Forecasts with Judgmental Adjustments

forecasts_with_judgment.m

by Jaromir Benes

May 27, 2015

Summary

Use the Kalman filtered data as the starting point for forecasts, both unconditional and conditional, i.e. with various types of judgmental adjustments.

Contents

1 Clear Workspace

Clear workspace, close all graphics figures, clear command window, and check the IRIS version.

```
clear;
close all;
clc;
irisrequired 20140315;
```

2 Load Estimated Model Object, Filtered Data, and Historical Database

Load the model object estimated in `estimate_params`, the filtered (smoothed) data from a Kalman filter in `filter_hist_data`, and the historical database created in `read_data`. Run `estimate_params` and `filter_hist_data` at least once before running this m-file.

```
load estimate_params.mat mest;
load filter_hist_data.mat f;
load read_data.mat d startHist endHist;
```

3 Define Dates

```
startFcst = endHist + 1;
endFcst = startFcst + 3*4;
startPlot = startFcst - 12;
plotRng = startPlot:endFcst;
highRng = startPlot:endHist;
```

4 Define Graphics Styles

The structs `sty1` and `sty2` are used in the option `'style='` in `qplot` to automatically style the graphs plotted.

```
sty1 = struct();
sty1.line.color = {'blue','blue','blue'};
sty1.line.lineStyle = {'-','--','--'};
sty1.line.lineWidth = 1.5;
sty1.line.marker = {'.','none','none'};
sty1.axes.fontSize = 7;
```

```
sty1.legend.fontSize = 7;

sty2 = sty1;
sty2.line.color = {'blue','red','blue','red','blue','red'};
sty2.line.lineStyle = {'-','-','--','--','--','--'};
sty2.line.lineWidth = 1.5;
sty2.line.marker = {'.','.','none','none','none','none'};
sty2.axes.fontSize = 7;
sty2.legend.fontSize = 7;
```

5 Run Unconditional Forecast

Unconditional forecast runs from the initial condition supplied in the input database, `f`. The initial conditions consist of the mean and the root mean square error (initial uncertainty) for each variable. Directly observed variables have obviously RMSE zero, the unobservables (such as productivity) have non-zero initial uncertainty.

```
u = jforecast(mest,f,plotRng);

u %#ok<NOPTS>
u.mean

u.mean = dboverlay(f.mean,u.mean);
u.std = dboverlay(f.std,u.std);
```

```
u =
    mean: [1x1 struct]
     std: [1x1 struct]
ans =
      Short: [25x1 tseries]
       Infl: [25x1 tseries]
     Growth: [25x1 tseries]
       Wage: [25x1 tseries]
          Y: [27x1 tseries]
          N: [25x1 tseries]
          W: [27x1 tseries]
          Q: [25x1 tseries]
          H: [25x1 tseries]
          A: [27x1 tseries]
          P: [29x1 tseries]
          R: [26x1 tseries]
         Pk: [25x1 tseries]
         Rk: [25x1 tseries]
```

```
    Lambda: [25x1 tseries]
        dP: [26x1 tseries]
       d4P: [25x1 tseries]
        dW: [26x1 tseries]
       RMC: [25x1 tseries]
        Mp: [25x1 tseries]
        Mw: [25x1 tseries]
        Ey: [25x1 tseries]
        Ep: [25x1 tseries]
        Ea: [25x1 tseries]
        Er: [25x1 tseries]
        Ew: [25x1 tseries]
     alpha: 1.0074
      beta: 0.9962
     gamma: 0.6000
     delta: 0.0300
         k: 10
        pi: 1.0062
       eta: 6
       psi: 0.2500
       chi: 0.9138
       xiw: 133.8447
       xip: 264.6905
      rhoa: 0.9000
      rhor: 0.8587
    kappap: 2.9459
    kappan: 0.3419
    Short_: -3.9012
     Infl_: -0.3539
   Growth_: 0.0078
     Wage_: -1.9244
```

6 Create Plot Lists

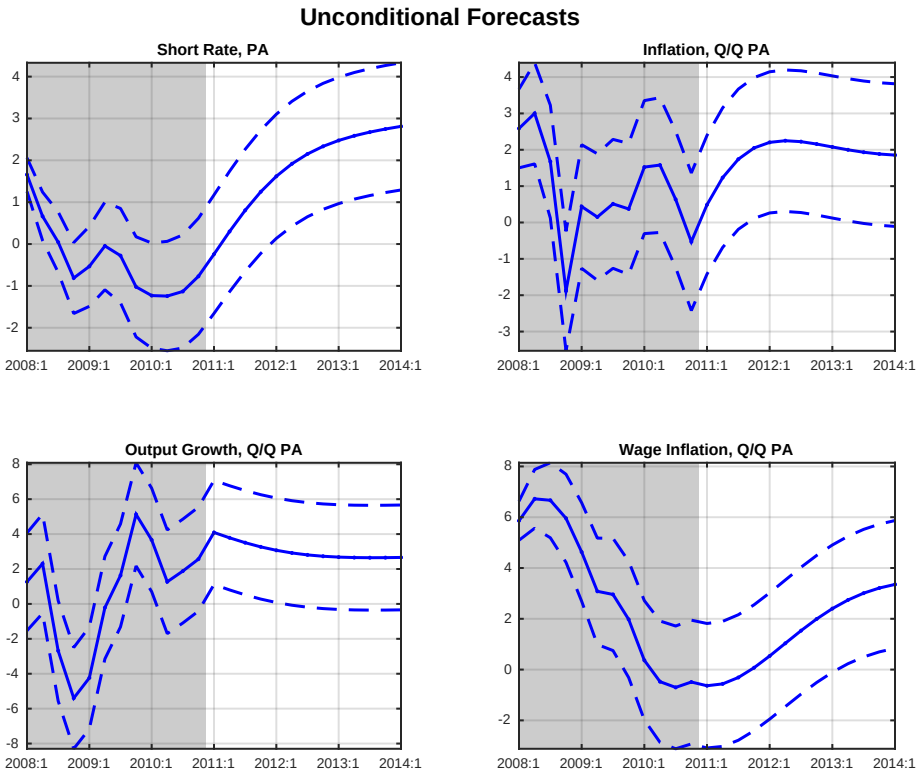
Define variables and titles to appear in graphs created by `dbplot` functions after each forecast experiment.

