

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

1. Bardsley 2.9. In exercise 1.2, you were asked to modify `Deblur1d.m` so that the convolution kernel

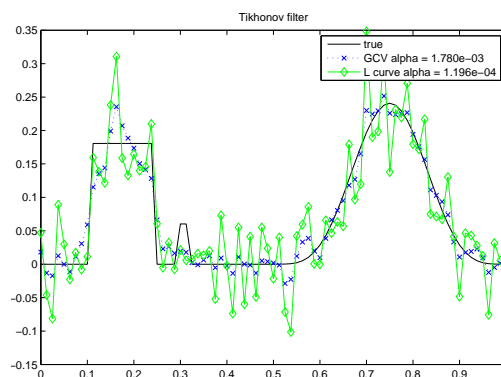
$$a(s) = \begin{cases} 100s + 10, & -\frac{1}{10} \leq s < 0, \\ -100s + 10, & 0 \leq s \leq \frac{1}{10}, \\ 0, & \text{otherwise} \end{cases}$$

is used instead to define \mathbf{A} .

(a) Use Tikhonov regularization together with GCV and L-curve to reconstruct \mathbf{x} from observations \mathbf{b} . What is the optimal regularization parameter α in each case? Which gives the better reconstruction in your opinion?

Solution:

The reconstructed solutions are given in the following figure with the optimal regularization parameter (based on the data) indicated in the legend:



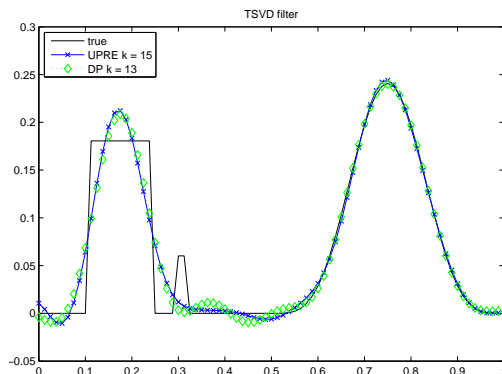
The reconstructions are nearly indistinguishable, however GCV seems to be “closer” to the true solution.

□

(b) Use TSVD regularization together with UPRE and DP to reconstruct \mathbf{x} from observations \mathbf{b} . What is the optimal regularization parameter k in each case? Which gives the better reconstruction in your opinion?

Solution:

The reconstructed solutions are given in the following figure with the optimal regularization parameter (based on the data) indicated in the legend:

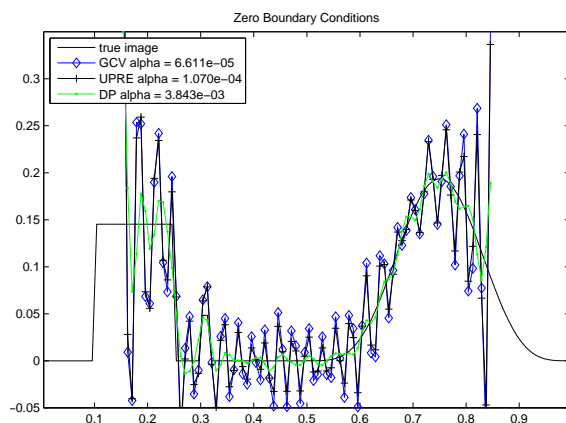


The reconstructions are practically indistinguishable, and on several runs the selection methods give the same parameter. In my opinion, UPRE gives a better reconstruction.

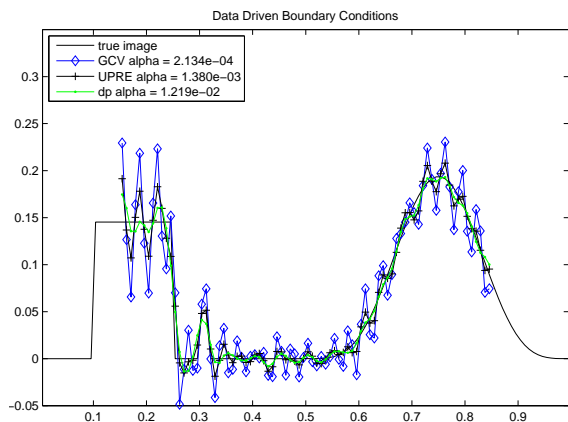
□

2. Bardsley 3.1. Modify `OnedDeblurBCs.m` so that it implements GCV and UPRE regularization parameter selection methods. How do these parameter selection methods perform in terms of the visual quality of the regularized reconstructions?

Solution:



The solutions with zero boundary conditions perform poorly especially near the boundary due to the inappropriate condition. Note that in this particular reconstruction GCV and UPRE produced nearly identical parameters.



In these reconstructions, it appears that GCV provides less aggressive filtering parameters than UPRE. Note the improvement from the zero boundary reconstruction in that the small feature to the right of the square wave would be difficult to detect with GCV and UPRE in the previous case. Whereas in the data driven boundary conditions, each parameter selection method reconstructs a solution that has the feature.

□

3. Bardsley 3.5. Use the Kronecker product properties

$$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T \quad (3.12)$$

$$(\mathbf{A} \otimes \mathbf{B})^\dagger = \mathbf{A}^\dagger \otimes \mathbf{B}^\dagger \quad (3.13)$$

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD} \quad (3.14)$$

to prove

$$\mathbf{x}_{\text{LS}} = ((\mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T) \otimes ((\mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T) \mathbf{b}, \quad (3.15)$$

and

$$\mathbf{A} = (\mathbf{U}_1 \otimes \mathbf{U}_2)(\mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_2)(\mathbf{V}_1 \otimes \mathbf{V}_2)^T. \quad (3.16)$$

Solution:

This solution is due to Cody Palmer.

