Math 6397 Bayesian Inverse Problems

Problem Set 1

Due on Friday, March 24, at 10:00 PM

1 Background

Let $\mathcal{K} : \mathcal{X} \to \mathcal{Y}$ denote a compact operator. For a function $x : \mathbb{R} \to \mathbb{R}$ we consider the operator equation $\mathcal{K}x = y$,

$$(\mathcal{K}x)(s) = \int_{-\infty}^{\infty} \mathrm{ker}(s-t)x(t)\,\mathrm{d}t = y(s), \quad s \in (-\infty, \infty) \tag{1}$$

with kernel $\mathrm{ker} : \mathbb{R} \to \mathbb{R}$. The equation above represents a Fredholm first kind integral equation.

The *direct problem* associated with (1) is the following: Given the source function x and the kernel ker determine the blurred data y. The *inverse problem* associated with (1) is as follows: Given the kernel ker and the blurred data y determine the source x. We will refer to this problem as *source reconstruction problem*. Likewise, we can consider the problem of reconstructing the kernel ker from data y. We will refer to this problem as a *kernel reconstruction problem*.

1.1 Source Reconstruction in 1D

In this section, we discuss the formulation and discretization of the source reconstruction (deconvolution) problem associated with (1) in a one-dimensional setting.

Problem Formulation

We consider the kernel $\mathrm{ker} : \mathbb{R} \to \mathbb{R}$, $\mathrm{ker} =: \mathrm{ker}_\tau$, $s \mapsto \mathrm{ker}_\tau(s)$,

$$\mathrm{ker}_\tau(s) = \frac{1}{\sqrt{2\pi\tau^2}} \exp\Big(-\frac{s^2}{2\tau^2}\Big), \tag{2}$$

parameterized by $\tau > 0$. As an example for an analytic source function we select

$$x_{\mathrm{true}}(s) = \begin{cases} 0.75 & \text{for } s \in \ (1/10, 1/4), \\ 0.25 & \text{for } s \in \ (3/10, 7/20), \\ \sin(2\pi s)^4 & \text{for } s \in \ (1/2, 1), \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We will assume that we do not have access to y but only a perturbed (measured) observation y^δ. In particular, $y^\delta = y + \delta\eta$, $\eta \sim \mathcal{N}(0,1)$, $\delta > 0$. The parameter δ controls the amount of perturbation. We visualize the data associated with (1) in Fig. 1. The script used to generate this data is prbsets/deconv1D/scVizData1D.m.

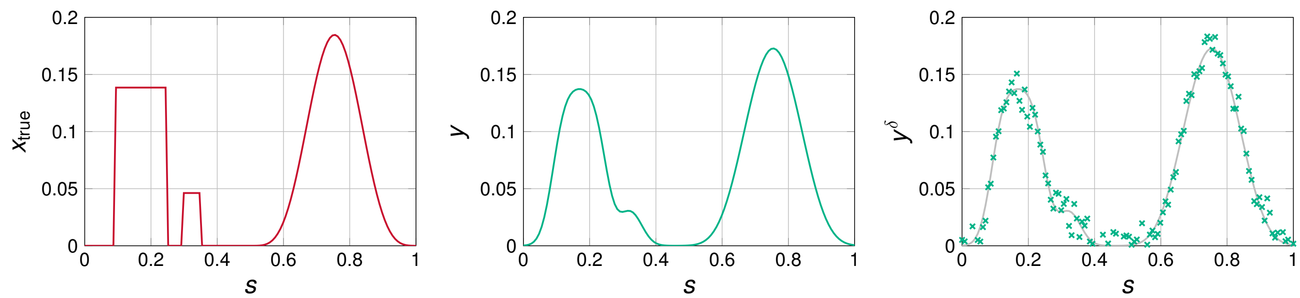

Figure 1: One dimensional deconvolution problem. Left: The true source $\mathbf{x}_{\text{true}}$ (see (3)). Middle: Convolution operator $\mathbf{K}$ applied to $\mathbf{x}_{\text{true}}$. Right: Perturbed data $\mathbf{y}^\delta$. Here, $n = 128$, δ is chosen such that the signal to noise ration γ is equal to 10, and $\tau = \frac{3}{100}$. The inverse source reconstruction problem corresponds to recovering $\mathbf{x} = \mathbf{x}_{\text{true}}$ from $\mathbf{y}^\delta = \mathbf{K}\mathbf{x}_{\text{true}} + \delta\boldsymbol{\eta}$.