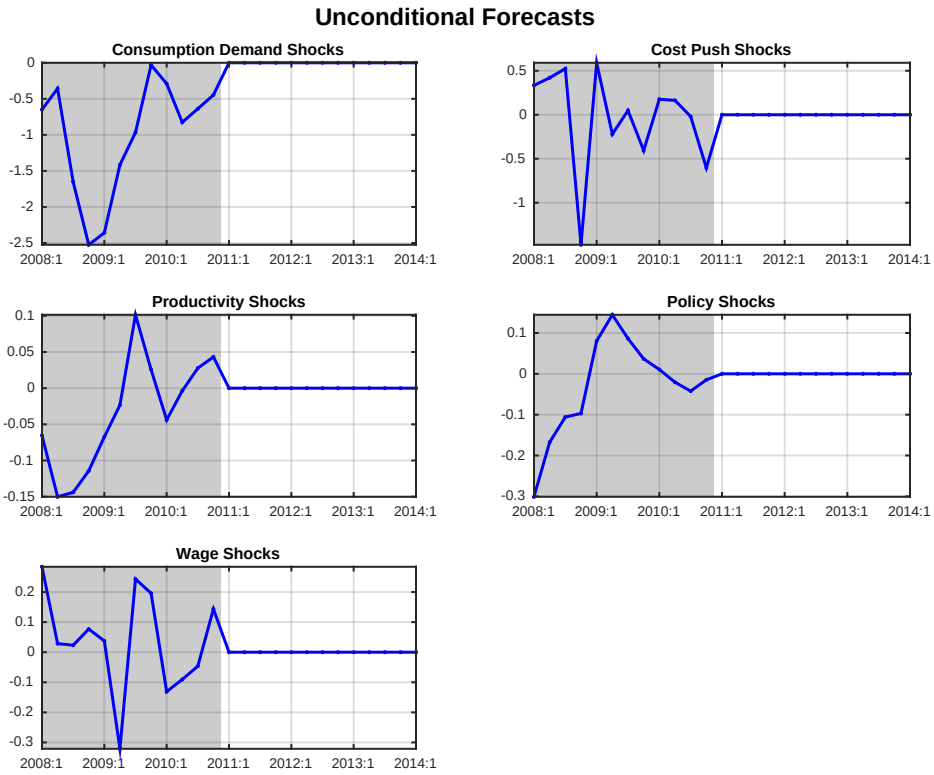
```
plotList1 = { ...
    ' "Short Rate, PA" [mean.Short, mean.Short+std.Short, mean.Short-std.Short]', ...
    ' "Inflation, Q/Q PA", [mean.Infl, mean.Infl+std.Infl, mean.Infl-std.Infl]', ...
    ' "Output Growth, Q/Q PA" [mean.Growth, mean.Growth+std.Growth, mean.Growth-std.Growth]', ...
    ' "Wage Inflation, Q/Q PA" [mean.Wage, mean.Wage+std.Wage, mean.Wage-std.Wage]', ...
    };
```

```
85
86  plotList2 = { ...
87          ' "Consumption Demand Shocks" mean.Ey', ...
88      ' "Cost Push Shocks" mean.Ep', ...
89      ' "Productivity Shocks" mean.Ea', ...
90      ' "Policy Shocks" mean.Er', ...
91      ' "Wage Shocks" mean.Ew', ...
92      };
```

7 Report Unconditional Forecast

```
96  dbplot(u,startPlot:endFcst,plotList1, ...
97      'tight=',true,'style=',sty1,'highlight=',highRng);
98  grfun.ftitle('Unconditional Forecasts');
99  grfun.bottomlegend('Mean','Mean +/- 1 Std');
100
101 dbplot(u,startPlot:endFcst,plotList2, ...
102     'tight=',true,'style=',sty1,'highlight=',highRng, ...
103     'transform=',@(x) 100*x);
104 grfun.ftitle('Unconditional Forecasts');
```


Unconditional Forecasts
Short Rate, PA
Inflation, Q/Q PA
Output Growth, Q/Q PA
Wage Inflation, Q/Q PA
2008:1
2009:1
2010:1
2011:1
2012:1
2013:1
2014:1



8 Exogenise Interest Rates

In this judgmentally adjusted forecast, swap the endogeneity and exogeneity of the short rates and the policy shocks. In other words, the short rates are kept fixed at a specified level (here, it is the last observed value), and the policy shocks become a new "endogenous variable" that adjust exactly so to make the policy rule consistent with the fixed interest rates.

