Using Iris to compute data moments

[dataMoments.m]

by

Antti Ripatti with help of Jaromir Benes

25 May 2011

Introduction

This document is part of the exercise classes of my Structural Macroeconometrics course at the Finnish Doctoral Program of Economics.

The course web site is http://teaching.ripatti.net/meconom/

In this file, we show how to compute the basic data moments that we explored in the course (see the slides)

Contents

1 Start up IRIS

Run the following two commands to start up an IRIS session if you haven't done so yet:

```
addpath c:\Matlab_Tbx\iris8; % or any other directory of Iris
irisstartup;
```

2 Clear workspace

```
home();
clear();
close('all');
```

3 Define global control variables

1 number of simulations in bootstrapping; try 500 when debugging 2 order of the VAR model. In computing auto- and crosscorrelations we use $2\times$ `varorder`.

```
myRange = qq(1976,1):qq(2009,4);
nDraws = 10000; 1
varorder = 4; 2
```

4 Load data

I assume that you are familiar with `tseries` object and `dbase` structure and you have constructed your data and stored it to Matlab binary file.

A pedagogic way to learn those animals is to use the Iris tutorials.

See: http://code.google.com/p/iris-toolbox-project/wiki/Petars

```
db = loadstruct('data.mat');
```

5 Simple time series plots

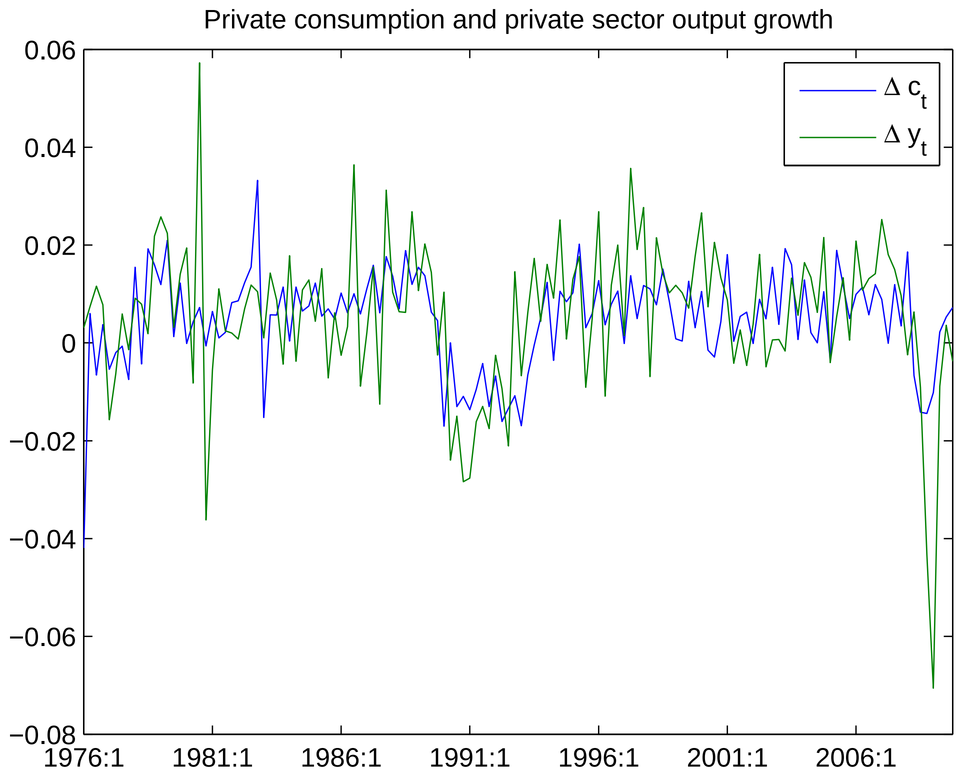
Always, and everywhere, plot your time series!!!

Eyeball econometrics of the graph below shows, among other things, the following

- Consumption is less volatile than output/income
- The trends of consumption and income correlate contemporaneously

Both the above observations can be explained by the permanent income hypotheses. Similar results can be obtained studying \emphdata moments.

```
figure();
plot(myRange,[db.dlCH_obs db.dlY_obs]);
legend('\Delta c_t','\Delta y_t');
title('Private consumption and private sector output growth');
```



6 Estimate VAR on dlCH and dlY

We estimate a VAR object on two series, consumption growth `dlCH` and output growth `dlY`, and then use the estimated VAR to calculate the implied second-moment characteristics in both the frequency and the time domains.

We use the following options in the `estimate` function:f

- `'order'` to specify the VAR order, i.e. the number of lags included,
- `'covParameters'` to tell IRIS to compute also the approximate covariance matrix of the parameter estimates.
- etc...

```
v = VAR();
[v,data] = estimate(v,db,{'dlCH_obs','dlY_obs'},myRange, ...
    'order=',varorder,'covParameters=',true);
```

7 Spectral analysis dlCH and dlY

We use the xsf function to compute the power spectrum and spectral density based on the estimated VAR model. The power spectrum is defined as follows: In addition to the spectra of single time series, the relationships between pairs of variables can be examined in the frequency domain.

The spectrum can be generalized for the vector case, and the cross spectrum between y_t and x_t \is defined analogously with the spectrum of a single series,

$$s_{yx}\left(\omega\right)=\frac{1}{2\pi}\sum_{\tau=-\infty}^{\infty}\gamma_{yx}\left(\tau\right)e^{-i\omega\tau}.$$

Instead of the cross spectrum, functions derived from it (phase and coherence) allow for convenient interpretation of the relationship between two variables in the frequency domain.

Given data on y and x , the cross spectrum and the derived functions can be computed analogously to the power spectrum. In particular, some smoothing is required to obtain consistent estimators of the phase and coherence.

The cross-spectral density function can be expressed in terms of its real component $c_{yx}(\omega)$ (the cospectrum) and imaginary component q_{yx} (the quadrature spectrum)

$$s_{yx}(\omega)=c_{yx}(\omega)+i\,q_{yx}(\omega).$$

In polar form, the cross-spectral density can be written

$$s_{yx}(\omega)=R(\omega)\,e^{i\theta(\omega)},$$

where

$$R(\omega)=\sqrt{c_{yx}(\omega)^2+q_{yx}(\omega)^2}$$

and

$$\theta(\omega)=\arctan\frac{-q_{yx}(\omega)}{c_{yx}(\omega)}.$$

3 We call xsf to compute the power spectrum, S, and spectral density, D. Both matrices are N-by-N-by-K, where N is the number of variables in the VAR, and K is the number of frequences at which the spectra are evaluated.

