Map Based Default Parameter Structure with varargin

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

Call Function with Default Parameters

Call function below without overriding

```
ff_defaultmap()
```

```
    'c_gap'    'c_max'    'c_min'    'c_min_for_util'    'fl_crra'    'it_rown'    'st_single_double'

    [1.0000e-03]    [60]    [1.0000e-03]    [1.0000e-03]    [1.5000]    [100]    'double'

  Elapsed time is 0.000896 seconds.
```

Call Function overriding some Parameters

```
param_map = containers.Map('KeyType','char', 'ValueType','any');
param_map('fl_w_max') = 1.11;
param_map('it_w_i') = 2.22;

support_map = containers.Map('KeyType','char', 'ValueType','any');
support_map('bl_display') = true;
ff_defaultmap(param_map, support_map)
```

```
    'c_gap'    'c_max'    'c_min'    'c_min_for_util'    'fl_crra'    'fl_w_max'    'it_rown'    'it_w_i'    'st_sin

    [1.0000e-03]    [60]    [1.0000e-03]    [1.0000e-03]    [1.5000]    [1.1100]    [100]    [2.2200]    'double'

  Elapsed time is 0.000667 seconds.
```

Function with Map Defaults and Overriding

This default parameter style is fairly succinct, allows for program testability, and easy adjustments/addition of additional parameters to models.

```
function ff_defaultmap(varargin)

% Parameters
params_len = length(varargin);
if params_len > 3
    error('ff_defaultmap:Can only have 3 container map parameters');
end
bl_input_override = 0;
if (params_len == 3)
    bl_input_override = varargin{3};
end

% Defaults
if (bl_input_override)
    % this relies on externally generated parameters, defaults do not have to be generated
    % if this file has to be invoked many times, then this saves time by avoiding
    % regenerating defaults over and over again
```

```
    [param_map, support_map, ~] = varargin{:};
else
    param_map = containers.Map('KeyType','char', 'ValueType','any');
    param_map('fl_crra') = 1.5;
    param_map('c_min') = 0.001;
    param_map('c_min_for_util') = 0.001;
    param_map('c_gap') = 10^-3;
    param_map('c_max') = 60;
    param_map('it_rown') = 100;
    param_map('st_single_double') = 'double';

    support_map = containers.Map('KeyType','char', 'ValueType','any');
    support_map('bl_display') = true;
    support_map('bl_graph') = true;
    support_map('bl_graph_onebyones') = true;
    support_map('bl_time') = true;
    support_map('bl_profile') = false;
    support_map('st_profile_path') = [pwd '/profile'];
    default_maps = {param_map, support_map};
end

% Parse Parameters
% see: C:\Users\fan\M4Econ\support\dtype\map_override.m
[default_maps{1:params_len}] = varargin{:};
param_map = [param_map; default_maps{1}];
support_map = [support_map; default_maps{2}];

params_group = values(param_map, {'fl_crra', 'c_min', 'c_min_for_util', 'c_gap', 'c_max'});
[fl_crra, c_min, c_min_for_util, c_gap, c_max] = params_group{:};
params_group = values(param_map, {'it_rown'});
[it_rown] = params_group{:};
params_group = values(param_map, {'st_single_double'});
[st_single_double] = params_group{:};

% support
params_group = values(support_map, {'bl_display', 'bl_graph', 'bl_graph_onebyones'});
[bl_display, bl_graph, bl_graph_onebyones] = params_group{:};
params_group = values(support_map, {'bl_time', 'bl_profile', 'st_profile_path'});
[bl_time, bl_profile, st_profile_path] = params_group{:};

% Tic toc starts
if (bl_time); tic; end

% Print Parameters
if (bl_display)
    disp(param_map.keys);
    disp(param_map.values);
end

% Profile On
if (bl_profile)
    close all;
    profile off;
    profile on;
```

```
end

%% Profiling
if (bl_profile)
    profile off
    profile viewer
    profsave(profile('info'), st_profile_path);
end

if (bl_time); toc; end

end
```