The forecast with exogenised interest rates is run in an anticipated mode.

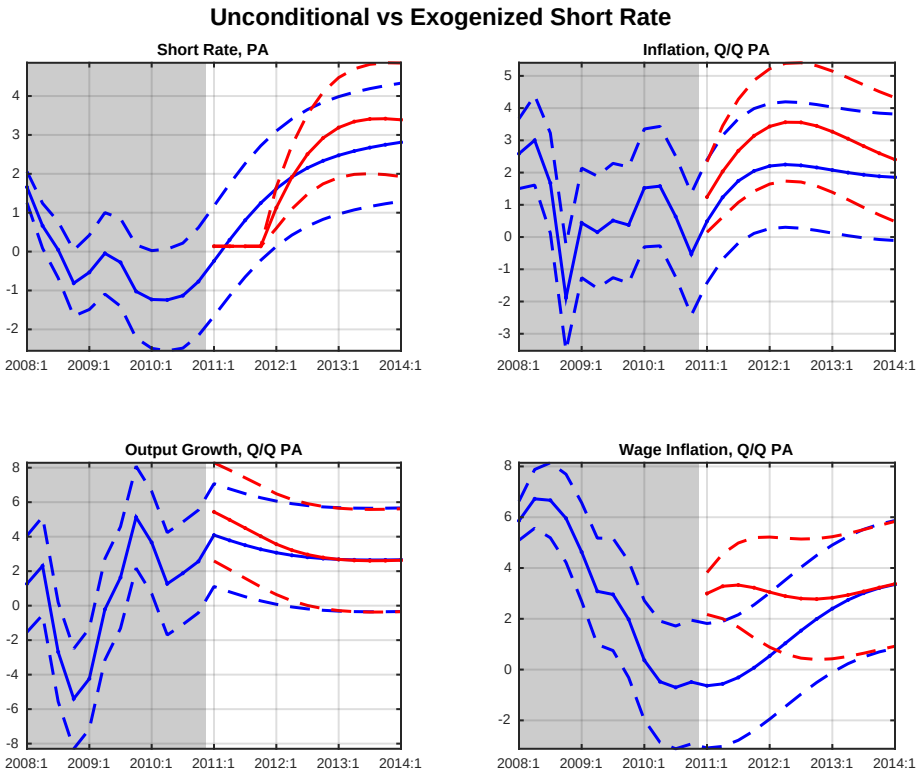
```
118 p1 = plan(mest,startFcst:endFcst);
119 p1 = exogenize(p1,'Short',startFcst:startFcst+3);
120 p1 = endogenize(p1,'Er',startFcst:startFcst+3);
121
122 f1 = f;
123 f1.mean.Short(startFcst:startFcst+3,1) = f.mean.Short(endHist);
124
125 detail(p1,f1);
126
```

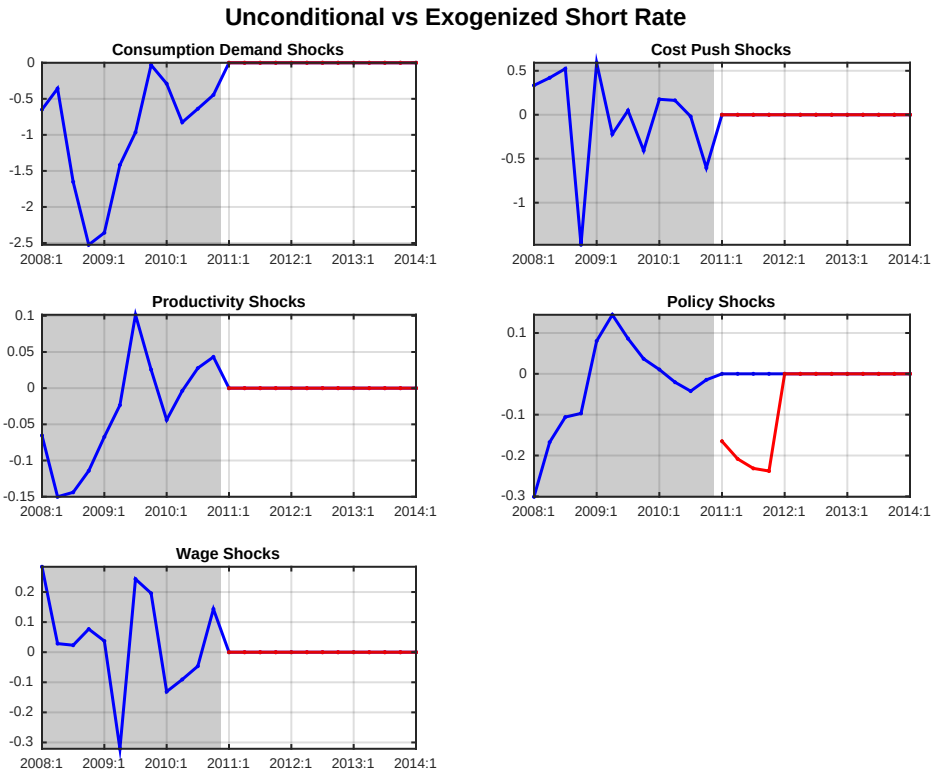
```
127 j1 = jforecast(mest,f1,startFcst:endFcst,'plan=',p1);
```

```
        Exogenized: [4]
                Short  *2011Q1[=0.136667] *2011Q2[=0.136667] *2011Q3[=0.136667] *2011Q4[=0.136667]
        Endogenized real: [4]
                Er     *2011Q1[@1] *2011Q2[@1] *2011Q3[@1] *2011Q4[@1]
        Endogenized imag: [0]
        -
        Conditioned upon: [0]
        -
```

