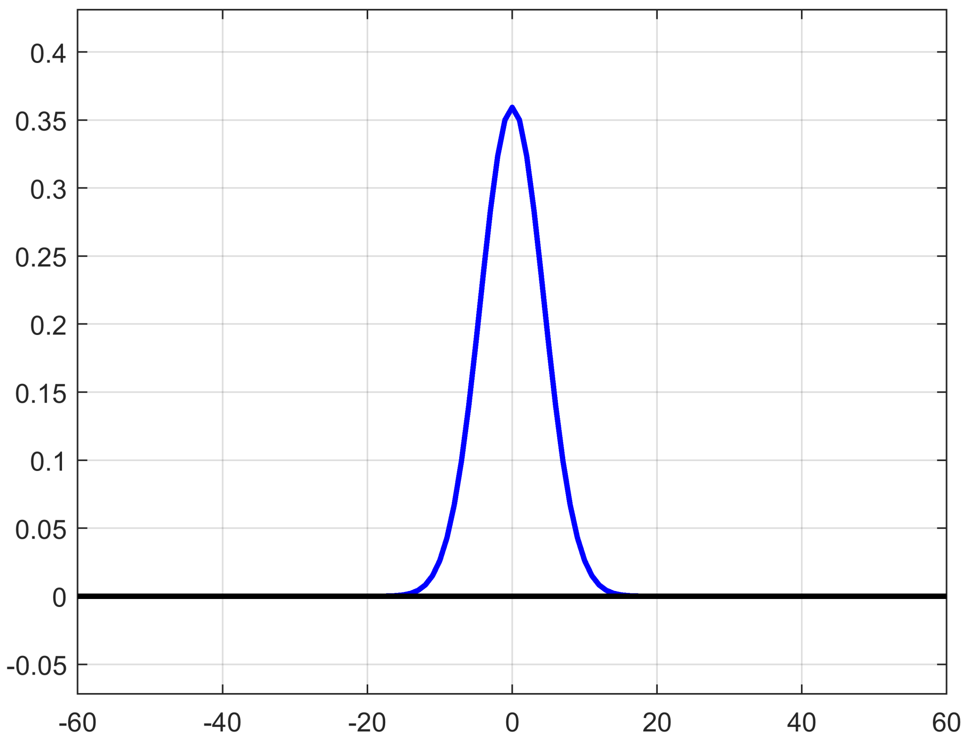
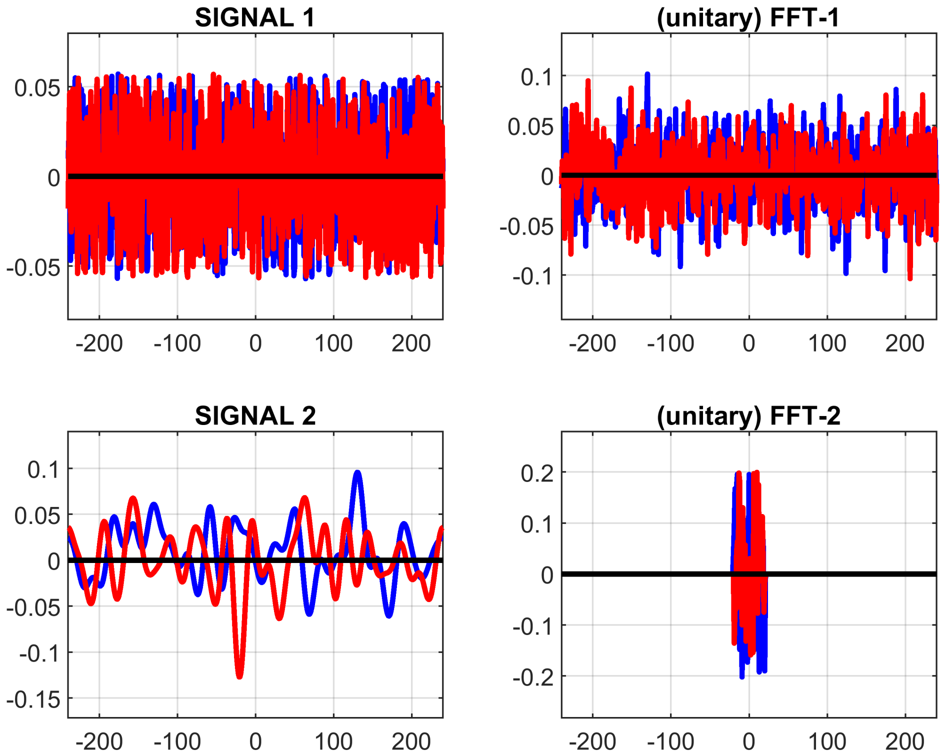
```
%  Gabdem09.mlx
%

% different representations


n = 480; sqrt(n)/2
```

```
ans = 10.9545
```

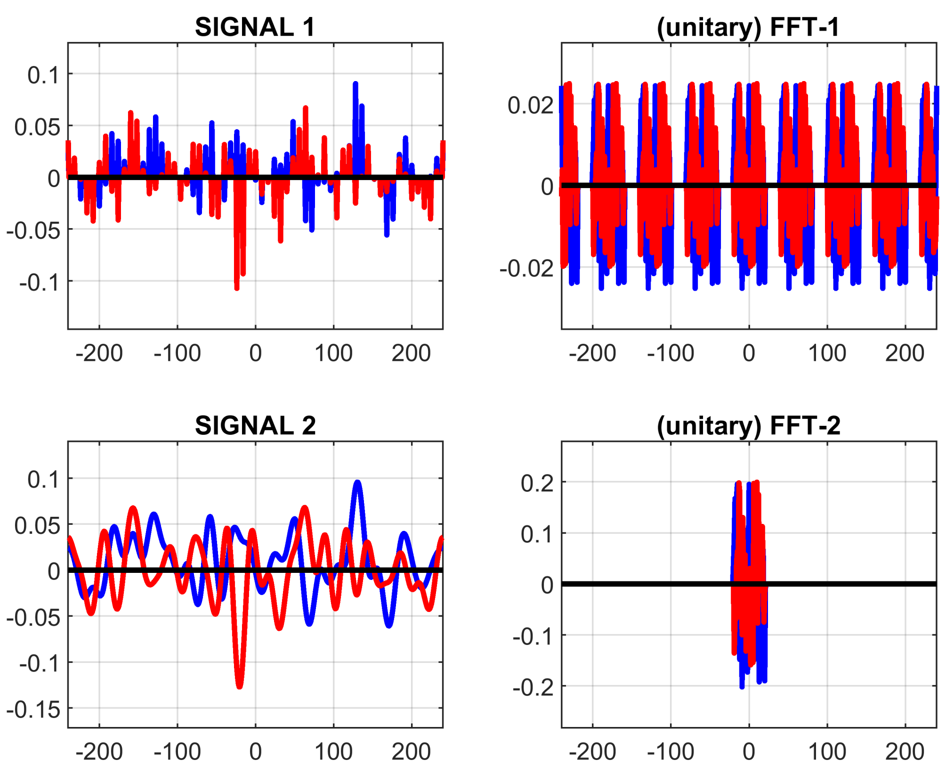
```
g = gaussnk(n);
figure;
plotc(gaussnk(120));;
```




```
figure;
xx = randcn(1,n);
yy = lowsign(n,20);
f2sp(xx,yy);
```