Numerical Discretization

For discretization, we assume that y is measured on the interval $[0, 1]$ and $\ker_\tau$ is measured on the interval $[-1, 1]$, with $\ker_\tau(s) = 0$ for $|s| \geq 1$. Based on these assumptions, (1) reduces to

$$\int_{-1}^{2} \ker_\tau(s-t)x(t)\,\mathrm{d}t, \quad 0 \leq s \leq 1. \tag{4}$$

To discretize the integral operator in (4), we split the domain $[0, 1]$ into n intervals $[ih, (i+1)h] \subset \mathbb{R}$, $i = 1, \ldots, n$, with $h = 1/n$. Let $z(t) = \ker_\tau(s-t)x(t)$. Then, if we consider a midpoint quadrature rule, we have

$$\int_{-1}^{2} z(t)\,\mathrm{d}t \approx h \sum_{i=-n+1}^{2n} z(t_i), \quad t_i = (i - \frac{1}{2})h.$$

Assuming that $s \in [0, 1]$ is discretized on the same grid, we define $y_i := y(s_i)$, $x_i := x(s_i)$, and $\kappa_{i-j} := \ker_\tau(s_i - t_j) = \ker_\tau((i-j)h)$. With $\kappa_{i-j} = 0$ for $|i-j| \geq n$, we obtain

$$y_i = h \sum_{j=i-n+1}^{i+n-1} \kappa_{i-j} x_j = h \sum_{j=-n+1}^{n-1} \kappa_j x_{i-j}, \quad i = 1, \ldots, n,$$

for the discretization of (1). With this, we obtain

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = h \begin{bmatrix} \kappa_{n-1} & \cdots & \kappa_0 & \cdots & \kappa_{-n+1} & & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & \kappa_{n-1} & \cdots & \kappa_0 & \cdots & \kappa_{-n+1} \end{bmatrix} \begin{bmatrix} x_{-n+2} \\ \vdots \\ x_1 \\ \vdots \\ x_n \\ \vdots \\ x_{2n-1} \end{bmatrix}.$$

This is an underdetermined linear system with n equations and $3n-1$ unknowns x_i. We have to introduce additional "constraints" (i.e., "equations") to our problem to have a chance at computing a unique solution. These additional equations correspond to the boundary conditions for x: The values for b_i for i close to 1 or

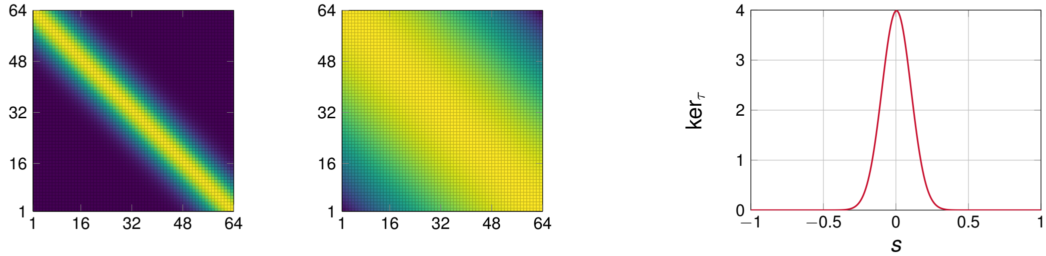

Figure 2: Illustration of the 1D kernel matrix $\mathbf{K}$ for $n = 64$ points. Left: 2D plot of the entries of the kernel matrix $\mathbf{K}$. Middle: 2D plot of the entries of the kernel matrix $\mathbf{K}$ in logarithmic scale. Right: 1D plot of $\ker_\tau(s)$ for $s \in [-1.1]$.

