CWI Autumn School 2024: Numerical Methods for Bayesian Inverse Problems

Exercise Sheet 1

Exercise 1: Deconvolution. Let us consider a deconvolution problem and solve it numerically.

Fix $N \in \mathbb{N}$ and let X be the space of periodic sequences in $\mathbb{C}$, i.e.

$$X = \{(x_j)_{j\in\mathbb{Z}} \in \mathbb{C}^{\mathbb{Z}} \,:\, x_j = x_{j+N} \;\forall j \in \mathbb{Z}\}.$$

(This could arise, e.g., after discretising a continuous deconvolution problem on a uniform grid.) For $K \in X$ introduce the discrete convolution operator $T_K : X \to X$

$$(T_K f)_k = \sum_{j=0}^{N-1} K_{k-j} f_j \qquad \forall k \in \mathbb{Z}.$$

Furthermore denote by $\mathcal{F}_N : X \to X$ the discrete Fourier transform, i.e. for $f \in X$

$$(\mathcal{F}_N f)_k = \sum_{j=0}^{N-1} \exp\Big(- 2\pi \mathrm{i} \frac{j \cdot k}{N} \Big) f_j \qquad \forall k \in \mathbb{Z}.$$

(a) Show that $\mathcal{F}_N^{-1} : X \to X$ is given through

$$(\mathcal{F}_N^{-1} f)_k = \frac{1}{N} \sum_{j=0}^{N-1} \exp\Big(2\pi \mathrm{i} \frac{j \cdot k}{N} \Big) f_j.$$

Solution: It holds

$$\begin{aligned}
(\mathcal{F}_N^{-1}\mathcal{F}_N f)_k &= \frac{1}{N} \sum_{j=0}^{N-1} \exp\left(2\pi \mathrm{i} \frac{j \cdot k}{N} \right) (\mathcal{F}_N f)_j \\
&= \frac{1}{N} \sum_{j=0}^{N-1} \exp\left(2\pi \mathrm{i} \frac{j \cdot k}{N} \right) \sum_{l=0}^{N-1} \exp\left(-2\pi \mathrm{i} \frac{l \cdot j}{N} \right) f_l \\
&= \frac{1}{N} \sum_{j=0}^{N-1} \sum_{l=0}^{N-1} \exp\left(2\pi \mathrm{i} \frac{j \cdot (k-l)}{N} \right) f_l.
\end{aligned}$$

Furthermore

$$\sum_{j=0}^{N-1} \exp\left(2\pi \mathrm{i} \frac{j \cdot (k-l)}{N} \right) = \begin{cases} N & \text{if } l = k \\ \frac{1-\exp(2\pi \mathrm{i}(k-l))}{1-\exp(2\pi \mathrm{i}\frac{k-l}{N})} = 0 & \text{if } l \neq k. \end{cases}$$

Hence $\mathcal{F}_N^{-1}\mathcal{F}_N f = f$. Similarly $\mathcal{F}_N\mathcal{F}_N^{-1} f = f$, which implies the claim.

(b) Show that

$$(\mathcal{F}_N T_K f)_k = (\mathcal{F}_N K)_k (\mathcal{F}_N f)_k \qquad \forall k \in \mathbb{Z}.$$

Solution: It holds

$$
\begin{aligned}
(T_K f)_k &= \sum_{j=0}^{N-1} K_{k-j} f_j \\
&= \sum_{j=0}^{N-1} \left(\frac{1}{N} \sum_{l_1=0}^{N-1} (\mathcal{F}_N K)_{l_1} \exp\left(2\pi \mathrm{i} \frac{l_1 \cdot (k-j)}{N}\right)\right) \left(\frac{1}{N} \sum_{l_2=0}^{N-1} (\mathcal{F}_N f)_{l_2} \exp\left(2\pi \mathrm{i} \frac{l_2 \cdot j}{N}\right)\right) \\
&= \frac{1}{N^2} \sum_{l_1, l_2 = 0}^{N-1} (\mathcal{F}_N K)_{l_1} (\mathcal{F}_N f)_{l_2} \exp\left(2\pi \mathrm{i} \frac{l_1 \cdot k}{N}\right) \sum_{j=0}^{N-1} \exp\left(2\pi \mathrm{i} \frac{(l_2 - l_1) \cdot j}{N}\right).
\end{aligned}
$$

As before, the last sum is N if $l_1 = l_2$ and 0 otherwise, hence

$$
(T_K f)_k = \frac{1}{N} \sum_{l=0}^{N-1} (\mathcal{F}_N K)_l (\mathcal{F}_N f)_l \exp\left(2\pi \mathrm{i} \frac{l \cdot k}{N}\right) = \mathcal{F}_N^{-1}((\mathcal{F}_N K) \cdot (\mathcal{F}_N f))_k,
$$

where we use the notation $h \cdot g = (h_i g_i)_i$ for vectors. Applying $\mathcal{F}_N$ to both sides yields the claim.