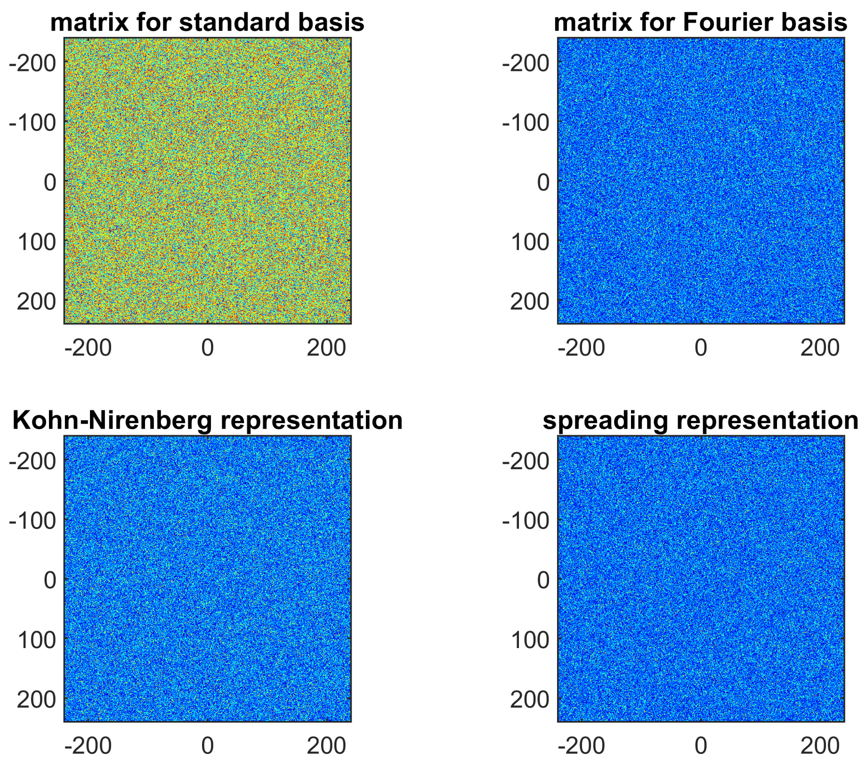

```
figure;  % changed to shah1, shah is an LTFAT routine (output columns)
f2sp(yy.*shah1(n,8),yy);
```



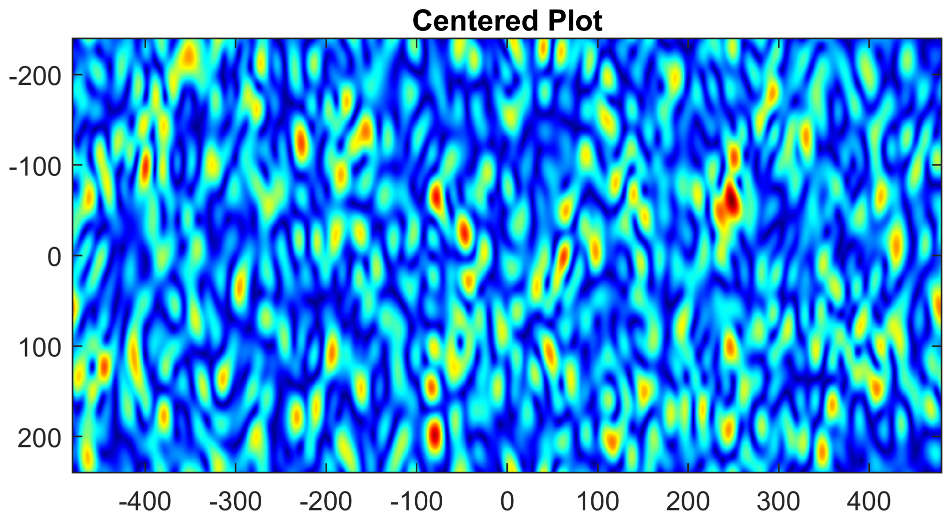
```
XX = randcn(n);figure;
norm(XX), norm(XX,'fro'), norm(XX(:)), sqrt(trace(XX*XX'))
```

```
ans = 0.0907
ans = 1
ans = 1
ans = 1
```

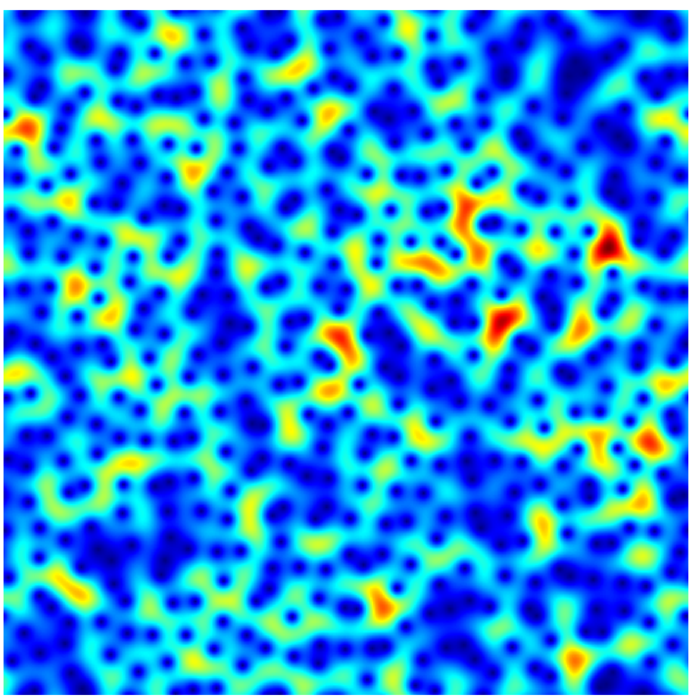
```
showmat4(XX);
```



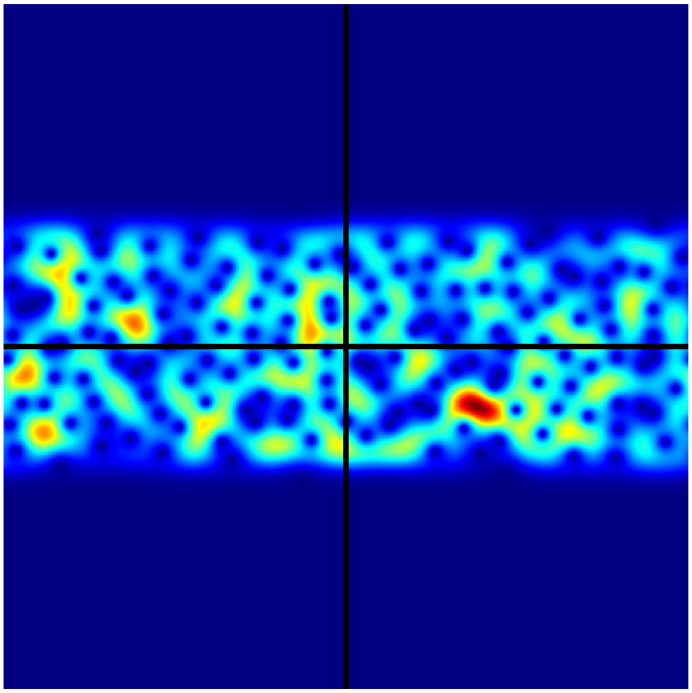

```
ZZ = lowsign(n,2*n,6,32); figure;
imgc(ZZ);
```