9 Compare Exogenised Forecasts with Unconditional Forecasts

```
131 dbplot(u & j1,startPlot:endFcst,plotList1, ...
132     'tight=',true,'style=',sty2,'highlight=',highRng);
133 grfun.ftitle('Unconditional vs Exogenized Short Rate');
134 grfun.bottomlegend('Uncond Mean','Exogen Mean', ...
135     'Uncond Mean +/- 1 Std','Exogen Mean +/- 1 Std');
136
137 dbplot(u & j1,startPlot:endFcst,plotList2, ...
138     'tight=',true,'style=',sty2,'highlight=',highRng, ...
139     'transform=',@(x) 100*x);
140 grfun.ftitle('Unconditional vs Exogenized Short Rate');
```


Unconditional vs Exogenized Short Rate
Short Rate, PA
Inflation, Q/Q PA
Output Growth, Q/Q PA
Wage Inflation, Q/Q PA
2008:1
2009:1
2010:1
2011:1
2012:1
2013:1
2014:1



10 Condition on Anticipated Interest Rates

In this exercise, keep the interest rates fixed, but use a very different mechanism to do that. Compute the most likely combination of all possible shocks, except the monetary policy shocks, and changes in the initial conditions to reproduce a given path for the interest rates (it is again a flat track). The forecast is produced in an anticipated mode, which means that all agents know the future shocks from the very beginning.

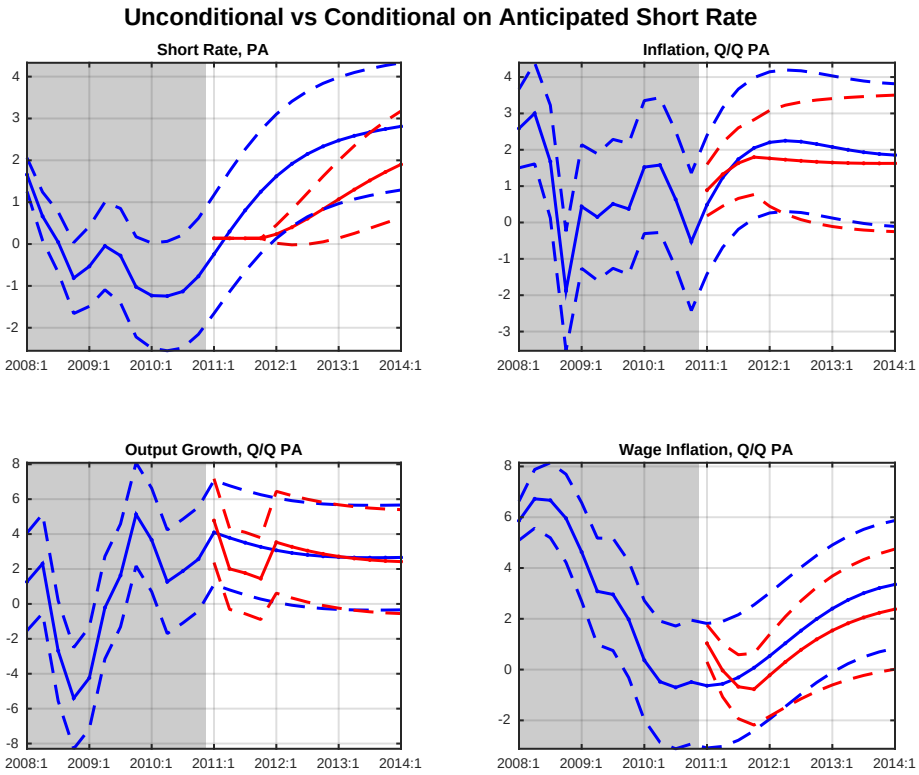
```
151 mest1 = mest;
152 mest1.std_Er = 0;
153
154 get(mest,'std') & get(mest1,'std') %#ok<NOPTS>
155
156 p2 = plan(mest1,startFcst:endFcst);
157 p2 = condition(p2,'Short',startFcst:startFcst+3);
158
159 f2 = f;
160 f2.mean.Short(startFcst:startFcst+3) = f2.mean.Short(endHist);
```

