Calibrate Beta and Normalize GDP

Taking advantage of **snw_ds_main** from the **PrjOptiSNW Package**, this function calibrates the discount factor and also solves for the normalizing constant.

Calibrate Parameter Controls for SNW Functions

Set up controls for shock process and tiny/small/dense/densemore

```
clear all;
bl_print_mp_params = false;
% st_shock_method = 'rouwenhorst';
st_shock_method = 'tauchen';
% st_param_group = 'default_tiny';
% st_param_group = 'default_small';
% st_param_group = 'default_base';
% st_param_group = 'default_dense';
% st_param_group = 'default_moredense';
st_param_group = 'default_docdense';
mp_params = snw_mp_param(st_param_group, bl_print_mp_params, st_shock_method);
Pop = mp_params('Pop');
```

Set up print defaults

```
mp_controls = snw_mp_control('default_test');
mp_controls('bl_timer') = timer;
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;
```

Calibrate Routine

Test this for 3 iterations

```
%% Calibration
err=1;
tol=0.005;
```

```
Start calibration
```

```
it_counter = 1;
while err>tol && it_counter <= 3
    disp('');
    it=1;

    while it>0

        % Solve optimization problem and get the distribution
        tm_start_a2 = tic;
        a2_old = mp_params('a2');
        [Phi_true,~,A_agg,Y_inc_agg,it,mp_dsvfi_results, a2] = snw_ds_main(mp_params, mp_contro
        mp_params('a2') = a2;
        tm_end_a2 = toc(tm_start_a2);
        disp(['a2_old:' num2str(a2_old) ', a2_new:' num2str(a2) ', tm_end_a2:' num2str(tm_end_a
    end

    % Get Stats
    mp_cl_mt_xyz_of_s = mp_dsvfi_results('mp_cl_mt_xyz_of_s');
    tb_outcomes = mp_cl_mt_xyz_of_s('tb_outcomes');
    A_agg_alt = tb_outcomes{'a_ss', 'mean'}*sum(Pop);
    Aprime_agg_alt = tb_outcomes{'ap_ss', 'mean'}*sum(Pop);
    Y_inc_agg_alt = tb_outcomes{'y_all', 'mean'}*sum(Pop);
    Y_inc_median = tb_outcomes{'y_all', 'p50'};

    % Comparison
    name='Median household income (target=1.0)=';
    name2=[name,num2str(Y_inc_median)];
    disp(name2);
    name='Aggregate wealth to aggregate income (target=3.0)=';
    name2=[name,num2str(A_agg/Y_inc_agg)];
    disp(name2);

    err1=abs(Y_inc_median-1.0); % Target: Median household income (normalized to 1 in the model
    err2=abs((A_agg/Y_inc_agg)-3.0); % Target: Annual capital/income ratio of 3

    err=max(err1,err2);

    % Beta and Theta
    theta = mp_params('theta');
    beta = mp_params('beta');
    param_update=[theta;beta];

    if err>tol

        theta=theta*((1.0/Y_inc_median)^0.2); % Normalize theta such that median household inco
        beta=beta*((3.0/(A_agg/Y_inc_agg))^0.025); % Calibrate beta such that annual capital/ir

    end
    mp_params('theta') = theta;
    mp_params('beta') = beta;

    param_update=[param_update(1,1),theta;param_update(2,1),beta];

    it_counter = it_counter + 1;
    name='Old/updated theta:';
    st_theta=[name, num2str(param_update(1,:))];
    name='Old/updated beta:';
    st_beta=[name,num2str(param_update(2,:))];
    disp(['counter=' num2str(it_counter) ...
        ';beta=' num2str(beta) ...
        ';theta=' num2str(theta)]);
end
```

```
a2_old:1.5286, a2_new:1.5286, tm_end_a2:4134.1075
Median household income (target=1.0)=0.99853
Aggregate wealth to aggregate income (target=3.0)=3.0026
```

```
counter=2;beta=0.97116;theta=0.56523
```