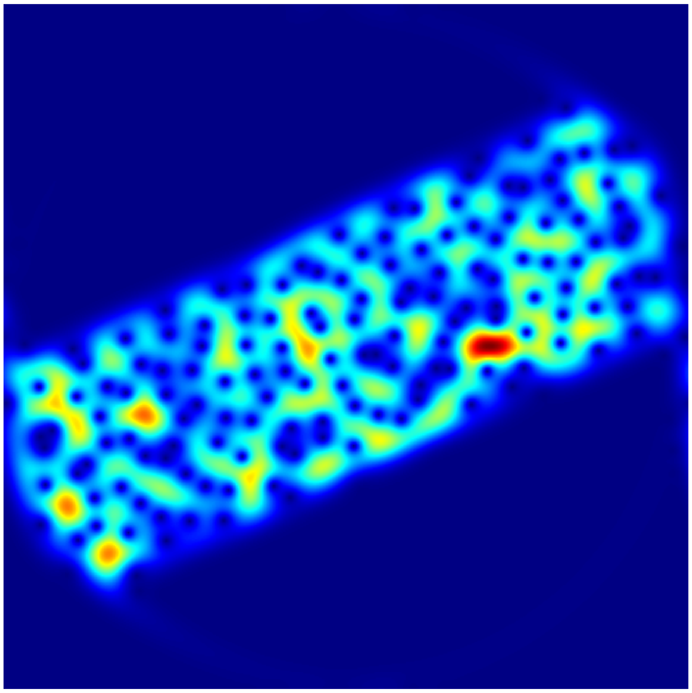

```
xx = randcn(1,n);  figure; imgstf(xx); figure;
```



```
zz = lowsign(n,80);
imgstf(zz); plotax; shg
```



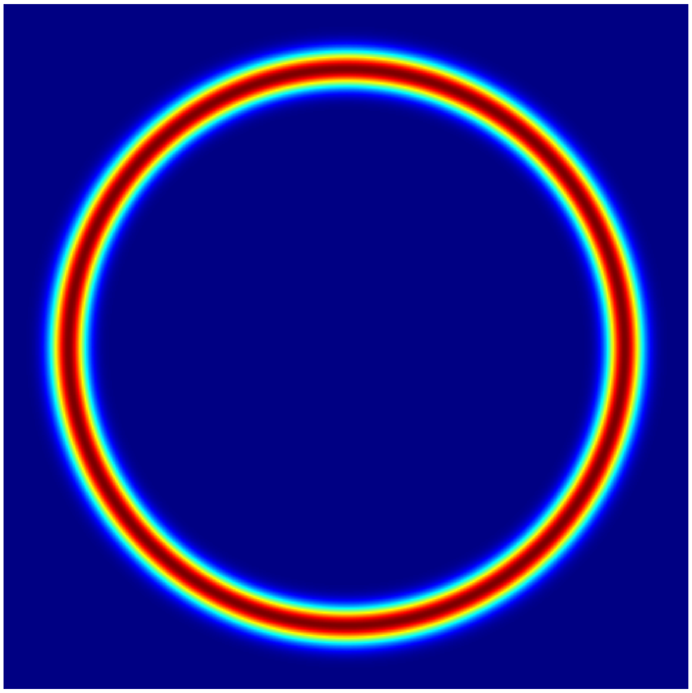
```
H = hermf(n); figure;
imgstf(hermrot(zz,25,H));
```



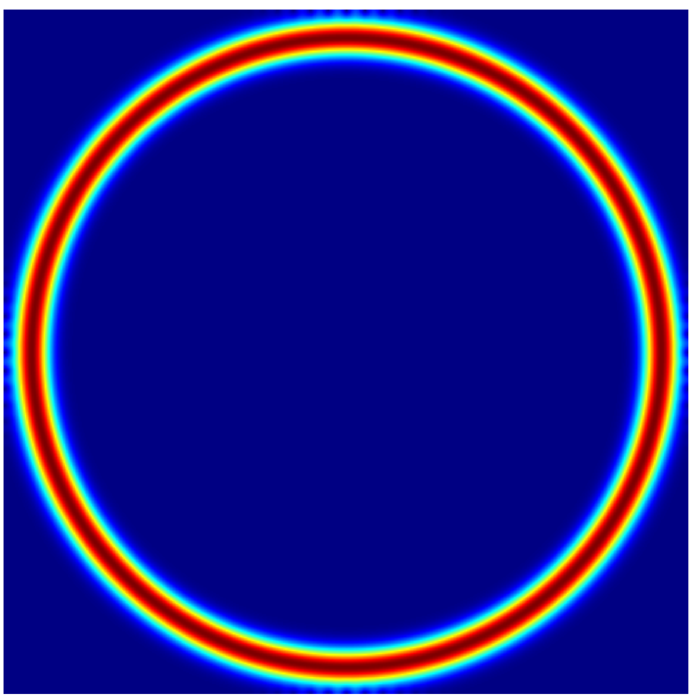
```
figure;
compnorm(g, H(1,:));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 6.451e-11
```

```
imgstf(H(250,:));
```