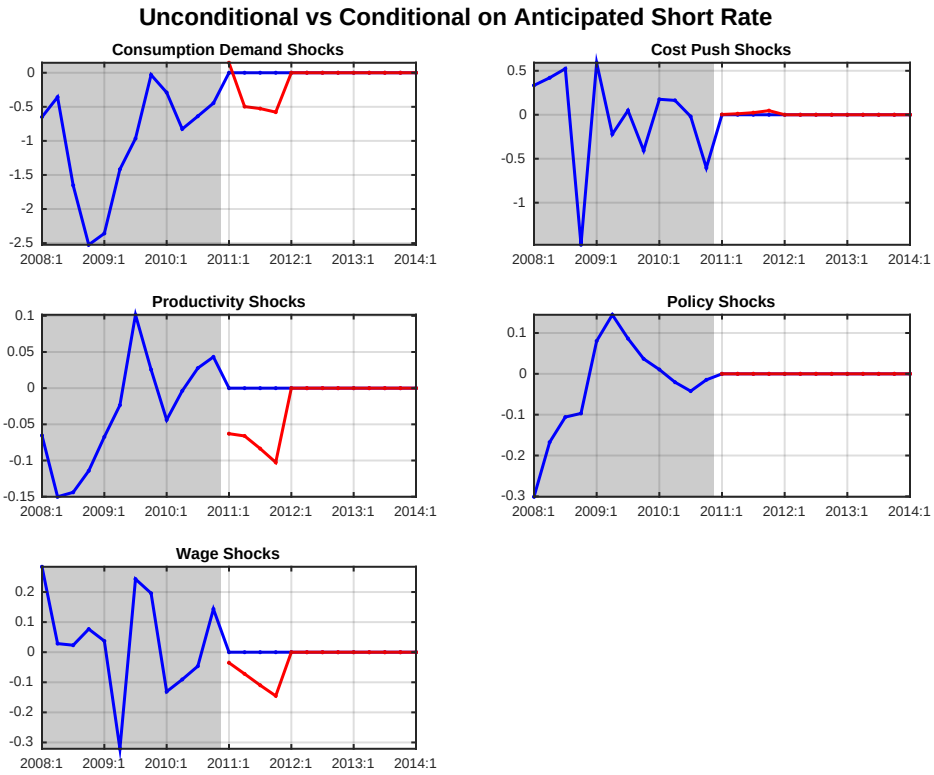
```
161
162 c = struct();
163 c.Short = f2.mean.Short;
164
165 detail(p2,f2);
166
167 j2 = jforecast(mest1,f2,startFcst:endFcst,c,'plan=',p2);
```

```
ans =
    std_Mp: [0 0]
    std_Mw: [0 0]
    std_Ey: [0.0079 0.0079]
    std_Ep: [0.0032 0.0032]
    std_Ea: [0.0011 0.0011]
    std_Er: [9.2918e-04 0]
    std_Ew: [0.0019 0.0019]

        Exogenized: [0]
        -
        Endogenized real: [0]
        -
        Endogenized imag: [0]
        -
        Conditioned upon: [4]
                Short  *2011Q1[=0.136667] *2011Q2[=0.136667] *2011Q3[=0.136667] *2011Q4[=0.136667]
```

11 Compare Anticipated Conditional Forecasts with Unconditional Forecasts

```
171 dbplot(u & j2,startPlot:endFcst,plotList1, ...
172     'tight=',true,'style=',sty2,'highlight=',highRng);
173 grfun.ftitle('Unconditional vs Conditional on Anticipated Short Rate');
174 grfun.bottomlegend('Uncond Mean','Cond Mean', ...
175     'Uncond Mean +/- 1 Std','Cond Mean +/ 1 Std');
176
177 dbplot(u & j2,startPlot:endFcst,plotList2, ...
178     'tight=',true,'style=',sty2,'highlight=',highRng, ...
179     'transform=',@(x) 100*x);
180 grfun.ftitle('Unconditional vs Conditional on Anticipated Short Rate');
```


Unconditional vs Conditional on Anticipated Short Rate
Short Rate, PA
Inflation, Q/Q PA
Output Growth, Q/Q PA
Wage Inflation, Q/Q PA
2008:1
2009:1
2010:1
2011:1
2012:1
2013:1
2014:1



12 Condition on Unanticipated Interest Rates

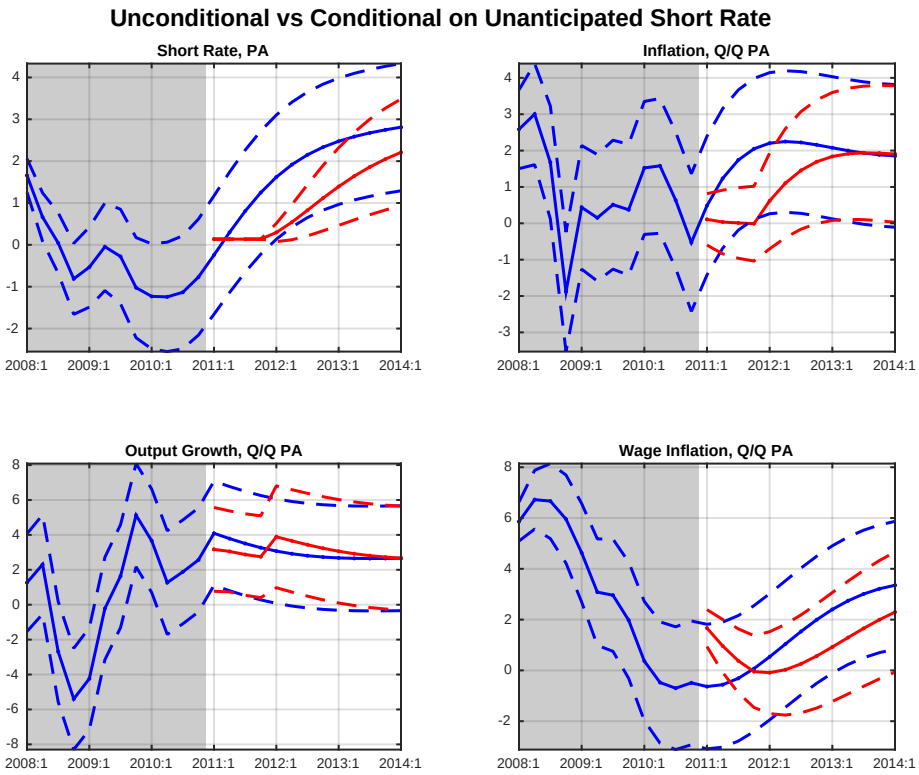
Do the same as above, but with the conditioning interest rate unanticipated.