n depend on the values of x_i for $i \in \{-n+2, \ldots, 0\}$ and $i \in \{n+1, \ldots, 2n-1\}$, respectively. These values are associated with $x(s)$ for $s \notin [0,1]$. We select zero boundary conditions, i.e., $x(s) = 0$ for $s \notin [0,1]$, which is in accordance with (3). With this, the above system reduces to a square $(n \times n)$ linear system of the form

$$y_i = h \sum_{j=1}^{n} \kappa_{i-j} x_j, \quad i = 1, \ldots, n,$$

or, more compactly,

$$\mathbf{K}\mathbf{x} = \mathbf{y}, \tag{5}$$

with $\mathbf{y} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$, and $\mathbf{K} = [k_{ij}]_{i,j=1}^{n,n}$, $k_{ij} = h\kappa_{i-j} = h \ker_\tau((i-j)h)$. Consequently, the estimation of the *source* $\mathbf{x}$ boils down to solving the linear system (5) for $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{K}$ as defined above. An implementation for computing the kernel matrix $\mathbf{K}$ for a given mesh size n (and parameter τ) for the one-dimensional case can be found in prbsets/deconv1D/getKernel1D.m. The kernel matrix for a particular choice of n and τ is given in Fig. 2. The script to generate these visualizations is prbsets/deconv1D/scVizKer1D.m.

1.2 Kernel Reconstruction in 1D

1.2.1 Problem Formulation

Find a function from integral

$$y(s) = \int_{-1}^{s} x(t)\,\mathrm{d}t, \quad -1 \le s \le 1. \tag{6}$$

According to the second fundamental theorem of calculus, x can be obtained from y defined on $[-1,1]$ via $x(s) = \mathrm{d}_s y(s)$. If we relate this to (1) and assume the function we convolve the kernel with is given by a step function

$$\chi(s) := \begin{cases} 1 & \text{if } s > 0, \\ 0 & \text{if } s \le 0. \end{cases}$$

Inserting χ into the integral equation (1) yields

$$y(s) = \int_{-\infty}^{\infty} \ker_\tau(s-t)\chi(t)\,\mathrm{d}t = \int_{-\infty}^{\infty} \ker_\tau(t)\chi(s-t)\,\mathrm{d}t = \int_{-\infty}^{s} \ker_\tau(t)\,\mathrm{d}t. \tag{7}$$

If we assume that $\ker_\tau(t) = 0$ for $|t| \ge 1$ and identify ker with x, this equation is equivalent to (6). Consequently, the kernel reconstruction problem involves estimating $x = \ker$ from y in (6).

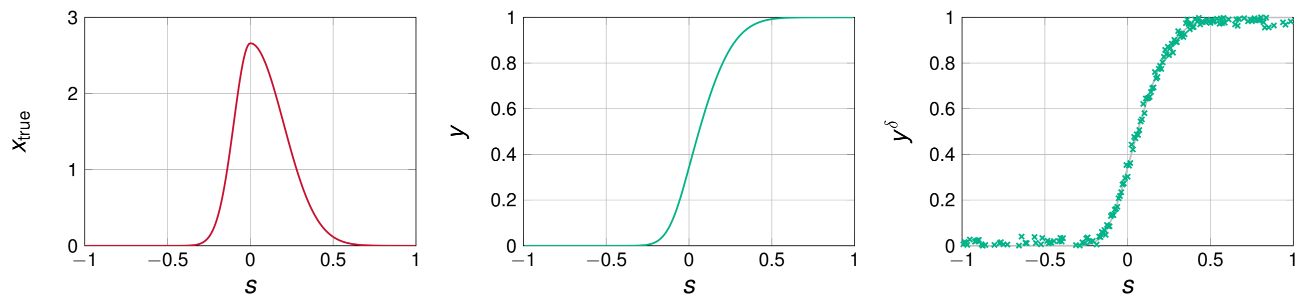

Figure 3: One dimensional kernel reconstruction problem. Left: True kernel $\mathbf{x}_{\text{true}}$. Middle: Integral operator $\mathbf{K}$ applied to $\mathbf{x}_{\text{true}}$. Right: Perturbed data $\mathbf{y}^\delta$. Here, $n = 128$, δ is chosen such that the signal to noise ration γ is equal to 50. The inverse kernel reconstruction problem corresponds to recovering $\mathbf{x} = \mathbf{x}_{\text{true}}$ from $\mathbf{y}^\delta = \mathbf{K}\mathbf{x}_{\text{true}} + \delta\boldsymbol{\eta}$.