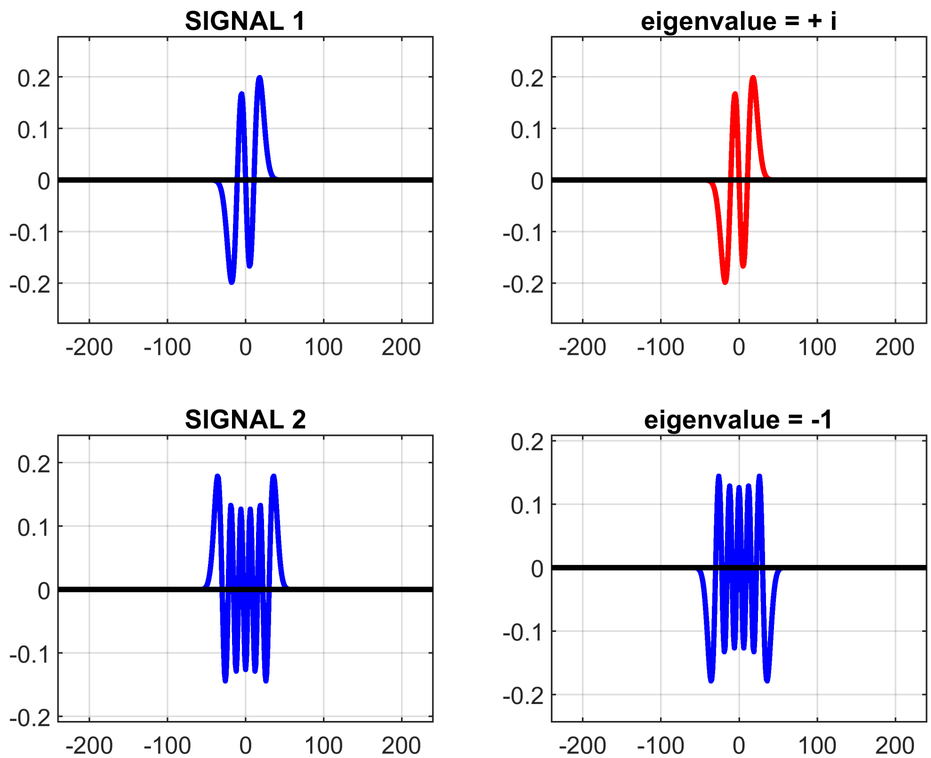
```
figure; imgstf(H(320,:));
```



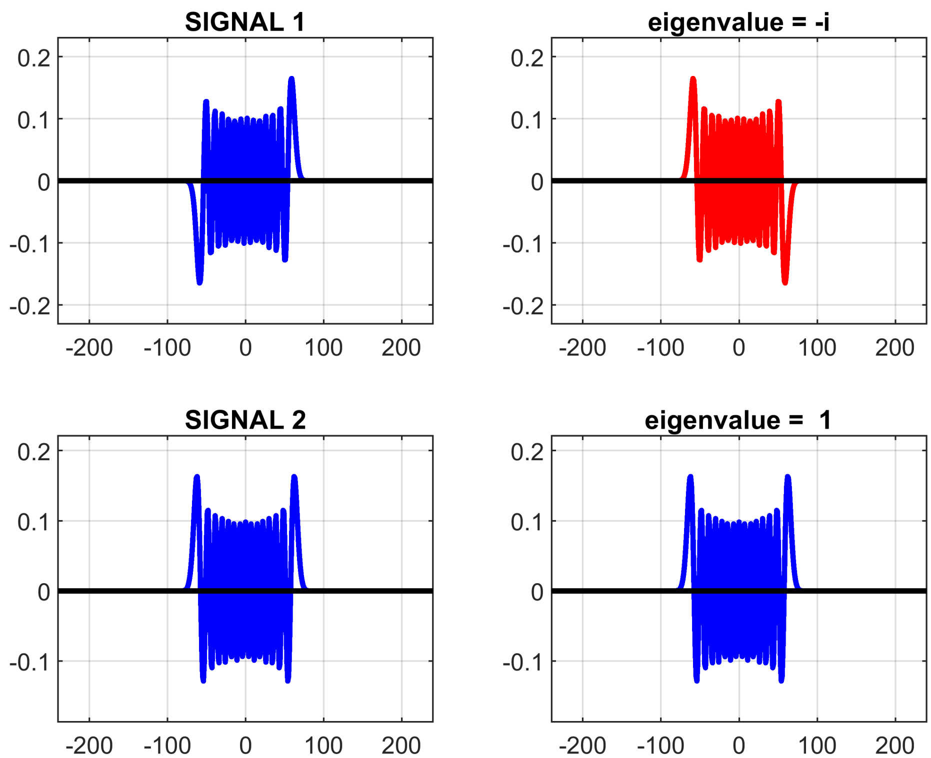
```
figure;
compnorm(H*H', eye(n));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 2.3747e-15
```

```
f2sp(H(4,:), H(11,:));subplot(222); title('eigenvalue = + i')
subplot(224); title('eigenvalue = -1');
```



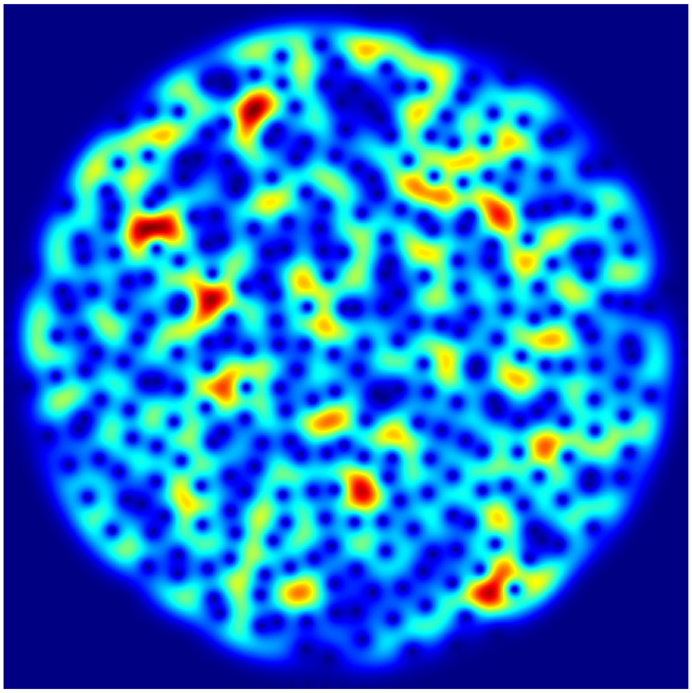

```
f2sp(H(26,:),H(29,:));
subplot(222); title('eigenvalue = -i');
subplot(224); title('eigenvalue =  1')
```




```
uu = randcn(1,10) * H(1:4:39,:);
compnorm(uu, fftu(uu));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 4.5087e-14
```

```
% projection onto the linear span of Fourier invariant functions
figure;
xxr= randcn(1,321)*H(1:321,:);
imgstf(xxr);
```

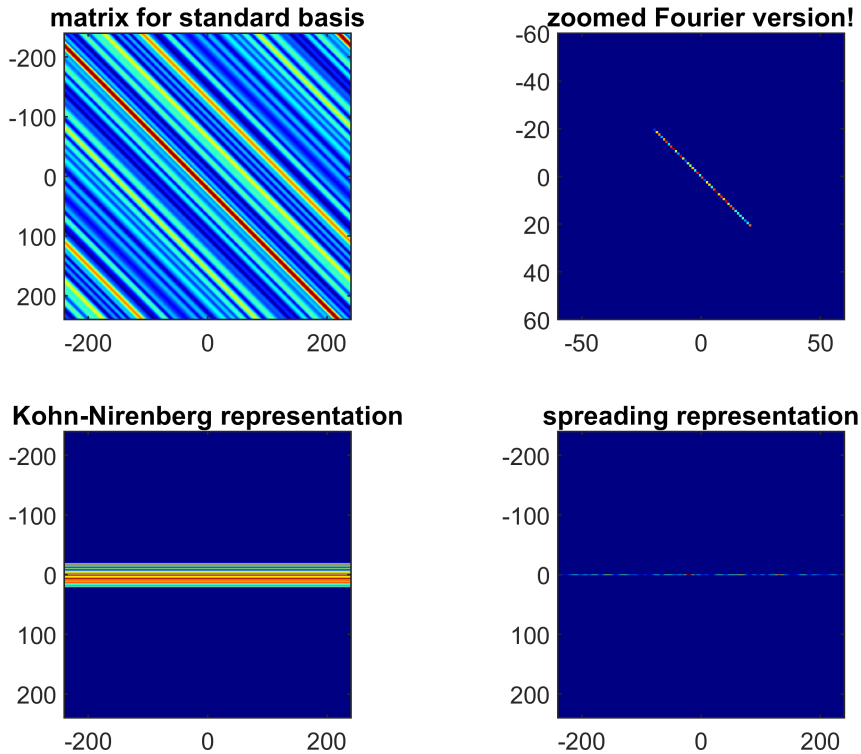


```
xxrinv = 0.25*(xxr+fftu(xxr)+ fftu(fftu(xxr))+ fftu(fftu(fftu(xxr))));
compnorm(xxrinv,  xxr*H(1:4:321,:)'*H(1:4:321,:));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 5.6609e-14
```