(c) Show that if $(\mathcal{F}_N K)_k \neq 0$ for all $k \in \{0, \dots, N-1\}$ then T_K is invertible and

$$
(T_K^{-1} f)_k = \sum_{j=0}^{N-1} L_{k-j} f_j \qquad \forall k \in \mathbb{Z},
$$

where $L \in X$ satisfies $(\mathcal{F}_N L)_k = 1/(\mathcal{F}_N K)_k$ for $k \in \{0, \dots, N-1\}$.

HINT: Consider $e_n := \sqrt{N} \mathcal{F}_N^{-1} b_n$, where $b_n = (\delta_{jn})_{j=0}^{N-1}$ and find a spectral decomposition of T_K (with a suitable inner product on X).

Solution: It holds

$$
(e_n)_k = \frac{1}{\sqrt{N}} \exp\left(2\pi \mathrm{i} \frac{n \cdot k}{N}\right).
$$

With the inner product $\langle a, b \rangle = \sum_{j=0}^{n} a_j \overline{b_j}$, the sequence $(e_n)_{n=0}^{N-1}$ forms an orthonormal basis of X. By **??**

$$
\begin{aligned}
T_K e_n &= \mathcal{F}_N^{-1}((\mathcal{F}_N K) \cdot (\mathcal{F}_N \sqrt{N} \mathcal{F}_N^{-1} b_n)) \\
&= \sqrt{N} \mathcal{F}_N^{-1}((\mathcal{F}_N K)_n b_n) \\
&= (\mathcal{F}_N K)_n \sqrt{N} \mathcal{F}_N^{-1} b_n = (\mathcal{F}_N K)_n e_n.
\end{aligned}
$$

Thus e_n is an eigenvector of T_K with eigenvalue $(\mathcal{F}_N K)_n$. Therefore T_K is bijective iff $(\mathcal{F}_N K)_n \neq 0$ for all $n = 0, \dots, N-1$, and T_K has the spectral decomposition

$$
T_K f = \sum_{n=0}^{N-1} (\mathcal{F}_N K)_n \langle f, e_n \rangle e_n.
$$

Furthermore, using the notation $\frac{1}{\mathcal{F}_N K} = (\frac{1}{(\mathcal{F}_N K)_l})_{l=0}^{N-1}$, we have

$$\begin{aligned}
(T_K^{-1} f)_k &= \sum_{n=0}^{N-1} \frac{1}{(\mathcal{F}_N K)_n} \langle f, e_n \rangle (e_n)_k \\
&= \sum_{n=0}^{N-1} \frac{1}{(\mathcal{F}_N K)_n} \sum_{j=0}^{N-1} f_j \frac{1}{\sqrt{N}} \exp\left(-2\pi \mathrm{i} \frac{n \cdot j}{N}\right) \frac{1}{\sqrt{N}} \exp\left(2\pi \mathrm{i} \frac{n \cdot k}{N}\right) \\
&= \sum_{j=0}^{N-1} f_j \underbrace{\left(\frac{1}{N} \sum_{n=0}^{N-1} \frac{1}{(\mathcal{F}_N K)_n} \exp\left(2\pi \mathrm{i} \frac{n(k-j)}{N}\right)\right)}_{=(\mathcal{F}_N^{-1}(\frac{1}{\mathcal{F}_N K}))_{k-j}}, .
\end{aligned}$$

(d) Implement a function `convolution(K,f)` that takes $(f_k)_{k=0}^{N-1}$ and returns $((T_k f)_k)_{k=0}^{N-1}$. Use the fast Fourier transform (FFT), e.g. the function `numpy.fft.fft` and its inverse `numpy.fft.ifft` in Python (or similar functions available in Matlab).