4 The xsf2coher computes coherence based on a power spectrum. Th resulting matrix, C, is again N-by-N-by-K.

The (squared) coherence is defined analogously to correlation:

$$C(\omega) = \frac{|s_{yx}(\omega)|^2}{s_x(\omega)s_y(\omega)},$$

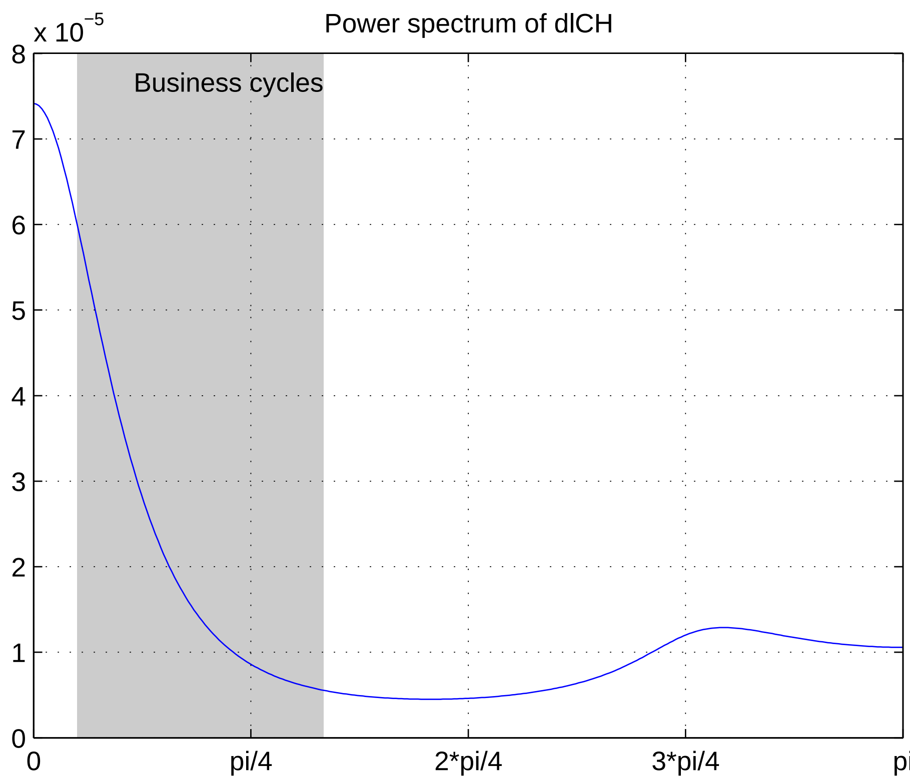
and it can be interpreted as a measure of the correlation between the series y and x at different frequencies.

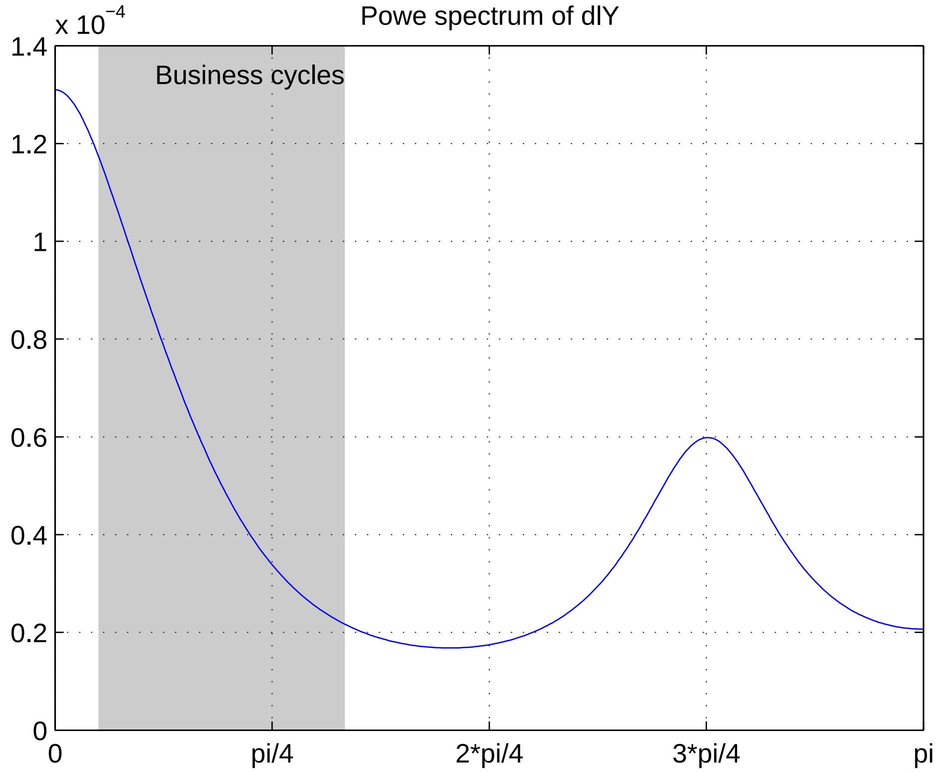
5 The `xsf2gain` function computes the gain based on a power spectrum. The resultng matrix, `G`, is the same size as `C` in 4. The function $R(\omega)$ is the gain function.

- Note that we highlight business cycle frequencies (between 6 and 40 quarters) in the graphs.

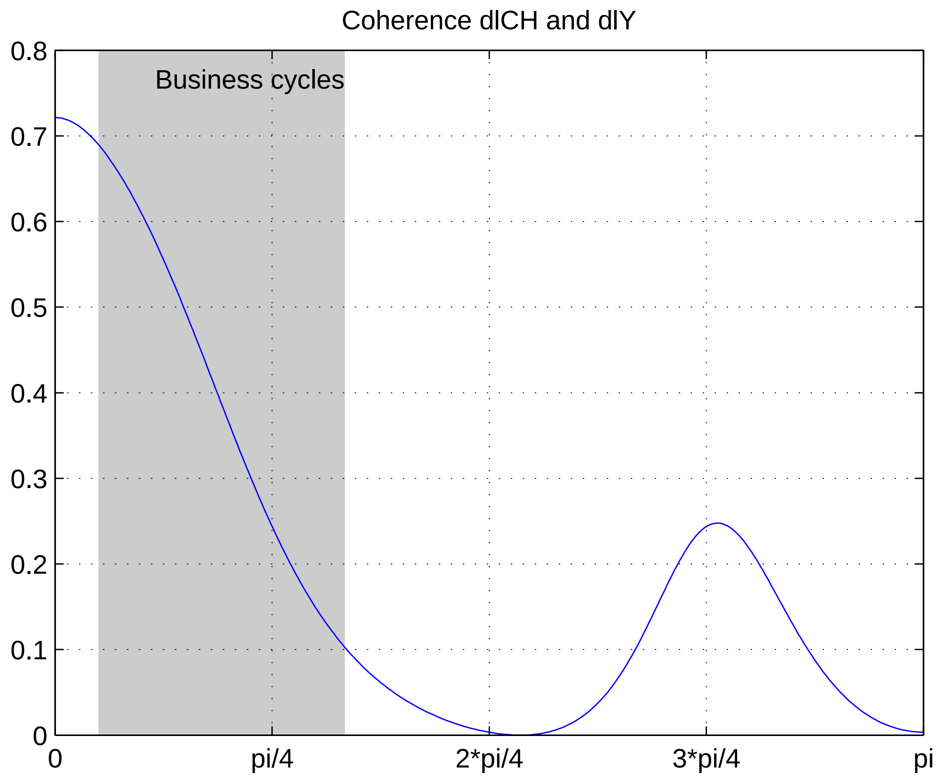
```
freq = 0:0.01:pi;
[S,D] = xsf(v,freq); 3

sx = S(1,1,:);
sy = S(2,2,:);

%{
sxy = S(1,2,:);
c = real(sxy); % cospectrum
q = imag(sxy); % quadrature spectrum
R = sqrt(c.^2+q.^2); % gain
thetaw = atan(-q./c); % phase
C = abs(sxy).^2./(sx.*sy); % coherence
%}

figure();
h = freqplot(freq,sx(:));
grid('on');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Power spectrum of dlCH');

figure();
h = freqplot(freq,sy(:));
grid('on');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Powe spectrum of dlY');

C = xsf2coher(S); 4
c = C(1,2,:);

figure();
h = freqplot(freq,c(:));
grid('on');
```