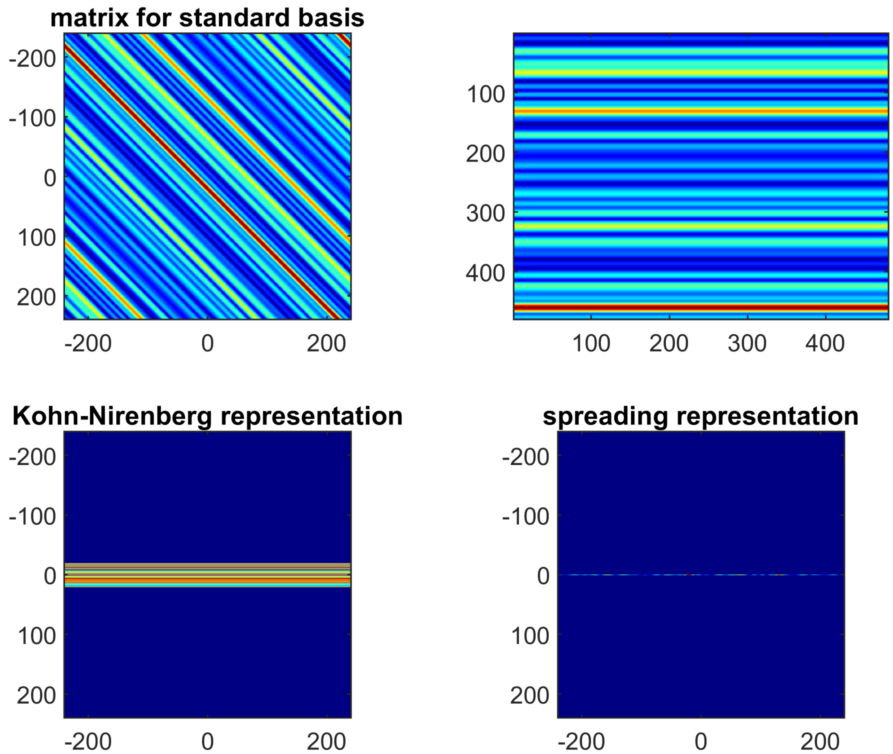
```
% showing that the projection onto the Fourier invariant part can be
%
```

```
CYY = convmat(yy);
showmat4(CYY); subplot(222); zoom(4);
title('zoomed Fourier version!')
```



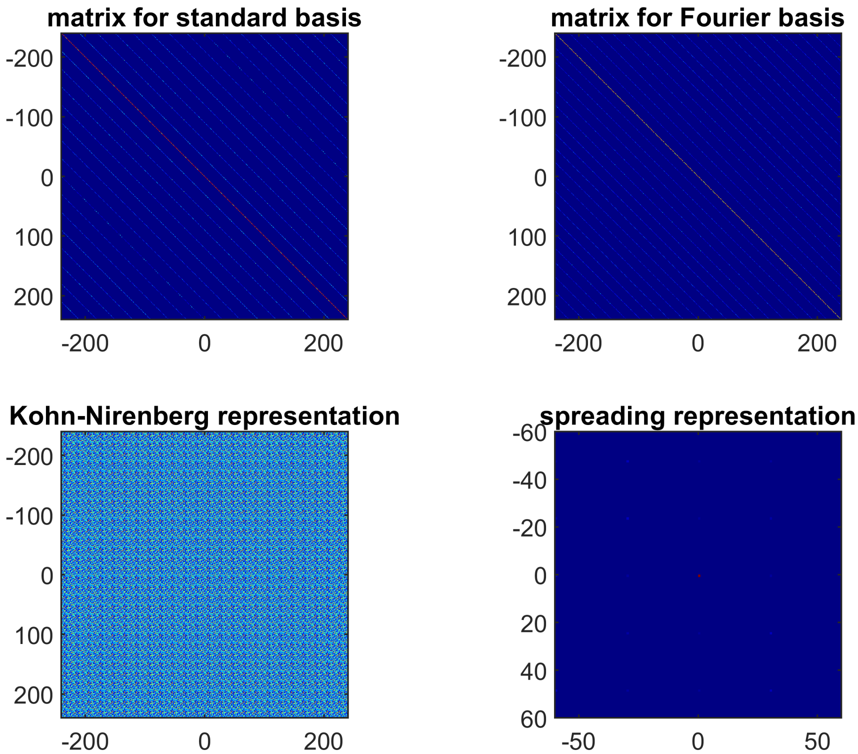

```
% you see that a convolution matrix is CIRCULANT, i.e. constant along
% (cyclic) sidediagonals. The spreading domain shows only the
% translation part, i.e. it encodes the impulse response, while
% the diaganal of the Fourier version (it is always a diagonal
% here) contains the information about the Fourier multiplier, also
```

```
imagesc(abs(sidedigm(CYY)));
```




```
% his exhibits the behaviour of the side-diagonals
% the first row is the main diagonal, diag(Sxx),
% then follows the first upper diagonal (in a czclic way)
```

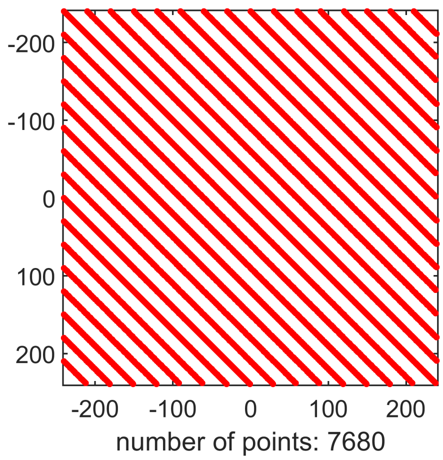
```
Sxx= gabfrmatr(xx,a,b);
showmat4(Sxx); subplot(224); zoom(4);
```

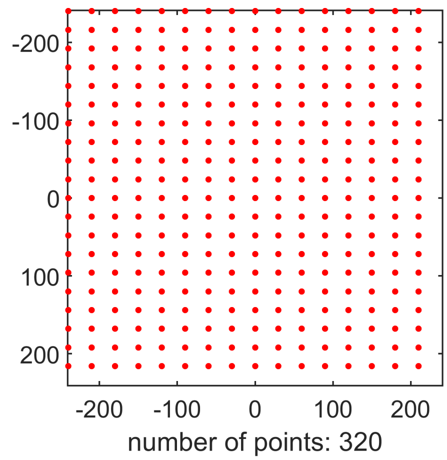



```
figure;
subplot(121); spyc(abs(Sxx)> 1000*eps); a, b
```

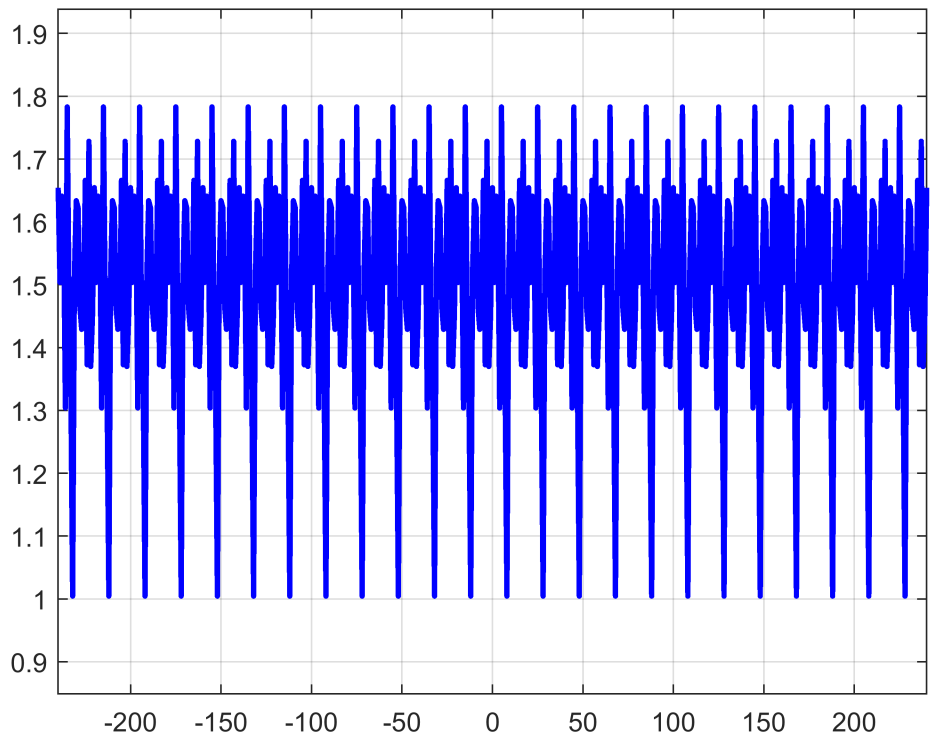
```
a = 20
b = 16
```

```
subplot(122);  spyc(abs(mat2spr(Sxx))> 1000*eps);
```