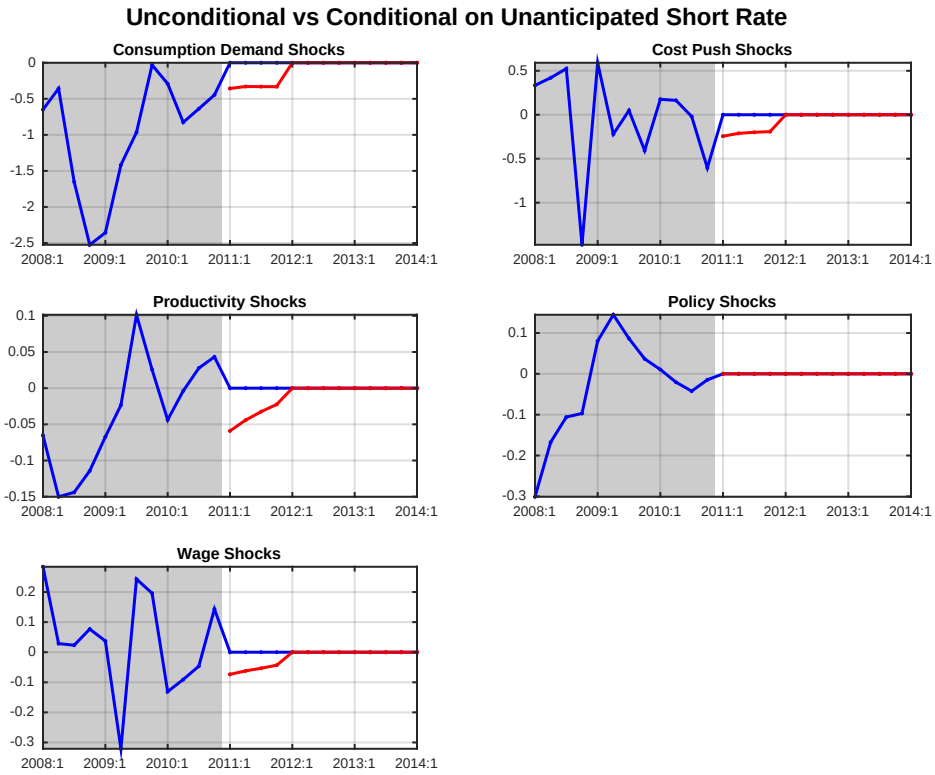
```
p3 = p2;
f3 = f2;

j3 = jforecast(mest1,f3,startFcst:endFcst+50, ...
    'plan=',p3,'anticipate=',false);
```

13 Compare Unanticipated Conditional Forecasts with Uncondtional Forecasts

```
dbplot(u & j3,startPlot:endFcst,plotList1, ...
    'tight=',true,'style=',sty2,'highlight=',highRng);
grfun.ftitle('Unconditional vs Conditional on Unanticipated Short Rate');
grfun.bottomlegend('Uncond Mean','Cond Mean', ...
```

```
    'Uncond Mean +/- 1 Std','Cond Mean +/ 1 Std');

dbplot(u & j3,startPlot:endFcst,plotList2, ...
    'tight=',true,'style=',sty2,'highlight=',highRng, ...
    'transform=',@(x) 100*x);
grfun.ftitle('Unconditional vs Conditional on Unanticipated Short Rate');
```





14 Exogenised Interest Rates and Condition on Inflation

Combine two techniques together: exogenizing and conditioning.

```
p4 = plan(mest,startFcst:endFcst);

p4 = exogenise(p4,'Short',startFcst:startFcst+3);
p4 = endogenise(p4,'Er',startFcst:startFcst+3);

p4 = condition(p4,'Infl',startFcst:startFcst+3);

f4 = f;
f4.mean.Short(startFcst:startFcst+3) = f4.mean.Short(endHist);
f4.mean.Infl(startFcst:startFcst+3) = f4.mean.Infl(endHist);

j4 = jforecast(mest1,f4,startFcst:endFcst+50,'plan=',p4);
```

15 Verify Exogenised and Conditioned Data Points

Print the forecasts for the interest rate and inflation, and compare the forecasts with the values we supplied in the input database.