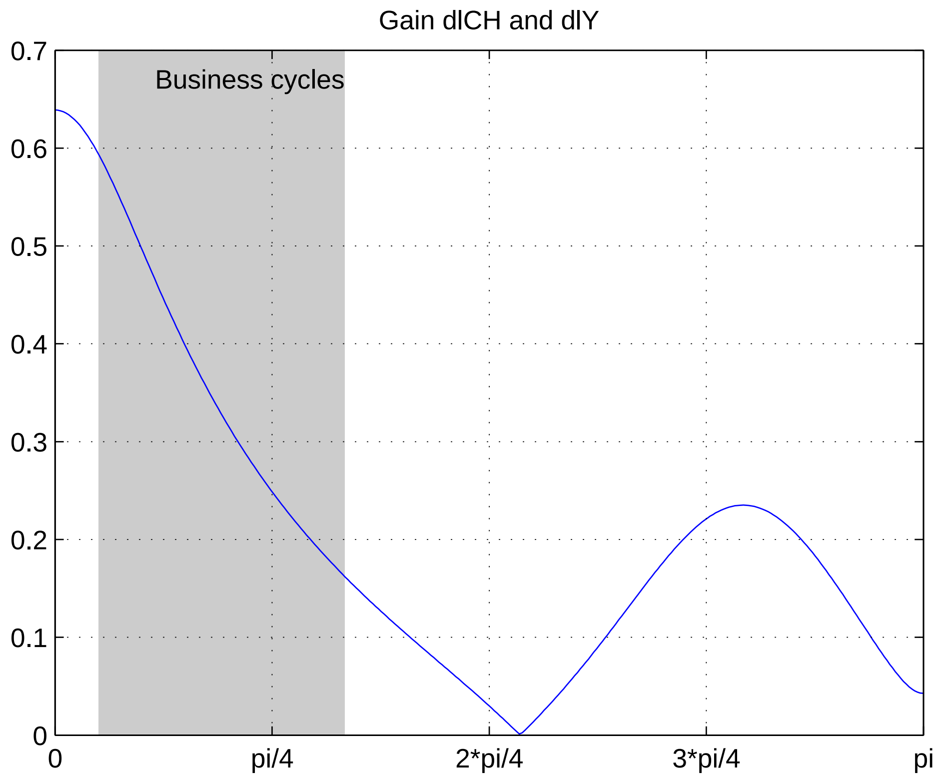
```
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Coherence dlCH and dlY');

G = xsf2gain(S); ⑤
g = G(1,2,:);

figure();
h = freqplot(freq,g(:));
grid('on');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Gain dlCH and dlY');
```





Powe spectrum of dlY
x 10^−4
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Business cycles
0
pi/4
2*pi/4
3*pi/4
pi


Coherence dlCH and dlY
Business cycles
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
pi/4
2*pi/4
3*pi/4
pi



8 Phase shift

$\theta(\omega)$ is called the phase function.

Computing and plotting the phase shift is a little bit trickier. Because the sine and cosine functions are periodic, we cannot tell if a particular phase shift is x or $x + 2\pi$, or $x - 2\pi$, etc. In other words, it's always a bit arbitrary choice. This also give us the freedom to change the phase shifts (by adding or subtracting multiples of 2π) at discontinuous points to make the graph smooth.

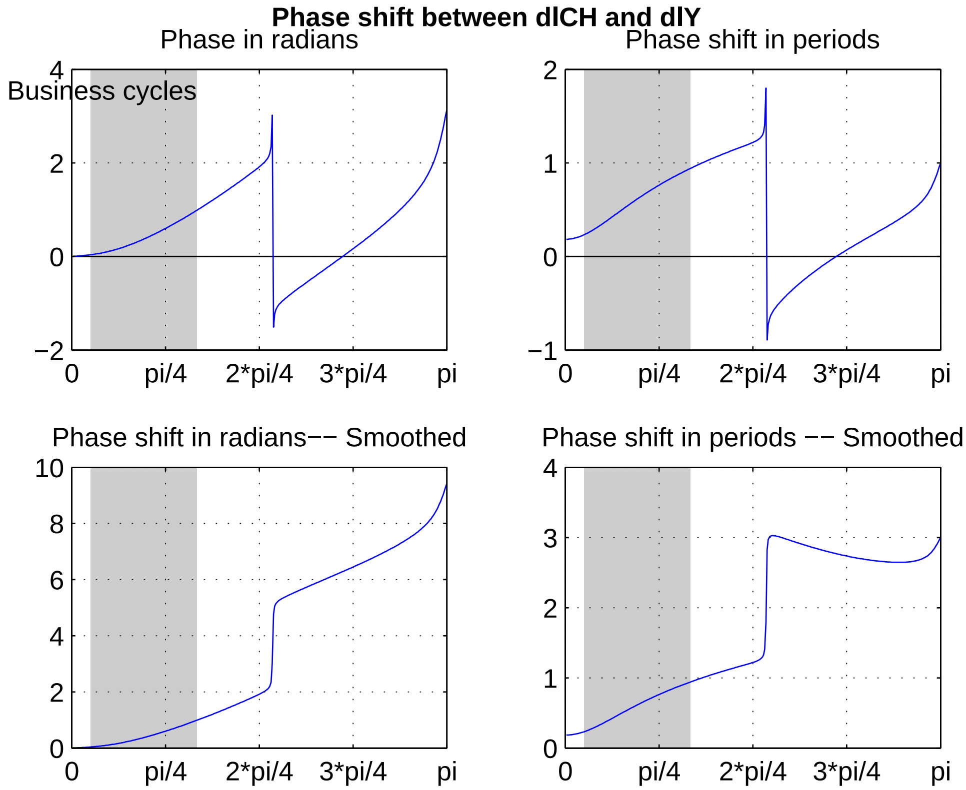
Note also that we can plot the phase shifts in two different units: either in radians or in periods (quarters here).

