1	0.4084
r2	0.26826
r3	0.28289
r4	0.21333
r5	0.54941
r6	0.59246
r7	0.50637
r8	0.56319

```
xxx TABLE:ar_opti_save_level xxxxxxxxxxxxxxxxxxxx
```

	c1	c2	c3	c4	c5	c6	c7	c8
	_____	_____	_____	_____	_____	_____	_____	_____
r1	0.53854	-0.050976	0.053729	0.0013119	1.0103	0.45895	2.098	0.85508

Test FF_OPTIM_MZOOM_SAVEZRONE Speed

Test Speed doing 6.25 million state-spcae points for a savings problem:

```matlab
% Generate the state-space and function
rng(123);
it_draws = 6250000; % must be even number
ar_z1 = exp(rand([it_draws,1])*3-1.5);
ar_z2 = exp(rand([it_draws,1])*3-1.5);
ar_r = (rand(it_draws,1)*10.0);
ar_beta = [rand(round(it_draws/2),1)*0.9+0.1; rand(round(it_draws/2),1)*0.9+1];
% ffi_intertemporal_max is a function in ff_optim_mlsec_savezrone for testing
fc_util = @(x) ffi_intertemporal_util(x, ar_z1, ar_z2, ar_r, ar_beta);
% Call Function
bl_verbose = false;
bl_timer = true;
% set parameters
mp_mzoom_ctrlinfo = containers.Map('KeyType','char', 'ValueType','any');
mp_mzoom_ctrlinfo('it_mzoom_jnt_pnts') = 20;
mp_mzoom_ctrlinfo('it_mzoom_max_iter') = 10;
mp_mzoom_ctrlinfo('it_mzoom_zm_ratio') = 0.25;
[ar_opti_save_frac, ar_opti_save_level] = ...
    ff_optim_mzoom_savezrone(fc_util, bl_verbose, bl_timer, mp_mzoom_ctrlinfo);
```

```
Elapsed time is 64.837799 seconds.
```

```matlab
mp_container_map = containers.Map('KeyType','char', 'ValueType','any');
mp_container_map('ar_opti_save_frac') = ar_opti_save_frac;
mp_container_map('ar_opti_save_level') = ar_opti_save_level;
mp_container_map('ar_opti_save_frac_notnan') = ar_opti_save_frac(~isnan(ar_opti_save_frac));
ff_container_map_display(mp_container_map);
```

```
----------------------------------------
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CONTAINER NAME: mp_container_map ND Array (Matrix etc)
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

	i	idx	ndim	numel	rowN	colN	sum	mean	std
	_	___	____	_____	_____	____	_____	_____	_____
ar_opti_save_frac	1	1	2	6.25e+06	6.25e+06	1	2.8839e+06	0.46142	0.15305
ar_opti_save_frac_notnan	2	2	2	6.25e+06	6.25e+06	1	2.8839e+06	0.46142	0.15305
ar_opti_save_level	3	3	2	6.25e+06	6.25e+06	1	2.9481e+06	0.47169	0.66665

```matlab
figure();
histogram(ar_opti_save_frac(~isnan(ar_opti_save_frac)),100);
title('Distribution of Optimal Savings Fractions');
xlabel('Savings Fractions');
grid on;
```

Distribution of Optimal Savings Fractions
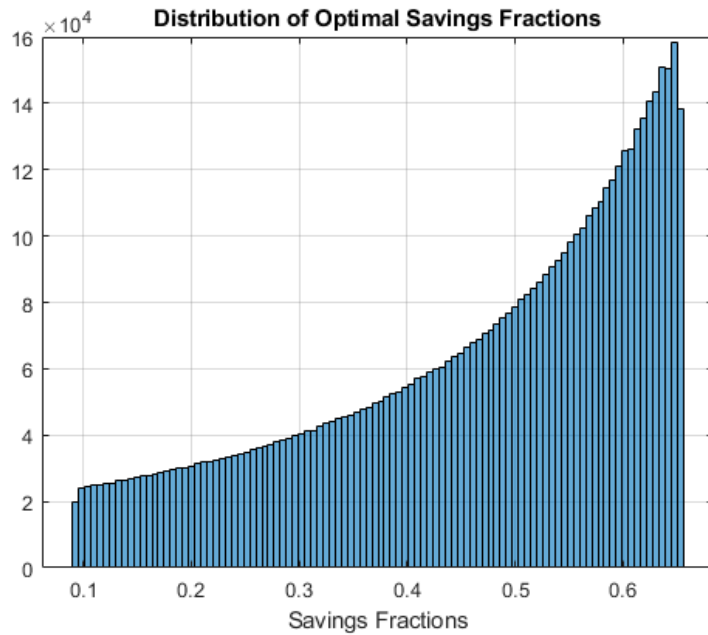
Define Two Period Intertemporal Log Utility No Shock Utility Function

See Household's Utility Maximization Problem and Two-Period Borrowing and Savings Problem given Endowments.

```
function [ar_util, ar_saveborr_level] = ...
    ffi_intertemporal_util(ar_saveborr_frac, z1, z2, r, beta)

ar_saveborr_level = ar_saveborr_frac.*(z1+z2./(1+r)) - z2./(1+r);
ar_util = log(z1 - ar_saveborr_level) + beta.*log(ar_saveborr_level.*(1+r) + z2);

end
```