1.2.2 Numerical Discretization

Likewise to the source reconstruction problem we again consider a midpoint quadrature. We subdivide $[0, 1]$ into n intervals $[ih, (i+1)h] \subset \mathbb{R}$, $i = 1, \ldots, n$, with $h = 1/n$. We seek to estimate the kernel $x(t)$ at grid points $t_i = ih$, $i = -n+1, \ldots, n-1$. Notice that t_i is the midpoint of the interval $[s_i, s_{i+1}]$ with $s_i = (i - \frac{1}{2})h$. Since $x(-1) = \ker(-1) = 0$, we assume $y(s_{-n+1}) = 0$. We obtain

$$y(s_i) = \int_{-1}^{s_i} x(t)\,\mathrm{d}t = \sum_{j=-n+1}^{i-1} \int_{s_i}^{s_{i+1}} x(t)\,\mathrm{d}t \approx h \sum_{j=-n+1}^{i-1} x(t_j)\,\mathrm{d}t, \quad i = -n+2, \ldots, n.$$

With $\mathbf{y} = \left[y_{-n+2}, \ldots, y_n\right] \in \mathbb{R}^{2n-1}$ and $\mathbf{x} = \left[x_{-n+1}, \ldots, x_{n-1}\right] \in \mathbb{R}^{2n-1}$, we obtain $\mathbf{y} = \mathbf{K}\mathbf{x}$,

$$\mathbf{K} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ 1 & 1 & \cdots & 1 \end{bmatrix} \in \mathbb{R}^{2n-1,2n-1}.$$

Consequently, the estimation of the *kernel* $\mathbf{x}$ boils down to solving the linear system $\mathbf{y} = \mathbf{K}\mathbf{x}$ for $\mathbf{x}$ given $\mathbf{y}$ and $\mathbf{K}$ as defined above. We illustrate an exemplary kernel and the associated data in Fig. 3. The script to generate this data is prbsets/deconv1D/scVizRecoData1D.m. The function to compute the kernel $\mathbf{x}_{\text{true}}$ is prbsets/deconv1D/getRecoKernel1D.m. The function to compute the matrix $\mathbf{K}$ is implemented in prbsets/deconv1D/getRecoMat1D.m.

1.3 Additive Noise Model

To compute the observation $\mathbf{y} := \mathbf{y}^\delta \in \mathbb{R}^n$, we apply $\mathbf{K} \in \mathbb{R}^{n,n}$ to $\mathbf{x}_{\text{true}} \in \mathbb{R}^n$ and perturb the resulting $\mathbf{y} \in \mathbb{R}^n$ by noise $\delta\boldsymbol{\eta} \in \mathbb{R}^n$, i.e., $\mathbf{y}^\delta = \mathbf{K}\mathbf{x}_{\text{true}} + \delta\boldsymbol{\eta}$. The noise level δ will be selected such that the signal-to-noise ratio $\|\mathbf{K}\mathbf{x}_{\text{true}}\|/\sqrt{n\delta^2}$ is equal to a constant γ. An implementation of this additive noise model can be found in core/addNoise.m.

2 Assignments

1. First, we explore the conditioning of the problem (5) as a function of the parameterization of the kernel as well as the discretization accuracy. **Hint:** *To compute the condition number of a matrix, you can use Matlab's* `cond` *command. The script to work on this assignment is prbsets/deconv1D/scCondK1D.m.*

   a) Compute the condition number of $\mathbf{K}$ as a function of τ and plot it (semi-logarithmic plot). Set n to 32. Select $\tau = \gamma(1\mathrm{e}-2)$, where γ represents integers from 1 to 10. What are your observations?

   b) Compute the condition number of $\mathbf{K}$ as a function of n and plot it (semi-logarithmic plot). Set $\tau = 1\mathrm{e}-2$. Select $n \in \{16, 32, 64, 128, 256\}$. What are your observations?