In the code below, we do the following:

- ...
- ...
- etc...

```
[Rad,Per] = xsf2phase(S,freq);
rad = Rad(1,2,:);
per = Per(1,2,:);
```

```

[Rad1,Per1] = xsf2phase(S,freq,'unwrap=',true);
rad1 = Rad1(1,2,:);
per1 = Per1(1,2,:);

figure();

subplot(2,2,1);
h = freqplot(freq,rad(:));
zeroline();
grid('on');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Phase in radians');

subplot(2,2,2);
h = freqplot(freq,per(:));
zeroline();
grid('on')
highlight(2*pi./[40,6]);
title('Phase shift in periods');

subplot(2,2,3);
h = freqplot(freq,rad1(:));
zeroline();
grid('on')
highlight(2*pi./[40,6]);
title('Phase shift in radians-- Smoothed');

subplot(2,2,4);
h = freqplot(freq,per1(:));
zeroline();
grid('on');
highlight(2*pi./[40,6]);
title('Phase shift in periods -- Smoothed');

ftitle('Phase shift between dlCH and dlY');
```



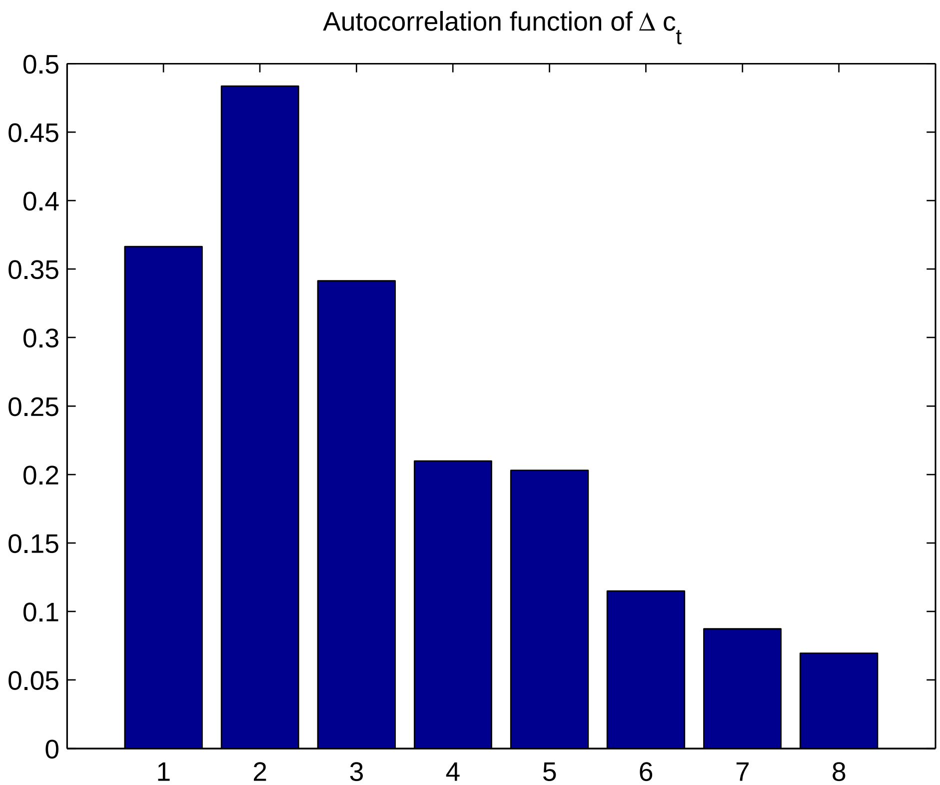
9 Auto- and cross-correlations

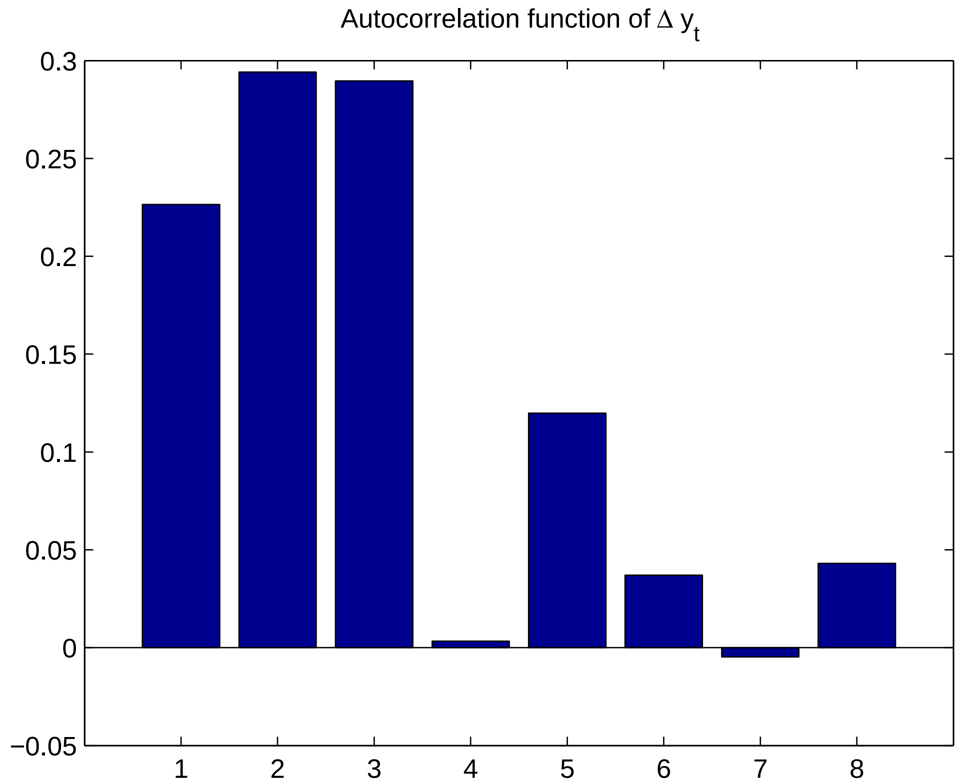
6 `vec` function helps us to get rid off the extra dimension in the multidimensional matrix. Matlab counterpart is `squeeze`.

