Distribution with One Period Policy Shift

This is the example vignette for function: **snw_ds_main_vec** from the **PrjOptiSNW Package.**

One-period Deviation from Steady-State given Alternative Policy Function

In addition to solving for distribution given one policy function, **snw_ds_main_vec** can also solve for the distributional shift from "steady-state" with a one-period policy shift.

If a 6th parameter, PHI_ADJ_BASE, is provided to **snw_ds_main_vec,** solve for next-period forward distribution conditional on PHI_ADJ_BASE, using the policy function provided to **snw_ds_main_vec** as the 3rd and 4th parameters.

When PHI_ADJ_BASE is provided, if the AP_SS, CONS_SS policy functions inputs are from the same problem that generated PHI_ADJ_BASE, output PHI_ADJ will be identical to PHI_ADJ_BASE. However, if AP_SS, CONS_SS are different policy functions from those that induced PHI_ADJ_BASE, PHI_ADJ output will be different from PHI_ADJ_BASE input.

This allows for obtaining the distributional impact of a one period policy, allowing for deviation from "steady-state" distribution. This is used to solve for the distribution after one-period MIT shock, given stimulus checks provided in that period.

This is used to model the distributional effects of CARES Act, the two rounds of Trump Stimulus Checks, on household asset distribution when then receive the Biden stimulus checks from the the American Recovery Act. In effect, we have two MIT shock periods.

Solve for "Steady-State" Policy and Value Functions

Steady-state policy and value functions

```
% mp_params = snw_mp_param('default_dense');
mp_params = snw_mp_param('default_docdense');
% mp_params = snw_mp_param('default_moredense_a65zh133zs5_e2m2');
mp_controls = snw_mp_control('default_test');
mp_controls('bl_print_vfi') = false;
mp_controls('bl_print_vfi_verbose') = false;
mp_controls('bl_print_ds') = false;
mp_controls('bl_print_ds_verbose') = false;
[V_ss,ap_ss,cons_ss,mp_valpol_more_ss] = snw_vfi_main_bisec_vec(mp_params, mp_controls);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=550.0624
```

Solve for "Steady-State" Distribution

Solve for steady-state distributions, using steady-state policy functions.

```
[Phi_true_ss,Phi_adj_ss,A_agg_ss,Y_inc_agg_ss,~,mp_dsvfi_results_ss] = ...
    snw_ds_main_vec(mp_params, mp_controls, ap_ss, cons_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=547.3022
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1786.6361
```

```
% [Phi_true,Phi_adj] = snw_ds_main(mp_params, mp_controls);
Phi_true_ss = Phi_true_ss/sum(Phi_true_ss(:));
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_ss('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min	max	pYis0
a_ss	4.3602	2228	6.8796	1.5778	0.6755	0	135	0.118
ap_ss	4.4621	5.3216e+08	6.9169	1.5501	0.67638	0	163.73	0.0979
cons_ss	1.0635	5.0787e+07	0.6938	0.65237	0.33936	0.036717	141.61	
n_ss	2.3554	21	1.4375	0.61029	0.3128	1	6	
y_all	1.4661	8.3558e+07	1.4665	1.0003	0.44546	0.038108	50.873	
y_head_inc	1.1081	1.9253e+06	1.013	0.91419	0.42164	0.038108	24.357	
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0	18.957	0.20
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0	26.627	0.524
yshr_interest	0.12865	3.8438e+06	0.17577	1.3663	0.65781	0	0.99299	0.118
yshr_wage	0.77402	8.8881e+06	0.33679	0.43512	0.2062	0	1	0.105
yshr_SS	0.097329	29012	0.2266	2.3282	0.91382	0	1	0.79
yshr_tax	0.17833	2.8338e+06	0.035661	0.19998	0.11386	0.036506	0.2552	
yshr_nttxss	0.080996	2.8048e+06	0.24691	3.0485	1.2592	-0.89715	0.2552	

Solve for Policy Function Under Trump Stimulus

Same continuation value as prior (steady-state continuation), but now solve for new policy (one round) due to Trump stimulus. Same tax rate in covid and other years, manna-from-heaven. This calls the **snw_vfi_main_bisec_vec_stimulus** function, which provides the stimulus checks as a function of income and family status.

```
mp_params('a2_covidyr') = mp_params('a2_covidyr_manna_heaven');
[~,ap_trumpchecks,cons_trumpchecks, mp_valpol_more_trumpchecks] = ...
    snw_vfi_main_bisec_vec_stimulus(mp_params, mp_controls, V_ss);
```

```
Completed SNW_VFI_MAIN_BISEC_VEC 1 Period Unemp Shock;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time
```

Solve for Updated Distribution given Trump Stimulus

Fixing mass at their steady-state distribution, policy functions shift to the Trump stimulus policies, resolve for one-period forward distribution. The distributional code is almost identical, except uses steady-state distribution as the "base" distribution via parameter PHI_ADJ_SS.