Recall that the least squares solution is given by left multiplication of the psuedo inverse. I.e.

$$\begin{aligned} \mathbf{A}^\dagger \mathbf{b} &= (\mathbf{A}_1 \otimes \mathbf{A}_2)^\dagger \mathbf{b} \\ &= (\mathbf{A}_1^\dagger \otimes \mathbf{A}_2^\dagger) \mathbf{b} && \text{by (3.13)} \\ &= ((\mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T) \otimes ((\mathbf{A}_1^T \mathbf{A}_1)^{-1} \mathbf{A}_1^T) \mathbf{b}. \end{aligned}$$

For (3.16), we write $\mathbf{A}_1 = \mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T$ and $\mathbf{A}_2 = \mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^T$. Then

$$\begin{aligned} \mathbf{A} &= (\mathbf{A}_1 \otimes \mathbf{A}_2) \\ &= (\mathbf{U}_1 \mathbf{\Sigma}_1 \mathbf{V}_1^T) \otimes (\mathbf{U}_2 \mathbf{\Sigma}_2 \mathbf{V}_2^T) \\ &= (\mathbf{U}_1 \otimes \mathbf{U}_2)(\mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_2)(\mathbf{V}_1^T \otimes \mathbf{V}_2^T) && \text{by (3.14),} \\ &= (\mathbf{U}_1 \otimes \mathbf{U}_2)(\mathbf{\Sigma}_1 \otimes \mathbf{\Sigma}_2)(\mathbf{V}_1 \otimes \mathbf{V}_2)^T && \text{by (3.12).} \end{aligned}$$

□

4. Bardsley 3.6a. Derive the formulas for GCV analogous to (3.18). Add lines of code to `Deblur2dSeparable.m` so that it implements GCV.

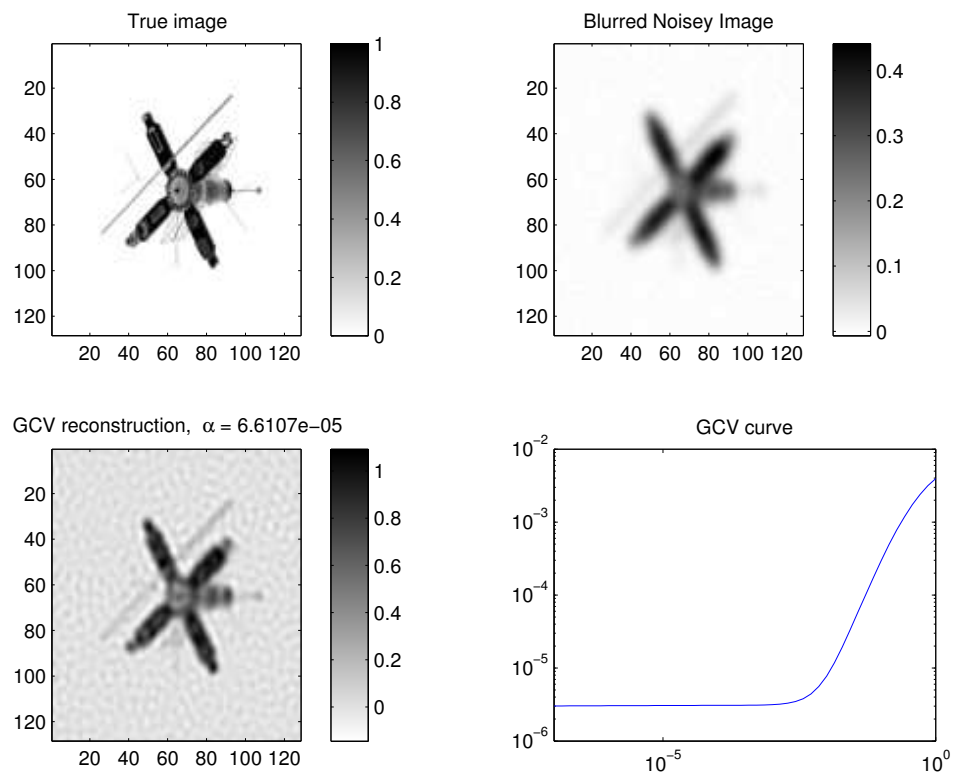
Solution:

The GCV curve for the separable kernel $\mathbf{A}_1 \otimes \mathbf{A}_2$ is given by

$$G(\alpha) = \left(\sum_{i,j=1}^n \frac{\alpha^2 ([\mathbf{U}_1^T \mathbf{B} \mathbf{U}_2]_{ij})^2}{([\boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2^T]_{ij}^2 + \alpha)} \right) \bigg/ \left(m - \sum_{i,j=1}^n \frac{[\boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2^T]_{ij}^2}{[\boldsymbol{\sigma}_1 \boldsymbol{\sigma}_2^T]_{ij}^2 + \alpha} \right).$$

The following line of code implements the GCV curve:

```
gcv_fn = @(a) a^2*sum(Utb(:).^2./(dS1dS2(:)+a).^2)/(n-sum(dS1dS2(:)./(dS1dS2(:)+a)));
```



□