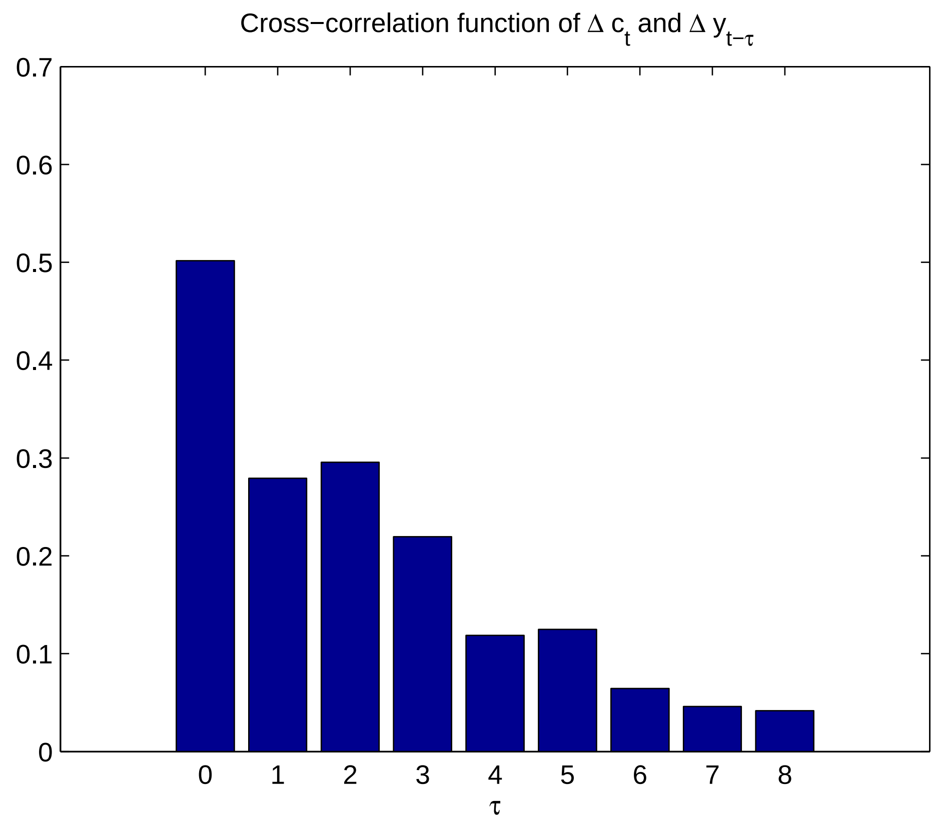
7 The `acf` function returns `R` that contains all auto and cross correlations in a 3-dimensional matrix where the third dimension is the lag 0 to 8.

```
vec = @(x) x(:); 6
[C,R] = acf(v,'order',2*varorder); 7
acCH = vec(R(1,1,:));
acY = vec(R(2,2,:));
figure;
bar(acCH(2:end));
title('Autocorrelation function of \Delta c_t');
figure;
bar(acY(2:end));
title('Autocorrelation function of \Delta y_t');
ccCHY = vec(R(1,2,:));
figure;
```

```
bar(0:8,ccCHY);
title('Cross-correlation function of \Delta c_t and \Delta y_{t-\tau}');
xlabel('\tau');
```




Autocorrelation function of Δ y_t
0.3
0.25
0.2
0.15
0.1
0.05
0
−0.05
1
2
3
4
5
6
7
8

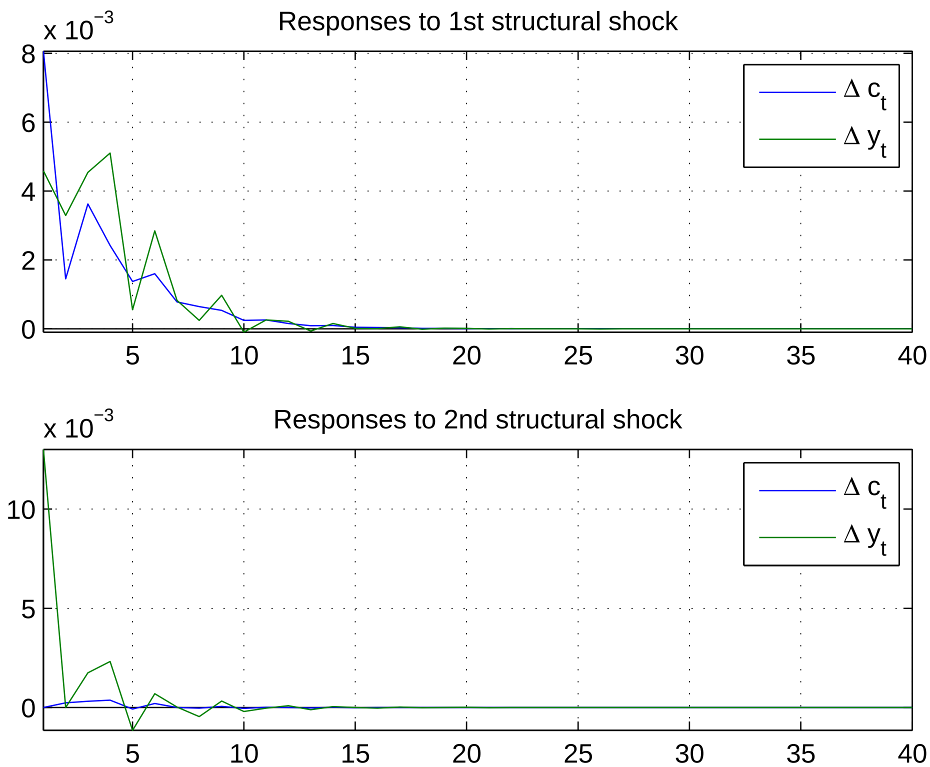


10 Impulse responses

We assume Cholesky decomposition. There is no economic content in this identification. Hence, this is to illustrate how to compute the impulse responses.

```
[w,data,B] = SVAR(v);
[Phi,Psi,irf,cirf] = srf(w,40);

figure();

subplot(2,1,1);
  plot(1:40,irf{:,:,1});
  grid('on');
  title('Responses to 1st structural shock');
  legend('\Delta c_t','\Delta y_t');
  axis('tight');
  zeroline();

subplot(2,1,2);
```

```
  plot(1:40,irf{:,:,2});
  grid('on');
  title('Responses to 2nd structural shock');
  legend('\Delta c_t','\Delta y_t');
  axis('tight');
  zeroline();
```



11 Forecast error variance decompositions

The same caveat as in impulse response analysis applies here too.