2. The first approach we are going to consider to compute a solution to (5) is based on the truncated singular value decomposition (**TSVD**). That is, we are going to consider the decomposition $\mathbf{K} = \mathbf{U}\mathbf{S}\mathbf{V}^\mathsf{T}$, $\mathbf{U} = \begin{bmatrix}\mathbf{u}_1, \dots, \mathbf{u}_m\end{bmatrix} \in \mathbb{R}^{m,m}$, $\mathbf{V} = \begin{bmatrix}\mathbf{v}_1, \dots, \mathbf{v}_n\end{bmatrix} \in \mathbb{R}^{n,n}$, $\mathbf{S} = \mathrm{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m,n}$, $p = \min\{n, m\}$, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. Under the assumption that $\mathbf{K} \succeq 0$, we have that $\sigma_i \geq 0$, $i = 1, \dots, p$, the singular values coincide with the eigenvalues of $\mathbf{K}$, and $\mathbf{U} = \mathbf{V}$. Moreover, the column $\mathbf{u}_j \in \mathbb{R}^m$ for an orthonormal basis with $\mathbf{U}^\mathsf{T}\mathbf{U} = \mathbf{I}$, i.e., $\mathbf{U}^\mathsf{T} = \mathbf{U}^{-1}$. The truncated SVD of $\mathbf{K}$ is given by $\mathbf{K} = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^\mathsf{T}$, with $\mathbf{U}_r \in \mathbb{R}^{m,r}$, $\mathbf{S} \in \mathbb{R}^{r,r}$, $\mathbf{V}_r \in \mathbb{R}^{n,r}$.

   a) Compute the SVD of the matrix $\mathbf{K}$. Plot the singular vectors (semi-logarithmic plot; threshold the values at $1\mathrm{e}-14$, i.e., all values below $1\mathrm{e}-14$ are set to $1\mathrm{e}-14$). Plot the singular vectors $\mathbf{v}_j$ (i.e., the columns of the matrix $\mathbf{V}$) corresponding to the singular vectors σ_j, $j \in \{1, 5, 10, 20\}$. What are your observations; i.e., how do the singular vectors $\mathbf{v}_j$ behave as the singular values σ_j decrease? **Hint:** *To compute the SVD you can use Matlab's* `svd` *function. The script for this assignment is prbsets/deconv1D/scCompKerSVD1D.m.*

   b) Compute the TSVD for the target ranks $r \in \{5, 10, 50\}$. To compute the rank-r approximations, you can simply compute the full SVD and select the first r columns of the matrices $\mathbf{U}$ and $\mathbf{V}$, and the $r \times r$ submatrix of the diagonal matrix $\mathbf{S}$. Compute $\mathbf{K}_r = \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r$, $\mathbf{U}_r \in \mathbb{R}^{m,r}$, $\mathbf{S} \in \mathbb{R}^{r,r}$, $\mathbf{V}_r \in \mathbb{R}^{n,r}$. **Hint:** *To compute the SVD you can use Matlab's* `svd` *function. The script for this assignment is prbsets/deconv1D/scTSVDK1D.m. To implemented the TSVD, you can use the template core/tSVD.m.*

      i. Visualize the matrices $\mathbf{K}_r$, $r \in \{5, 10, 50\}$ and $\mathbf{K}$. **Hint:** *You can use Matlab's* `imagesc` *to visualize these matrices.*

      ii. Visualize the point-wise absolute value of the residual matrix $\mathbf{R}_r = |\mathbf{K}_r - \mathbf{K}|$. **Hint:** *You can use Matlab's* `imagesc` *to visualize the matrices $\mathbf{R}_r$.*

      iii. Compute the relative error $e_r = \|\mathbf{K} - \mathbf{K}_r\|_2/\|\mathbf{K}\|_2$. How does the error behave?

   c) Consider the cases $\tau \in \{5\mathrm{e}-3, 2\mathrm{e}-2\}$. Set $n = 256$. Select the noise level δ such that the signal-to-noise ratio $\|\mathbf{K}\mathbf{x}_{\text{true}}\|/\sqrt{n\delta^2}$ is equal to a constant $\gamma = 50$. Use a TSVD to compute the regularized solution $\mathbf{x}_\alpha = \mathbf{R}_\alpha\mathbf{y}^\delta$, where

$$\mathbf{R}_\alpha\mathbf{y}^\delta = \sum_{i=1}^{n} w(\sigma_i)\langle\mathbf{u}_i, \mathbf{y}^\delta\rangle\mathbf{u}_i \quad \text{with} \quad w(\sigma_i) = \begin{cases} \sigma_i^{-1} & \text{for } \sigma_i > \alpha, \\ 0 & \text{otherwise.} \end{cases}$$