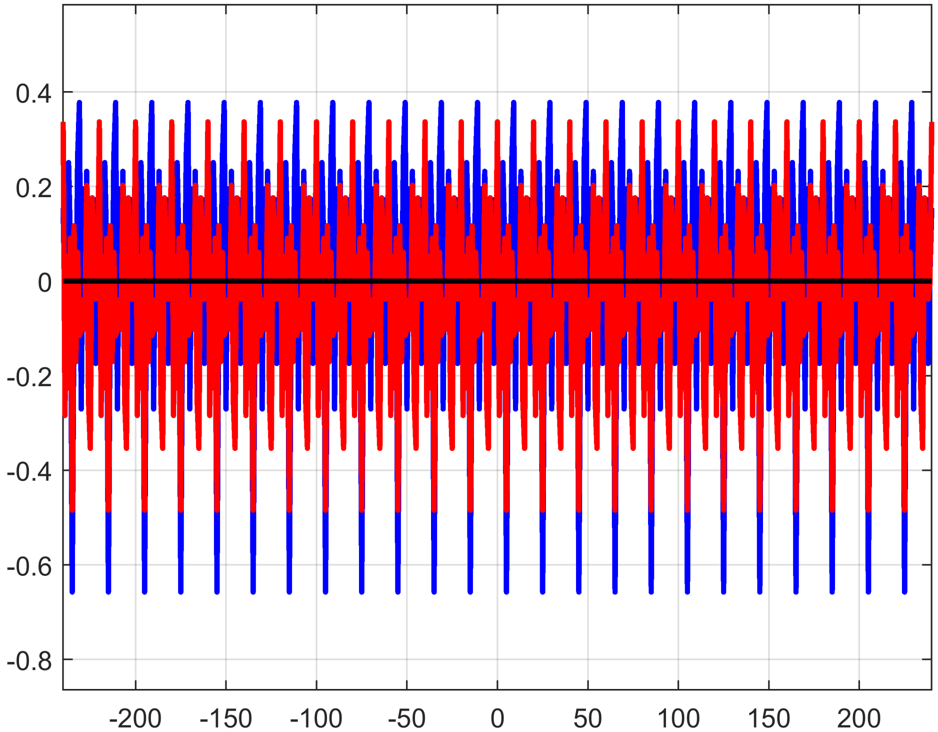



```
figure;
SDSxx = sidedigm(Sxx);
plotc(diag(Sxx));  % 24 =  n/a repetitions. So the main diagonal is
```

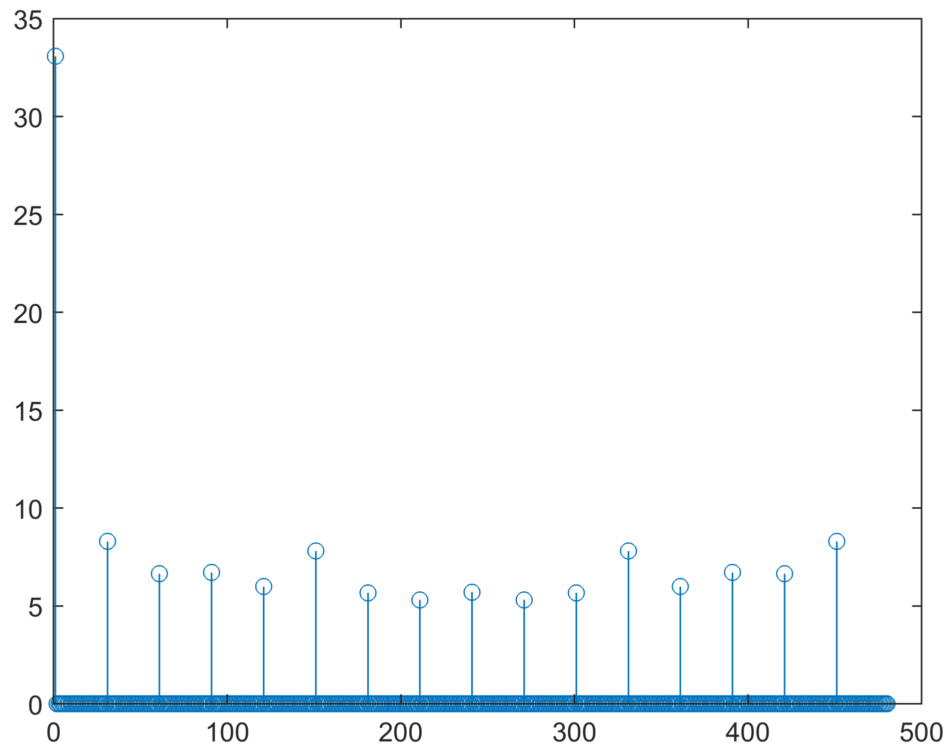



```
% real-valued, in fact even non-negative (bounded away from zero).
```

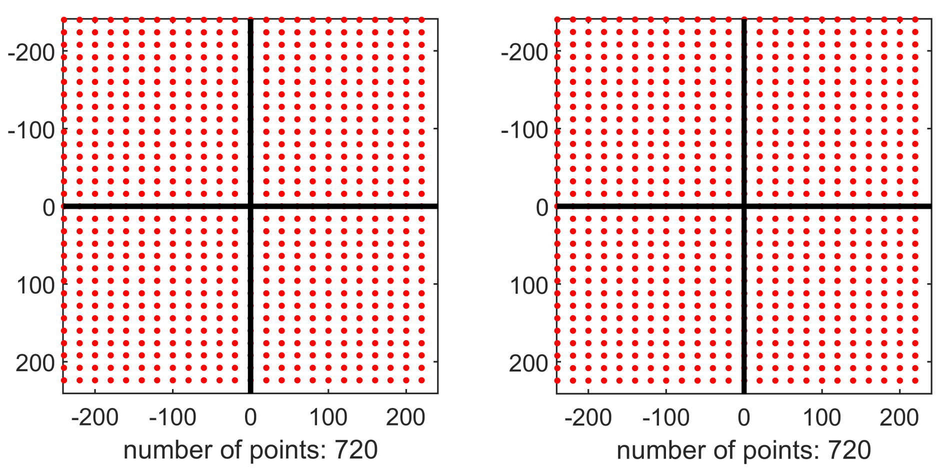
```
figure;
plotc(SDSxx(61,:));
```



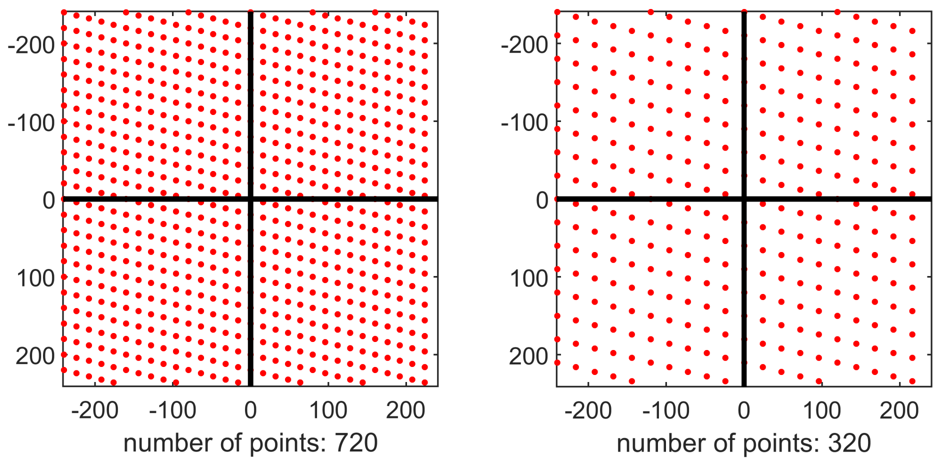

```
% showing that the other, non-trivial sidediagonals are not real-valued
% for the case of general randome window!
figure; stem(rownorms(SDSxx));
```




```
compspyc(LAM,LAMA);
```