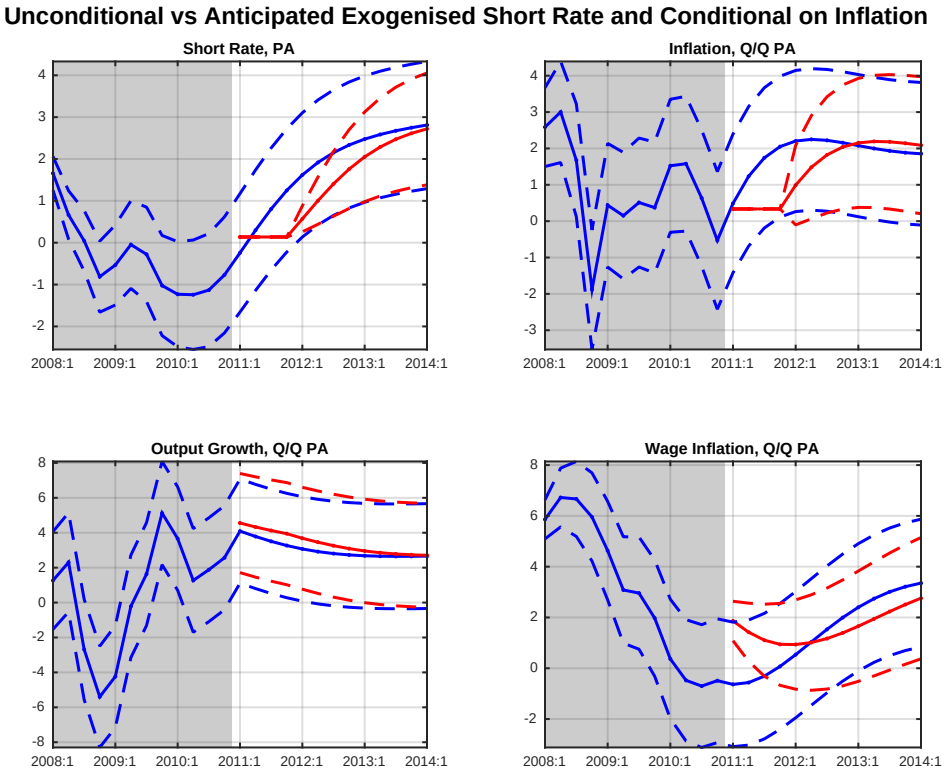
```
disp('Interest rate forecast and tunes');
[j4.mean.Short{startFcst:startFcst+3}, ...
    f4.mean.Short{startFcst:startFcst+3}] %#ok<NOPTS>

disp('Inflation forecast and conditions');
[j4.mean.Infl{startFcst:startFcst+3}, ...
    f4.mean.Infl{startFcst:startFcst+3}] %#ok<NOPTS>
```

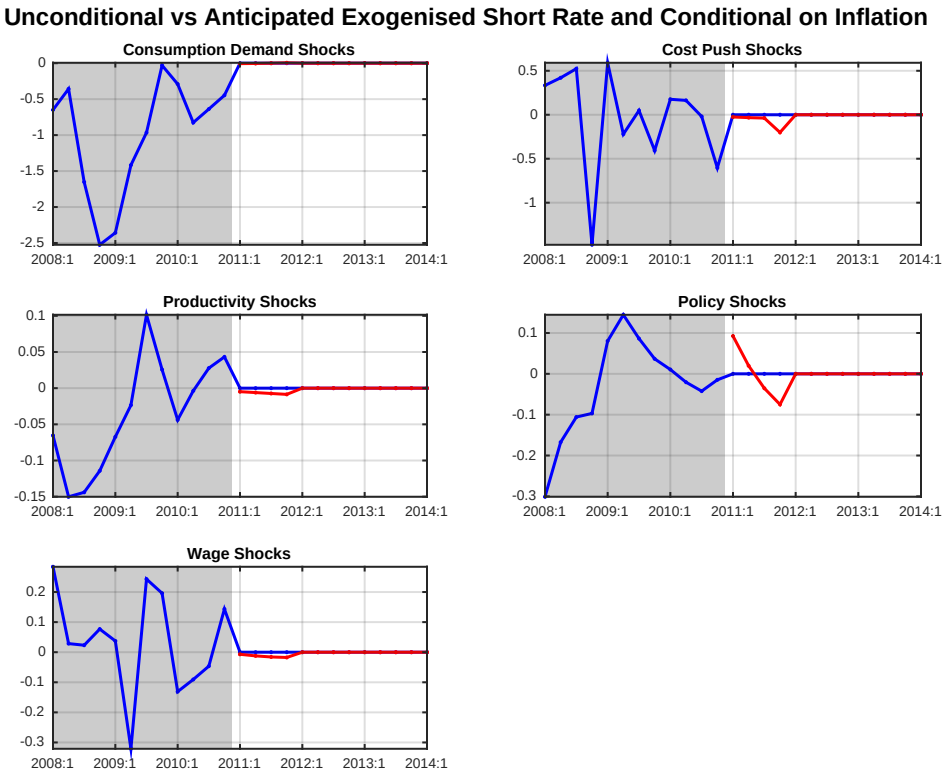
```
Interest rate forecast and tunes
ans =
        tseries object: 4-by-2
        2011Q1:  0.13667     0.13667
        2011Q2:  0.13667     0.13667
        2011Q3:  0.13667     0.13667
        2011Q4:  0.13667     0.13667
    'Short Term Rate'    'Short Term Rate'
        user data: empty
        export files: [0]
Inflation forecast and conditions
ans =
        tseries object: 4-by-2
        2011Q1:  0.33538     0.33538
        2011Q2:  0.33538     0.33538
        2011Q3:  0.33538     0.33538
        2011Q4:  0.33538     0.33538
    'Price Inflation'    'Price Inflation'
        user data: empty
        export files: [0]
```