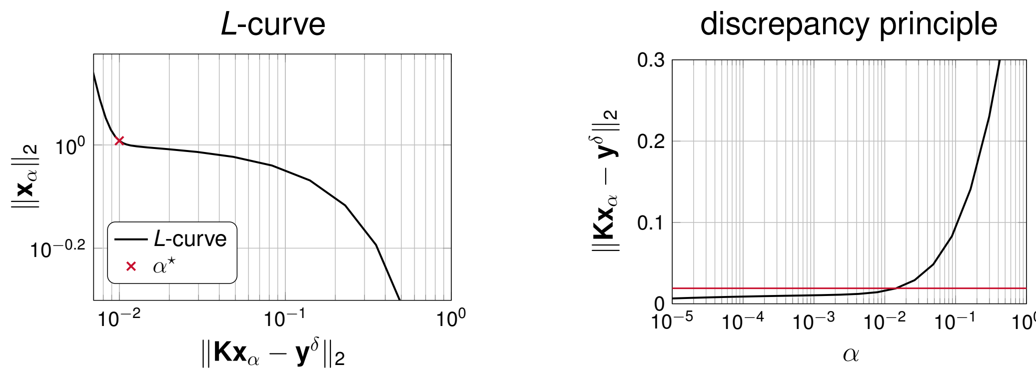

Figure 4: Choosing the regularization parameter α: The red cross on the L-curve (left plot), which corresponds to the point with largest curvature, yields the optimal regularization parameter according to the L-curve criterion. For the discrepancy criterion (right plot), the optimal parameter corresponds to the intersection of the data misfit curve with the red line indicating the noise level.

As indicated above, we define the TSVD in terms of a threshold α instead of a target rank r. Consider the thresholds $\alpha \in \{1\text{e}-3, 1\text{e}-2, 1\text{e}-1, 2\text{e}-1, 5\text{e}-1\}$ to compute the regularized solutions $\mathbf{x}_\alpha$. Plot the solutions $\mathbf{x}_\alpha$ and compare them to the true solution $\mathbf{x}_{\text{true}}$. **Hint:** *One way of computing the truncated SVD is to compute the SVD and identify the index j for which the singular value $\sigma_j < \alpha$. This can be accomplished by using Matlab's `find` function. The pseudoinverse can then be computed by inverting the truncated SVD matrix, i.e., $\mathbf{R}_\alpha = (\mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^\mathsf{T})^{-1}$, where $r = j$. The script for this assignment is prbsets/deconv1D/scDeconvTSVD1D.m. To implemented the TSVD based on thresholding, you can use the template core/tSVDTH.m.*

3. The second approach we are going to consider for computing a solution to (5) is based on a Tikhonov regularization scheme. Set $n = 256$ and $\tau = 5\text{e}-3$. Select the noise level δ such that the signal-to-noise ratio $\|\mathbf{K}\mathbf{x}_{\text{true}}\|/\sqrt{n\delta^2}$ is equal to a constant $\gamma = 50$. Consider the variational optimization problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \frac{1}{2}\|\mathbf{K}\mathbf{x} - \mathbf{y}^\delta\|_2^2 + \frac{\alpha}{2}\|\mathbf{x}\|_2^2.$$

The associated first order optimality conditions are given by

$$\mathbf{K}^\mathsf{T}(\mathbf{K}\mathbf{x}^\star - \mathbf{y}^\delta) + \alpha\mathbf{x}^\star = \mathbf{0}. \tag{8}$$

a) Compute the solution $\mathbf{x}_\alpha$ by solving the (linear) optimality system (8) for $\alpha \in \{1\text{e}-12, 1\text{e}-3, 1\text{e}-1\}$. Plot the solution $\mathbf{x}_\alpha$ and the true solution $\mathbf{x}_{\text{true}}$. **Hint:** *To compute the solution of (8), you can use Matlab's backslash operation. A script to help with the implementation of this direct solver is prbsets/deconv1D/scDeconvTRegDir1D.m.*