(e) To regularise the deconvolution problem we use truncation (or hard-thresholding), i.e. we cut off the high frequencies by replacing L with L^α such that $(\mathcal{F}_N L^\alpha)_k = g_\alpha((\mathcal{F}_N K)_k)$ with $g_\alpha(x) = 1/x$ if $|x| > \alpha$ and $g_\alpha(x) = 0$ otherwise.

The regularised inverse operator $T_{K,\alpha}^\dagger$ (replacing the ill-posed T_K^{-1}) is then defined as

$$(T_{K,\alpha}^\dagger f)_k = \sum_{j=0}^{N-1} L_{k-j}^\alpha f_j \qquad \forall k \in \mathbb{Z},$$

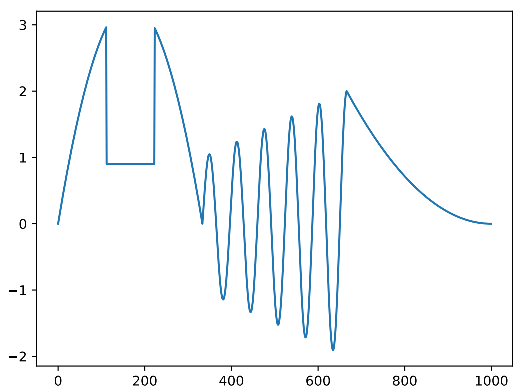
Implement a function `deconvolution(K,g,a)` that takes $(K_k)_{k=0}^{N-1}$, $(g_k)_{k=0}^{N-1}$ and α and returns $T_\alpha^\dagger g$.

(g) Test your code with the data $(f_k)_{k=0}^{N-1}$ provided in `signal.txt` (on Moodle) and use the following Gaussian kernel K:

```
import numpy as np
s = 100
gaussian = lambda x, s: np.exp(-s*x**2.)
K = gaussian(np.linspace(-1./2,1./2,N),s)
K = N*np.fft.ifftshift(K)/np.sum(K)
```

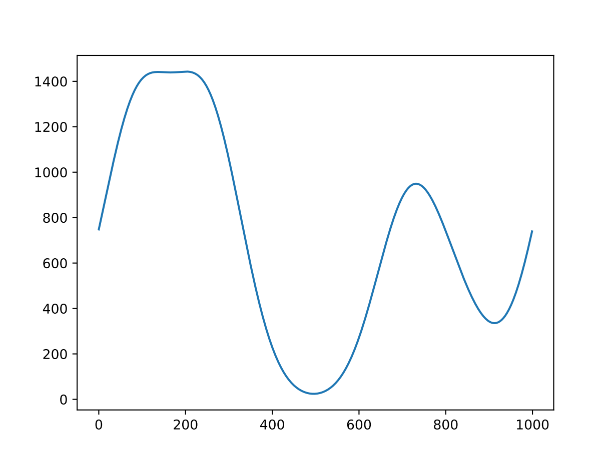
Plot f, $g = T_k f$ and the reconstructed signal $T_\alpha^\dagger g$ for $\alpha = 10^{-11}$.

Signal:



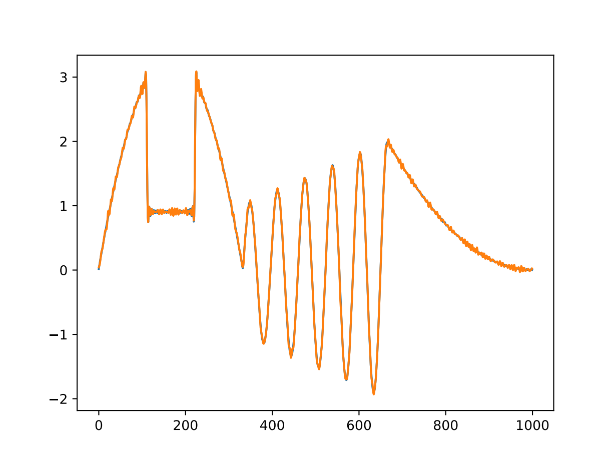
Solution:

Convoluted signal:



(h) Add noise to the signal via $g^\delta = g + \delta W$ with W being Gaussian noise, for example in Python `W=np.random.randn(N)`. Plot $T_\alpha^\dagger g^\delta$ for $\delta \in \{10^{-10}, 10^{-12}\}$. Explain what you observe.

Solution: Reconstruced signal for $\delta = 1e-12$:



Reconstruced signal for $\delta = 1e-10$: