Codes for each problem are available at `https://github.com/kjoyce/inverse_problems/tree/master/homework04/codes`

1.

(a) Derive the formulas for the UPRE analogous to (3.18) for GCV. Add lines of code to `Deblur2dPeriodic.m` so that it implements UPRE.

Assuming the derivation from the GCV formulation, we have the following forms for $\|\boldsymbol{A}\boldsymbol{x}_\alpha - \boldsymbol{b}\|^2$ and $\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}_\alpha)$,

$$U(\alpha) = \|\boldsymbol{A}\boldsymbol{x}_\alpha - \boldsymbol{b}\|^2 + 2\sigma^2\mathrm{tr}(\boldsymbol{A}\boldsymbol{A}_{\boldsymbol{\alpha}}) = \sum_{i,j}^{n} \frac{\alpha^2 \widehat{\boldsymbol{B}}_{i,j}}{(|\widehat{\boldsymbol{a}}_{\boldsymbol{s}}|_{ij}^2 + \alpha)^2} + 2\sigma^2 \sum_{i,j}^{n} \frac{|\widehat{\boldsymbol{a}}_{\boldsymbol{s}}|_{ij}^2}{|\widehat{\boldsymbol{a}}_{\boldsymbol{s}}|_{ij}^2 + \alpha}.$$

where $\widehat{\boldsymbol{B}} = \mathrm{DFT}(\boldsymbol{B})$ and $\widehat{\boldsymbol{a}}_{\boldsymbol{s}} = n^2\mathrm{DFT}(\boldsymbol{a}_{\boldsymbol{s}})$. The Matlab codes implementing this are as follows:

```
upre_fn = @(a) a^2*sum(sum((abs(bhat/nx).^2)./(abs(ahat).^2+a).^2))+...
          2*sigma^2*sum(sum(abs(ahat).^2./(abs(ahat).^2+a)));
upre_alpha = fminbnd(upre_fn,0,1)
```

(b) Derive the formulas for the DP analogous to (3.18) for GCV. Add lines of code to `Deblur2dPeriodic.m` so that it implements DP regularization parameter selection methods.

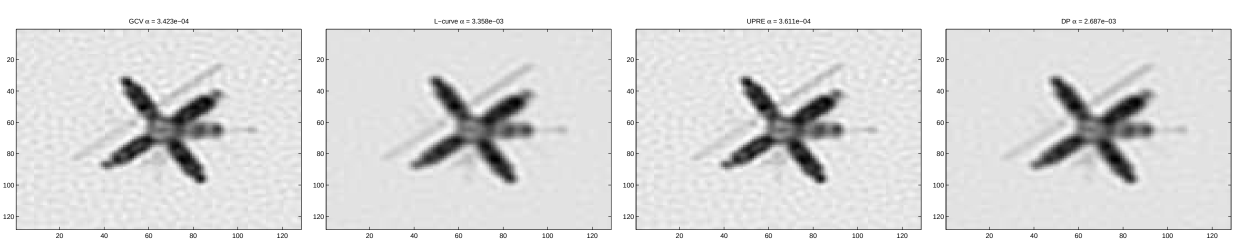
Under similar assumptions as above, we obtain

$$D(\alpha) = \|\boldsymbol{A}\boldsymbol{x}_\alpha - \boldsymbol{b}\|^2 - n^2\sigma^2 = \sum_{i,j}^{n} \frac{\alpha^2 \widehat{\boldsymbol{B}}_{\boldsymbol{i},\boldsymbol{j}}}{(|\widehat{\boldsymbol{a}}_{\boldsymbol{s}}|_{ij}^2 + \alpha)^2} - n^2\sigma^2.$$

The relevant Matlab code is:

```
dp_fn = @(a)  (a^2*sum(sum((abs(bhat/nx).^2)./(abs(ahat).^2+a).^2))-n*sigma^2)^2;
dp_alpha = fminbnd(dp_fn,0,1)
```

The reconstructions from these, and the two given parameter selection methods are given in the figure below.



2. Bardsley 3.11. *An example in which the periodic boundary conditions assumption is poor.* Implement Tikhonov regularization with periodic a boundary condition and GCV stopping rule on the example given in `PeriodicBCtest.m`. Note that the blurred image is the central 128×128 pixels of a 256×256 image and hence does not contain boundary artifacts from the periodic boundary conditions assumed for the forward model. Perform the deblurring on the 128×128 subimage. Plot a picture of the deblurred image.