b) Determine the (approximate) optimal value $\alpha_{\text{opt}} > 0$ of the regularization parameter $\alpha > 0$ for the Tikhonov regularization using the L-curve criterion. To do so, one needs to compute the solution of the inverse problem (here, this means solving the optimality system (8)) for varying regularization parameters $\alpha_i > 0$. Use the implementation from part (a) to do so. Select $\alpha_i \in [1\text{e}-5, 1]$. Select 20 different values. To determine the optimal regularization parameter, plot the norm of the computed solution $\|\mathbf{x}_\alpha\|_2$ versus the norm of the residual $\|\mathbf{K}\mathbf{x}_\alpha - \mathbf{y}^\delta\|_2$ using a "log-log plot" (i.e., using a two-dimensional graph of numerical data that uses a logarithmic scale on both the horizontal and vertical axes). The optimal regularization parameter α_{opt} is located in the corner of the resulting L-shaped curve. This is illustrated in Fig. 4. Compute the solution using the determined regularization parameter α_{opt} and compare it (visually) to the true solution $\mathbf{x}_{\text{true}}$. **Hint:** *A script to help with the implementation of the search for an optimal regularization parameter $\alpha_{opt} > 0$ based on the L-curve criterion is prbsets/deconv1D/scKerRecoTRegLC1D.m. A template for implementing a plot for the L-curve is core/evalLCurve.m.*

c) Determine the (approximate) optimal value $\alpha_{\text{opt}} > 0$ of the regularization parameter $\alpha > 0$ for the Tikhonov regularization using Morozov's discrepancy principle, i.e., find the largest value of α such that

$$\|\mathbf{K}\mathbf{x}_\alpha - \mathbf{y}^\delta\|_2 \leq \mu,$$

where $\mu = \|\delta\boldsymbol{\eta}\|_2$ and $\mathbf{x}_\alpha$ is the solution of the Tikhonov-regularized inverse problem with regularization parameter α. To search for an optimal α, select $\alpha_i \in [1\text{e}-5, 1]$. Select 20 different values. **Hint:** *A script to help with the implementation of the search for an optimal regularization parameter $\alpha_{opt} > 0$ based on the discrepancy principle is prbsets/deconv1D/scDeconvTRegMDP1D.m. A template for implementing the search for an optimal α using Morozov's discrepancy principle is core/evalDisPrinc.m.*

d) Plot the relative error in the reconstruction, $e_\alpha = \|\mathbf{x}_{\text{true}} - \mathbf{x}_\alpha\|_2 / \|\mathbf{x}_{\text{true}}\|_2$ as a function of $\alpha > 0$, where $\mathbf{x}_\alpha \in \mathbb{R}^n$ is the Tikhonov regularized solution. Which value of α_i (approximately) minimizes this error? Compare your observations to the optimal values of α obtained for the L-curve method, Morozov's principle, and the truncated SVD. A template for the computation of this error can be found in prbsets/deconv1D/scDeconvTRegERR1D.m.

4. Next, we consider the kernel-reconstruction problem (6). Set $n = 256$. Select the noise level δ such that the signal-to-noise ratio $\|\mathbf{K}\mathbf{x}_{\text{true}}\|/\sqrt{n\delta^2}$ is equal to a constant $\gamma = 50$.

   a) Likewise to the deconvolution problem, use a TSVD to compute the regularized solution $\mathbf{x}_\alpha = \mathbf{R}_\alpha \mathbf{y}^\delta$. Consider the thresholds $\alpha \in \{1\text{e}-3, 1\text{e}-2, 5\text{e}-2, 8\text{e}-2, 1\text{e}-1\}$ to compute the regularized solutions $\mathbf{x}_\alpha$. Plot the solutions $\mathbf{x}_\alpha$ and compare them to the true solution $\mathbf{x}_{\text{true}}$. **Hint:** *A script to help you with this implementation is prbsets/deconv1D/scKerRecoTSVD1D.m.*

   b) Determine the (approximate) optimal value $\alpha_{\text{opt}} > 0$ of the regularization parameter $\alpha > 0$ for the Tikhonov regularization using the L-curve criterion. Compute the solution using the determined regularization parameter α_{opt} and compare it (visually) to the true solution $\mathbf{x}_{\text{true}}$. **Hint:** *A script to help with the implementation of the search for an optimal regularization parameter $\alpha_{opt} > 0$ based on the L-curve criterion is prbsets/deconv1D/scKerRecoTRegLC1D.m. A template for implementing a plot for the L-curve is core/evalLCurve.m.*