8 the last parameter is the fid file name indicator. To get the output to the screen (stdin), use 1.

```
[fevda,fevdr,tsa,tsr] = fevd(w,40);
fid = fopen('fevd.txt','w');
for h = [1,4,20,40]
  tabulatefevd(fevdr,h,{'1st shock','2nd shock'},{'dlC', 'dlY'},1); 8
end;
fclose(fid);
%
```

```
horison: 1
                        1st shock    2nd shock
dlC                          1.00         0.00
dlY                          0.11         0.89
horison: 4
                        1st shock    2nd shock
dlC                          1.00         0.00
dlY                          0.31         0.69
horison: 20
                        1st shock    2nd shock
dlC                          1.00         0.00
dlY                          0.33         0.67
horison: 40
                        1st shock    2nd shock
dlC                          1.00         0.00
dlY                          0.33         0.67
```

12 Confidence bounds by bootstrapping

In order to evaluate the fit of the model moments to the data moments we need confidence bounds around the data moments. Bootstrapping, ie drawing from estimated shock innovations is a natural candidate approach for statistical inference we are going to apply.

There is one caveat: we need a statistical model to perform bootstrapping. Therefore, we need to rely on (V)AR model even for autocorrelation.

Iris provides a neat and efficient way of performing bootstrap (see, for example, Jaromir's tutorial on VAR:

http://code.google.com/p/iris-toolbox-project/wiki/Jaromirs#VAR_basics

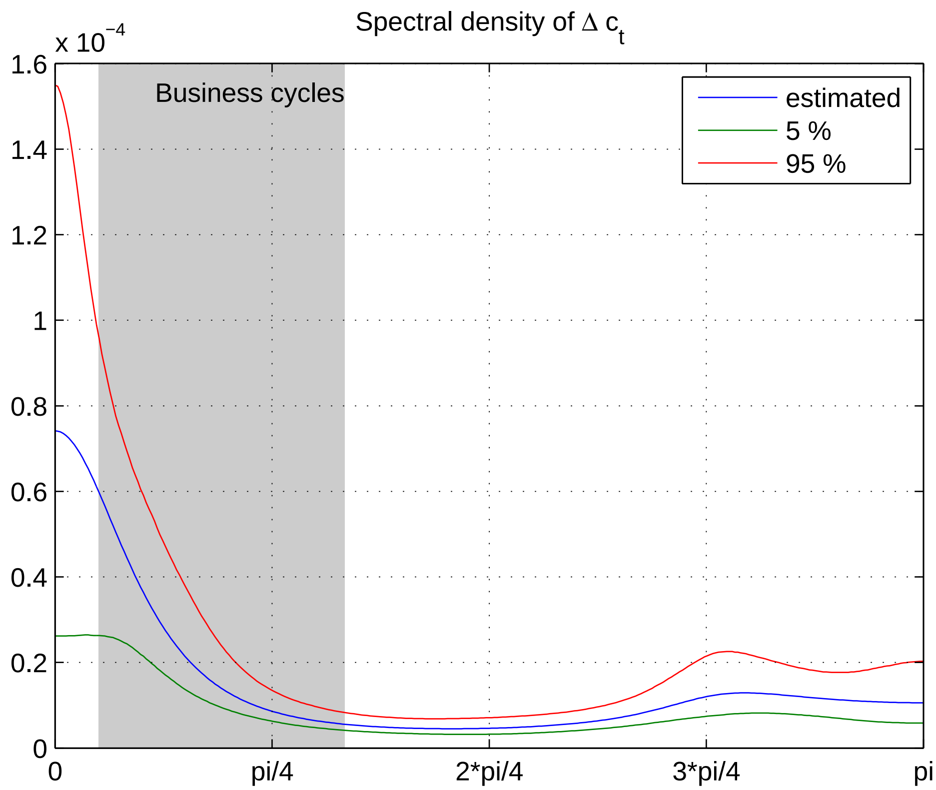
9 Note that we simulate/bootstrap nDraws draws a and, hence, we have nDraws different VARs embedded into single VAR object. Neat, isn't it!

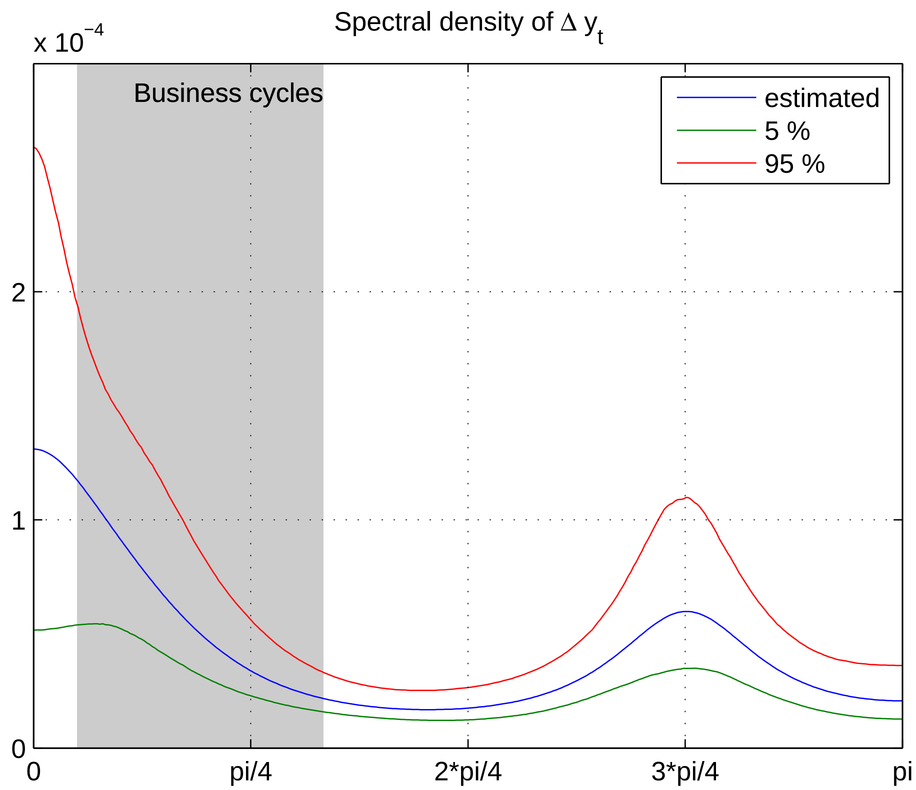
```
reset(RandStream.getDefaultStream);
dboot = resample(v,data,myRange,nDraws,'wild',true); 9
Nv = VAR(); % bootstrapped VAR
Nv = estimate(Nv,dboot,myRange,'order',varorder); a
index = isstationary(Nv);
%clear dboot;
Nv(~index) = []; % kills explosive parameterizations
```