```
[Phi_true_trumpchecks,Phi_adj_trumpchecks,...
    A_agg_trumpchecks,Y_inc_agg_trumpchecks,~,mp_dsvfi_results_trumpchecks] = snw_ds_main_vec(.
    mp_params, mp_controls, ...
    ap_trumpchecks, cons_trumpchecks, ...
    mp_valpol_more_trumpchecks, ...
    Phi_adj_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1546.0593
```

```
Phi_true_trumpchecks = Phi_true_trumpchecks/sum(Phi_true_trumpchecks(:));
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_trumpchecks('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min	max	pYis
a_ss	4.3988	2228	6.8731	1.5625	0.66733	0	135	0.034
ap_ss	4.5367	5.326e+08	6.9044	1.5219	0.66162	0	163.73	0.0090
cons_ss	1.0761	5.0871e+07	0.68968	0.64091	0.33326	0.048012	141.61	
n_ss	2.3554	21	1.4375	0.61029	0.3128	1	6	
y_all	1.4676	8.3558e+07	1.4664	0.99915	0.44498	0.038108	50.873	
y_head_inc	1.1097	1.9253e+06	1.0127	0.9126	0.42095	0.038108	24.357	
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0	18.957	0.2
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0	26.627	0.52
yshr_interest	0.13035	3.8438e+06	0.1754	1.3457	0.64346	0	0.99299	0.034
yshr_wage	0.77264	8.8881e+06	0.33616	0.43508	0.21167	0	1	0.10
yshr_SS	0.097017	29012	0.22582	2.3276	0.91391	0	1	0.7
yshr_tax	0.17842	2.8338e+06	0.035595	0.1995	0.11358	0.036506	0.2552	
yshr_nttxss	0.081403	2.8048e+06	0.24609	3.0231	1.249	-0.89715	0.2552	

Debug Check, SNW_DS_MAIN_VEC with Steady State Policies

This is to confirm that code is working properly. If we use steady-state policy functions and also provide as a sixth parameter the steady-state distribution, PHI_ADJ_SS, to **snw_ds_main_vec**, we should get back the same distribution, PHI_TRUE_SS_WITH_EXISTDIST_DEBUG, which is the same as PHI_ADJ_SS. See that the distributional outputs at the end of this subsection is the same as the distributional table before the table directly prior.

```
[Phi_true_ss_with_existdist_debug,~,~,~,~,mp_dsvfi_results_ss_with_existdist_debug] = snw_ds_ma
    mp_params, mp_controls, ...
    ap_ss, cons_ss, ...
    mp_valpol_more_ss, ...
    Phi_adj_ss);
```

```
Completed SNW_DS_MAIN_VEC;SNW_MP_PARAM=default_docdense;SNW_MP_CONTROL=default_test;time=1848.7637
```

```
Phi_true_ss_with_existdist_debug = Phi_true_ss_with_existdist_debug/sum(Phi_true_ss_with_existd
```

Show distributional results.

```
mp_cl_mt_xyz_of_s = mp_dsvfi_results_ss_with_existdist_debug('mp_cl_mt_xyz_of_s');
disp(mp_cl_mt_xyz_of_s('tb_outcomes'));
```

	mean	unweighted_sum	sd	coefofvar	gini	min	max	pYis0
a_ss	4.3602	2228	6.8796	1.5778	0.6755	0	135	0.118
ap_ss	4.4621	5.3216e+08	6.9169	1.5501	0.67638	0	163.73	0.0979
cons_ss	1.0635	5.0787e+07	0.6938	0.65237	0.33936	0.036717	141.61	
n_ss	2.3554	21	1.4375	0.61029	0.3128	1	6	
y_all	1.4661	8.3558e+07	1.4665	1.0003	0.44546	0.038108	50.873	
y_head_inc	1.1081	1.9253e+06	1.013	0.91419	0.42164	0.038108	24.357	
y_head_earn	0.88655	19732	0.92804	1.0468	0.53121	0	18.957	0.20
y_spouse_inc	0.35797	4.827e+05	0.95437	2.6661	0.85269	0	26.627	0.524
yshr_interest	0.12865	3.8438e+06	0.17577	1.3663	0.65781	0	0.99299	0.118

yshr_wage	0.77402	8.8881e+06	0.33679	0.43512	0.2062	0	1	0.105
yshr_SS	0.097329	29012	0.2266	2.3282	0.91382	0	1	0.79
yshr_tax	0.17833	2.8338e+06	0.035661	0.19998	0.11386	0.036506	0.2552	
yshr_nttxss	0.080996	2.8048e+06	0.24691	3.0485	1.2592	-0.89715	0.2552	