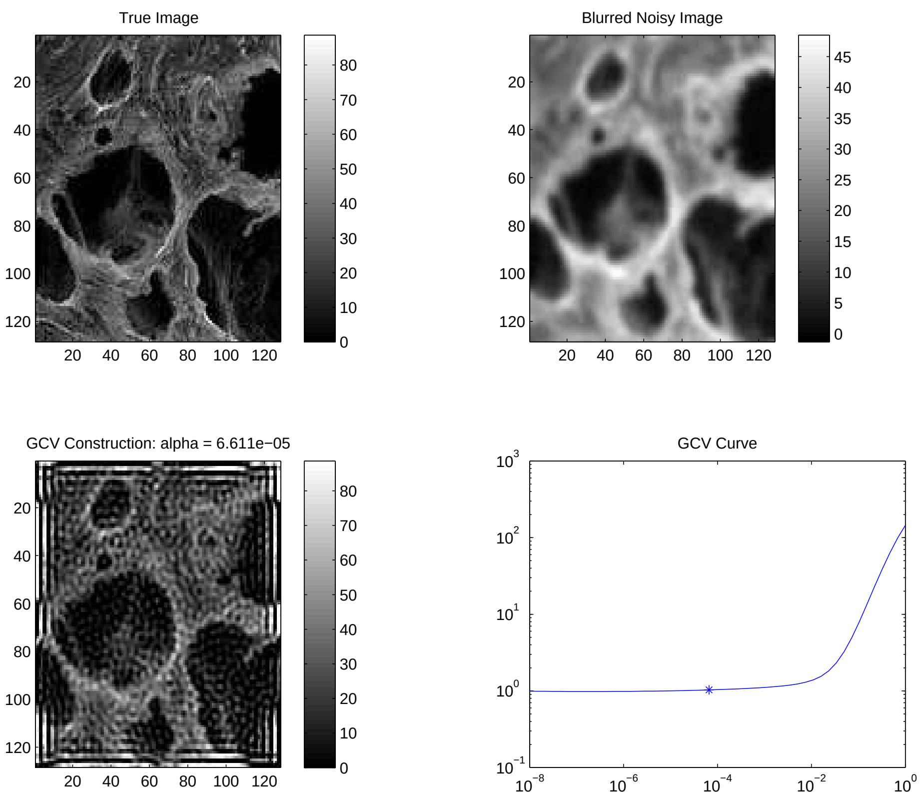
Solution:

The relevant codes for implementing this are given below:

```
n = nx*ny;
ahat = fft2(fftshift(kernel));
bhat = fft2(b);
G_fn=@(a)(sum(sum((a^2*abs(bhat).^2)./(abs(ahat).^2+a).^2))) / ...
      (n-sum(sum(abs(ahat).^2./(abs(ahat).^2+a))))^2;
gcv_alpha =  fminbnd(G_fn,0,1);

xalpha = @(a) real(ifft2((conj(ahat)./(abs(ahat).^2+a).*bhat)));
```

A plot of the reconstruction is given below. Note the severe boundary artifacts.



□

3. Modify `Deblur2DataDriven.m` so that the truncated Landweber iteration, introduced in Chapter 2, is used for solving the deblurring problem. Use the DP stopping rule, i.e. choose the first k such that $\|\boldsymbol{A}\boldsymbol{x_k} - \boldsymbol{b}\|^2 \le n^2\sigma^2$.

Solution:

The relevant codes for implementing this are given below.

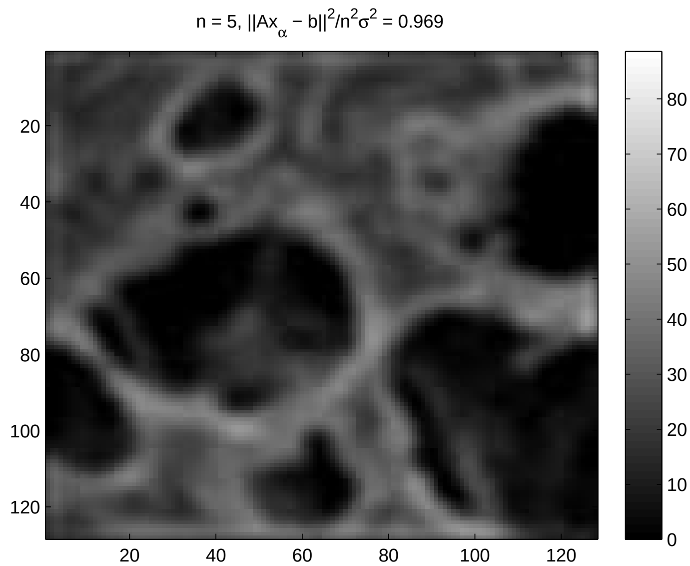
```
x = zeros(nx,ny);
tau = .8;% * 1/max(ahat(:));
figure()
```

```
n = 0;
while( true )
  n = n+1;
  resid = DA_mult(x,ahat)-b;
  x = x - tau*AtDt_mult(resid,ahat);
  if norm(resid)^2 <= nx*ny*noise^2; break; end; % end if you actually can
  end;
end
```

```
function DAx = DA_mult(x,ahat)
Ax = real(ifft2(ahat.*fft2(x)));
[nx, ny] = size(x);
DAx = Ax(nx/4+1:3*nx/4,ny/4+1:3*ny/4);
```

```
function AtDtx = AtDt_mult(x,ahat)
[nnx, nny] = size(x);
Dtx = padarray(x,[nnx/2,nny/2]);
AtDtx = real(ifft2(conj(ahat).*fft2(Dtx)));
```

An image of the resulting reconstruction is given below.



□