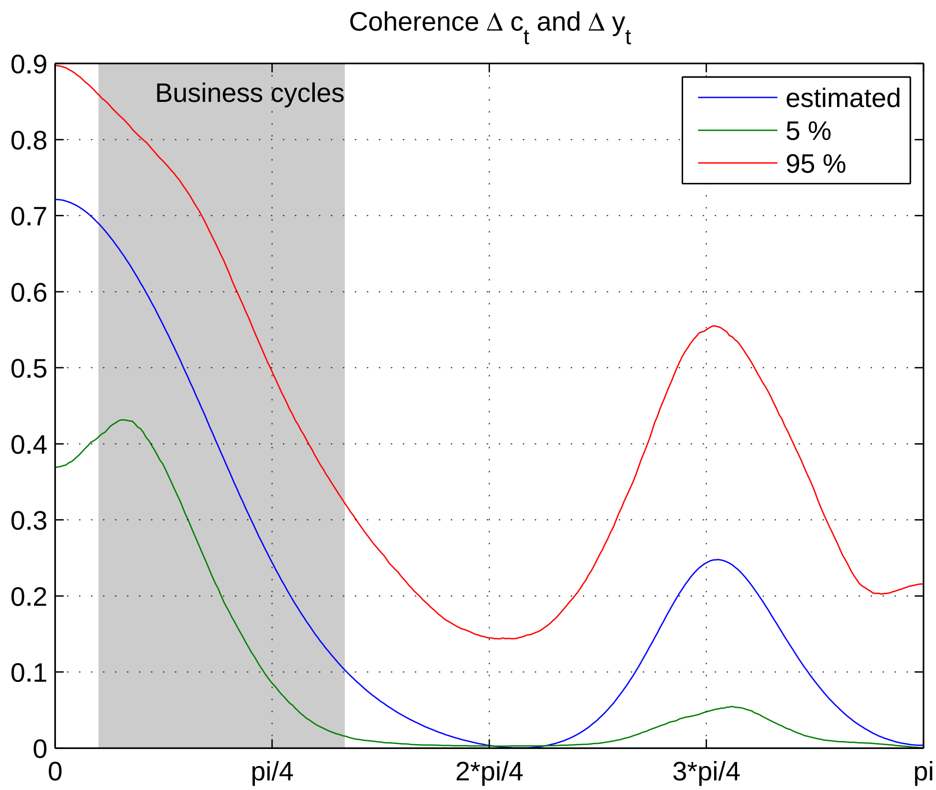
13 Regraphing spectral stuff with confidence bounds

Next compute the spectral densities and coherence with confidence bounds. The percentiles are obtained with `prctile` (see b) function. Statistical Toolbox has simular function, but I include a free one. My guess is that is sorts the draws and returns the percentiles that are needed.

```
[NS,ND] = xsf(Nv,freq);
clear ND; % not needed
Nsx = squeeze(NS(1,1,:,:));
Nsy = squeeze(NS(2,2,:,:));
Nsx = prctile(Nsx',[5 95])'; b
Nsy = prctile(Nsy',[5 95])';
figure();
freqplot(freq,[squeeze(sx) Nsx]);
grid('on');
legend('estimated','5 %','95 %');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Spectral density of \Delta c_t');
figure();
freqplot(freq,[squeeze(sy) Nsy]);
grid('on');
legend('estimated','5 %','95 %');
highlight(2*pi./[40,6],'caption=','Business cycles');highlight(2*pi./[40,6],'caption=','Business cycles');
title('Spectral density of \Delta y_t');
%clear Nsx Nsy;
NC = xsf2coher(NS);
Nc = squeeze(NC(1,2,:,:));
Nc = prctile(Nc',[5 95])'; %
figure();
h = freqplot(freq,[squeeze(c) Nc]);
legend('estimated','5 %','95 %');
grid('on');
highlight(2*pi./[40,6],'caption=','Business cycles');
title('Coherence \Delta c_t and \Delta y_t');
%clear Nc;
```


Spectral density of Δ c_t
x 10^{-4}
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
pi/4
2*pi/4
3*pi/4
pi
Business cycles
estimated
5 %
95 %


Spectral density of Δ y_t
x 10^-4
Business cycles
estimated
5 %
95 %
2
1
0
0
pi/4
2*pi/4
3*pi/4
pi