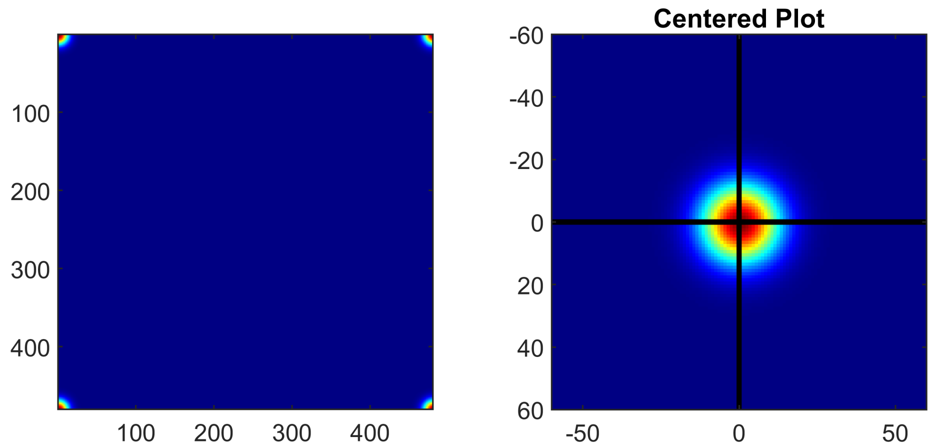
```
LAMS = sidedigm(LAM); LAMSA = adjlat(LAMS);
compspyc(LAMS, LAMSA);
```




```
Pgg = g'*g;
norm(Pgg*Pgg - Pgg)
```

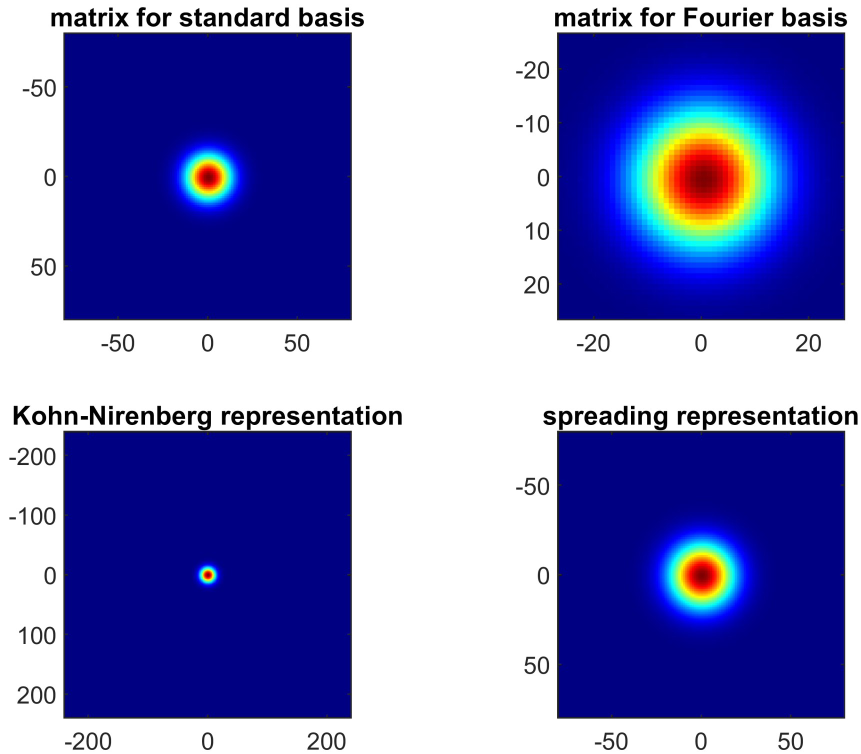
```
ans = 1.3396e-16
```

```
figure;
subplot(121); imagesc(Pgg); axis square;
subplot(122);
imgc(Pgg); plotax; zoom(4);
```




```
figure;
showmat4(Pgg);  doall zoom(3)
```

```
Executing: zoom(3)
```




```
figure;  doall zoom(5)
```

```
Executing: zoom(5)
```

```
% comparing absolute values (only then it is OK)
compnorm(abs(stft(g,g)), abs(mat2spr(Pgg)));
```

```
% quotient of norms: norm(x)/norm(y) = 480
% difference of normalized versions  = 2.5205e-16
```

```
frs = 5; tms = 3;
TXXT =  pipist(XX,tms,frs,1);
```

```
compnorm(rotrc(mat2kns(XX),frs,tms), mat2kns(TXXT));
```

```
frs2 = 7; tms2 = 12;
compnorm(pipist(TXXT,tms2,frs2), pipist(XX,15,12));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 2.902e-15
```

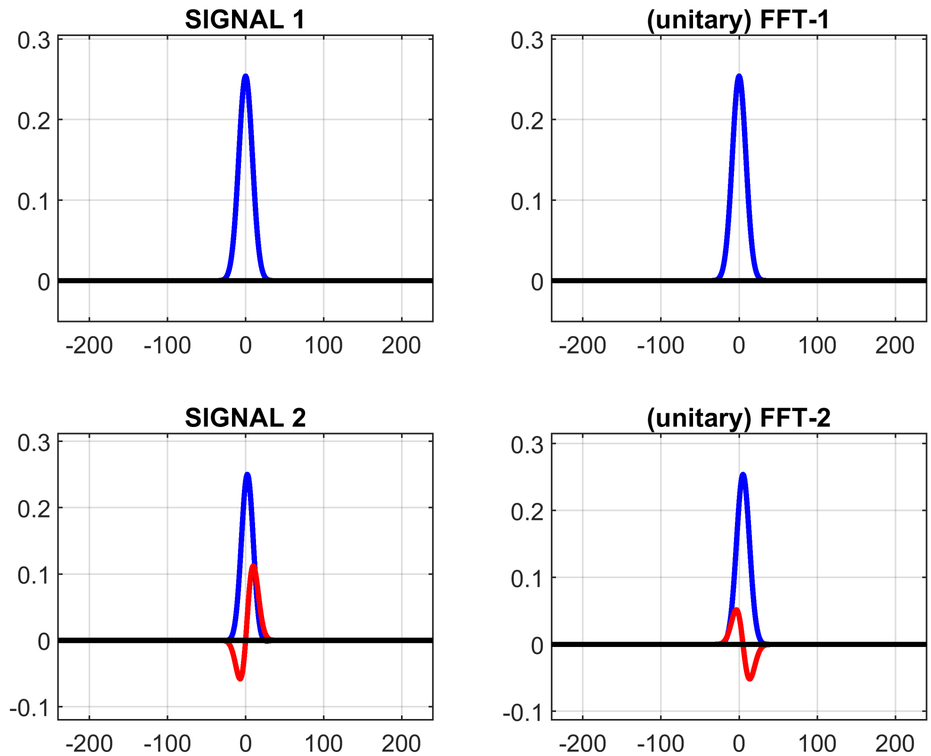
```
compnorm(rotrc(mat2kns(XX),frs,tms), mat2kns(TXXT));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 1.9942e-15
```

```
% this demonstrates the covariance property of the KNS symbol
```

```
% application of pipistar MOVES the operator by a certain amount
% in the KNS setting (same parameters)
```

```
g315 = rotmod(g,3,15); f2sp(g,g35);
```