16 Compare Exogenised/Conditional Forecasts with Unconditional Forecasts

```
dbplot(u & j4,startPlot:endFcst,plotList1, ...
    'tight=',true,'style=',sty2,'highlight=',highRng);
grfun.ftitle(['Unconditional vs ', ...
    'Anticipated Exogenised Short Rate and Conditional on Inflation']);
grfun.bottomlegend('Uncond Mean','Cond Mean', ...
    'Uncond Mean +/- 1 Std','Cond Mean +/ 1 Std');

```

```
246 dbplot(u & j4,startPlot:endFcst,plotList2, ...
247     'tight=',true,'style=',sty2,'highlight=',highRng, ...
248     'transform=',@(x) 100*x);
249 grfun.ftitle(['Unconditional vs ', ...
250     'Anticipated Exogenised Short Rate and Conditional on Inflation']);
```





17 Resimulate Point Forecasts

The function `simulate` only uses the input database for initial condition and in-sample shocks. The shocks backed out by `jforecast` are such that they exactly reproduce the exogenised and/or conditioned data points.

1 Use the function `maxabs` to report the max abs differences between the fields of the same name in two structs (databases).

```
s1 = simulate(mest1,j1.mean,startFcst:endFcst);
s2 = simulate(mest1,j2.mean,startFcst:endFcst);
s3 = simulate(mest1,j3.mean,startFcst:endFcst,'anticipate=',false);
s4 = simulate(mest1,j4.mean,startFcst:endFcst);

maxabs(s1,j1.mean) ... 1
    & maxabs(s2,j2.mean) ...
    & maxabs(s3,j3.mean) ...
```

```
270     & maxabs(s4,j4.mean) %#ok<NOPTS>
```

```
ans =
      Short: [4.6185e-14 1.0392e-13 7.1942e-14 8.5265e-14]
       Infl: [3.8192e-14 8.9706e-14 9.2371e-14 8.9262e-14]
     Growth: [3.1974e-14 7.5495e-14 4.8406e-14 8.1712e-14]
       Wage: [1.4477e-13 1.4033e-13 8.0824e-14 2.0428e-13]
          Y: [2.2204e-15 2.8866e-15 2.6645e-15 2.8866e-15]
          N: [2.2204e-16 4.4409e-16 3.3307e-16 4.4409e-16]
          W: [6.2172e-15 1.0658e-14 4.8850e-15 1.0658e-14]
          Q: [6.6613e-16 1.7764e-15 8.8818e-16 1.5543e-15]
          H: [2.2204e-15 3.1086e-15 2.6645e-15 2.6645e-15]
          A: [1.5543e-15 2.4425e-15 1.9984e-15 2.6645e-15]
          P: [8.8818e-16 2.2204e-15 4.4409e-16 1.7764e-15]
          R: [2.2204e-16 2.2204e-16 2.2204e-16 2.2204e-16]
         Pk: [4.4409e-15 7.9936e-15 5.7732e-15 8.8818e-15]
         Rk: [1.6653e-16 2.3592e-16 1.5266e-16 2.4980e-16]
     Lambda: [6.1062e-16 1.3878e-15 9.4369e-16 1.3323e-15]
         dP: [2.2204e-16 2.2204e-16 2.2204e-16 2.2204e-16]
        d4P: [2.2204e-16 6.6613e-16 6.6613e-16 8.8818e-16]
         dW: [2.2204e-16 4.4409e-16 2.2204e-16 4.4409e-16]
        RMC: [3.3307e-16 3.3307e-16 4.4409e-16 6.6613e-16]
         Mp: [0 0 0 0]
         Mw: [0 0 0 0]
         Ey: [0 0 0 0]
         Ep: [0 0 0 0]
         Ea: [0 0 0 0]
         Er: [0 0 0 0]
         Ew: [0 0 0 0]
      alpha: [0 0 0 0]
       beta: [0 0 0 0]
      gamma: [0 0 0 0]
      delta: [0 0 0 0]
          k: [0 0 0 0]
         pi: [0 0 0 0]
        eta: [0 0 0 0]
        psi: [0 0 0 0]
        chi: [0 0 0 0]
        xiw: [0 0 0 0]
        xip: [0 0 0 0]
       rhoa: [0 0 0 0]
       rhor: [0 0 0 0]
     kappap: [0 0 0 0]
     kappan: [0 0 0 0]
     Short_: [0 0 0 0]
      Infl_: [0 0 0 0]
```

```
    Growth_: [0 0 0 0]
      Wage_: [0 0 0 0]
```

18 Help on IRIS Functions Used in This Files

Use either `help` to display help in the command window, or `idoc` to display help in an HTML browser window.

```
help data/dbextend
help model/jforecast
help model/subsasgn
help qreport/qplot
help grfun/ftitle
help maxabs
```