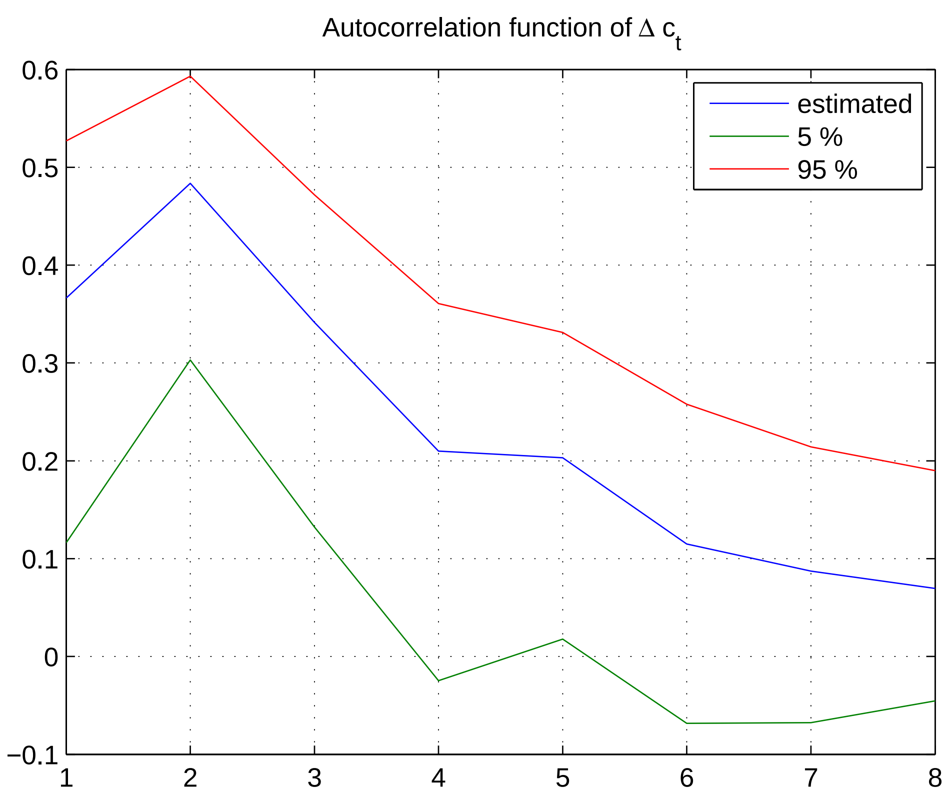


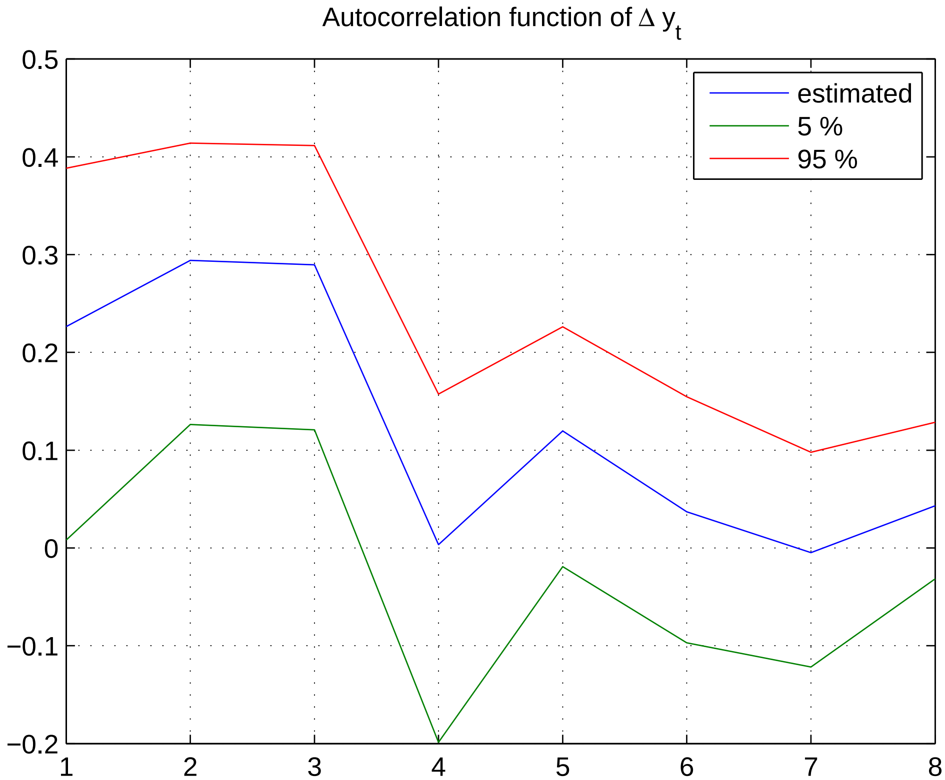
14 Regraphing auto- and crosscorrelations

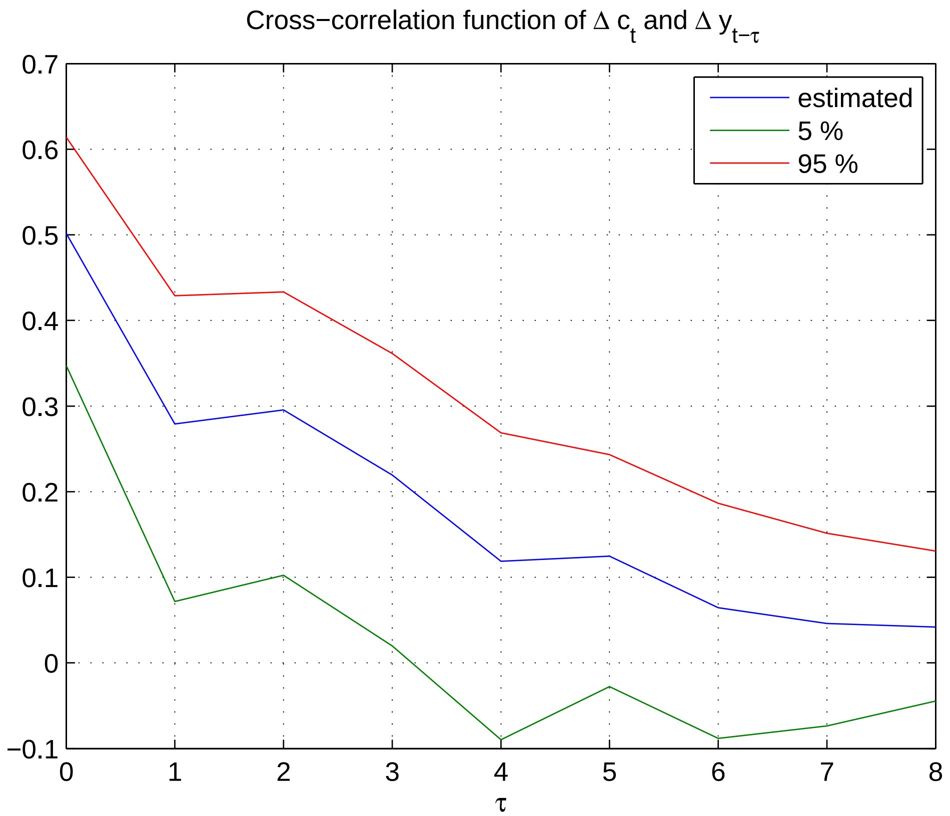
We move next to the auto- and correlations as before. Note that these correlation matrices are computed from the estimated VAR — not from the data.

```
[NC,NR] = acf(Nv,'order',2*varorder); 7
NacCH = squeeze(NR(1,1,:,:));
NacCH = prctile(NacCH',[5 95])'; %
NacY = squeeze(NR(2,2,:,:));
NacY = prctile(NacY',[5 95])'; %
figure;
plot([acCH(2:end), NacCH(2:end,:)]);
grid('on');
legend('estimated','5 %','95 %');
title('Autocorrelation function of \Delta c_t');
figure;
plot([acY(2:end), NacY(2:end,:)]);
grid('on');
legend('estimated','5 %','95 %');
title('Autocorrelation function of \Delta y_t');
```

```
NccCHY = squeeze(NR(1,2,:,:));
NccCHY = prctile(NccCHY',[5 95])'; %
figure;
plot(0:8,[ccCHY, NccCHY]);
grid('on');
legend('estimated','5 %','95 %');
title('Cross-correlation function of \Delta c_t and \Delta y_{t-\tau}');
xlabel('\tau');
%clear NC NR;
```




Autocorrelation function of Δ y_t
estimated
5 %
95 %
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
1
2
3
4
5
6
7
8



15 What next?!

Similar steps may be run for impulse responses (and FEVDs). In this case one should use the structural VAR (SVAR) object instead of VAR object. The methods/commands are still the same.

The final step should consist of calibrating/estimating the theoretical model and computing similar model moments. Those would show-up as an extra line in the previous graphs.

The beauty (or one of the beauties) of Iris lies on the fact that you may replace the VAR object by the object of the theoretical model and use exactly the same methods/commands to compute the model counterparts of the above moments.

16 Help on IRIS functions used in the file

in order of appearance:

```
help model/get
help dates/qq
help VAR/VAR
help VAR/estimate
help VAR/resample
```

```
... etc
```

Alternatively, use `idoc` instead of `doc` to display the help topic in a browser window.