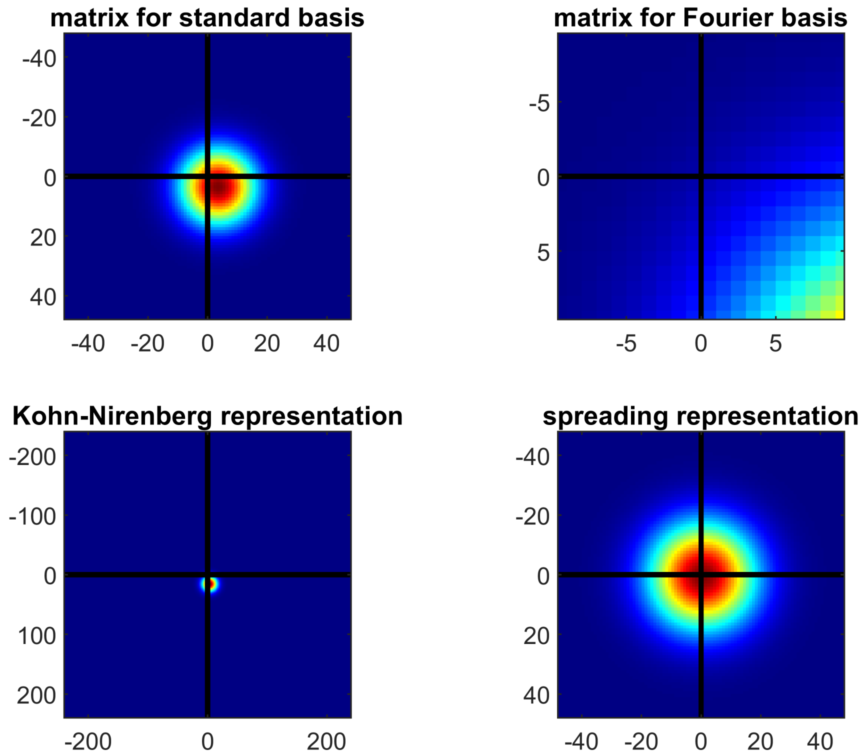



```
Pg315 = g315'*g315;
compnorm(Pg315, pipist(Pgg,3,15));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 2.8414e-15
```

```
figure; showmat4(Pg315);  doall zoom(5); doall plotax;
```

```
Executing: zoom(5)
Executing: plotax
```

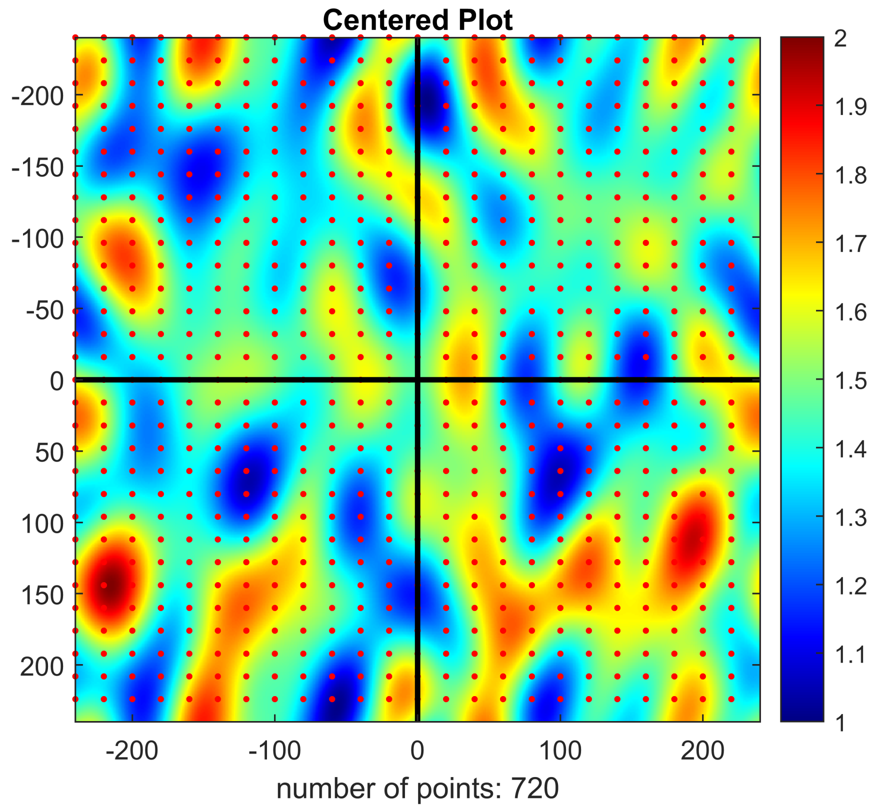



```
gt = gabtgtgj(g,a,b);
GT = gabbasp(gt,a,b);
```

```
compnorm(GT'*GT , eye(n));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 5.176e-14
```

```
W = 1 + lowzosig(n,n,4,7); figure;
imgc(W); colorbar; spych(lattp(n,a,b),'r')
```




```
compnorm(stft(xx,gt,a,b), xx * GT');  ndlat = find(lattp(n,a,b)); lnd = length(ndlat)
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 7.7916e-14
lnd = 720
```

```
Ws = W(ndlat);
compnorm(Ws,W(1:b:n,1:a:n));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 0
```

```
GMW =  gabmulmh(W,gt,a,b);
GMW1 =  gabmulmh(W(1:b:n,1:a:n),gt);
compnorm(GMW,GMW1);
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 0
```

```
GMWD =  GT'*diag(Ws)*GT;
compnorm(GMW,GMWD);
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 5.2298e-14
```

```
compnorm(gabmulmh(W,g,4,4),4*gabmulmh(W,g,8,8));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 1.6241e-05
```

```
compnorm(gabmulmh(W,g,1,1),gabmulmh(W,g,4,4));
```

```
% quotient of norms: norm(x)/norm(y) = 16
% difference of normalized versions  = 5.9297e-16
```

```
lnd
```

```
lnd = 720
```

```
GMWP = zeros(n);  TGT= zeros(n);
for jj=1:lnd;
    Pgt = GT(jj,:)'*GT(jj,:);
    TGT = TGT + Pgt ;
    GMWP = GMWP + Ws(jj)*Pgt; end;
compnorm(GMWP, GMW);
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 5.2304e-14
```

```
compnorm(TGT, eye(n));
```

```
% quotient of norms: norm(x)/norm(y) = 1
% difference of normalized versions  = 5.1799e-14
```

```
% showing that the sum of the projection operators on the (tight) Gabor
% atoms is just the identity operator
```