Group Exercise: 5.13

Homework (10 Points): 5.10(b), (c), (d)

Due Date: Monday , December 13, 2021, 11am.

Material: https://collaborating.tuhh.de/e-10/teaching/mathematical-image-processing/mathimproc-wt2021/-/archive/master/mathimproc-wt2021-master.zip?path=ex09

For tasks that do not require source code but a handwritten solution, add a .pdf-scan of your handwritten solution to your repository.

5.11 Exercises

Exercise 5.1 (Normed Vector Spaces of Sequences)
Recall Definition 5.1 and let $\ell^p(\mathbb{Z})$ denote the space of all complex valued sequences $(x_n)_{n\in\mathbb{Z}}$ such that

$$\|(x_n)\|_p := \left(\sum_{n\in\mathbb{Z}} |x_n|^p\right)^{\frac{1}{p}} < \infty. \tag{5.29}$$

A sequence $(x_n)_{n\in\mathbb{Z}}$ can be interpreted as an two-sided infinite vector of the form

$$(\dots, x_{-3}, x_{-2}, x_{-1}, x_0, x_1, x_2, x_3, \dots)\,. \tag{5.30}$$

We will sometimes write $x(k)$ instead of x_k in order to denote the kth element of the sequence x.

(a) Consider the sequences e_l and ψ_q which are defined via

$$e_l(k) := \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l, \end{cases} \qquad \psi_q(k) := \begin{cases} q^k & \text{if } k \in \mathbb{N}_0, \\ 0 & \text{if } k \notin \mathbb{N}_0, \end{cases} \qquad k \in \mathbb{Z},$$

for some $l \in \mathbb{Z}$ and some $q \in \mathbb{C}$ with $|q| < 1$.

(i) Let $l = 2$ and give a description of the sequences e_2 and ψ_q as infinite vectors similar to (5.30).

(ii) Calculate the norms $\|e_2\|_p$ and $\|\psi_{0.5}\|_p$ for all $p \in [1, \infty)$.
Hint: Recall the geometric sum formula, i.e. for $\lambda \in \mathbb{C}, \lambda \neq 1$ we have

$$\sum_{n=0}^{N} \lambda^n = \underline{\qquad\qquad\qquad}\,.$$

(iii) Give an example of a sequence $x \notin \ell^1(\mathbb{Z})$ and of another sequence $y \notin \ell^2(\mathbb{Z})$.

(b) Convince yourself (no proof or calculation needed) that $\ell^1(\mathbb{Z})$ is a complex *vector subspace* (of which vector space?) and that (5.29) defines a *norm* on $\ell^1(\mathbb{Z})$. Do you know further properties of $\ell^1(\mathbb{Z})$?

Exercise 5.2 (Convolution on $\ell^1(\mathbb{Z})$)
In the Definition 5.2, you saw that it is possible to define an operation

$$*\colon \ell^1(\mathbb{Z}) \times \ell^1(\mathbb{Z}) \to \ell^1(\mathbb{Z}), \quad (x, y) \mapsto x * y, \tag{5.31}$$

where

$$(x * y)(n) := \sum_{k\in\mathbb{Z}} x(k)\, y(n-k)$$

for all $n \in \mathbb{Z}$. In particular, $x * y$ is again a sequence.

(a) Consider the sequences e_l and ψ_q which are defined via

$$e_l(k) \coloneqq \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l, \end{cases} \qquad \psi_q(k) \coloneqq \begin{cases} q^k & \text{if } k \in \mathbb{N}_0, \\ 0 & \text{if } k \notin \mathbb{N}_0, \end{cases} \qquad k \in \mathbb{Z},$$

for some $l \in \mathbb{Z}$ and some $q \in \mathbb{C}$ with $|q| < 1$. Calculate $e_{42} * \psi_{\frac{\mathrm{i}}{42}}$ and $\psi_{0.5\mathrm{e}^{\pi\mathrm{i}}} * \psi_{0.5\mathrm{e}^{\pi\mathrm{i}}}$.

(b) The operation $*$ can be seen as a *product* on the space $\ell^1(\mathbb{Z})$. In the following, we will prove that the product (5.31) is *well-defined* and compatible with the other algebraic operations and the topology on $\ell^1(\mathbb{Z})$. Prove the following properties of the mapping $*$:

(i) For all $x, y \in \ell^1(\mathbb{Z})$ we have that $\|x * y\|_1 \leq \|x\|_1 \|y\|_1$.
This shows that we have $x * y \in \ell^1(\mathbb{Z})$.

(ii) Give a sufficient condition for $x, y \in \ell^1(\mathbb{Z})$ such that we have $\|x * y\|_1 = \|x\|_1 \|y\|_1$.

(iii) The mapping $(x, y) \mapsto x * y$ is *bilinear*.
This implies that $\ell^1(\mathbb{Z})$ together with $*$ defines an *algebra* (over the field $\mathbb{C}$).

(iv) The operation $*$ is *commutative*. Also give an example of an algebra that is not commutative, i.e. specify a vector space and a bilinear product.

(v) There exists an element $e \in \ell^1(\mathbb{Z})$ with the following property: For all $x \in \ell^1(\mathbb{Z})$, we have that $e * x = x = x * e$. This element is called *unit element* and therefore $\ell^1(\mathbb{Z})$ is called a *unital* algebra. Give an example of a *non-unital* associative algebra (without a proof). Provide the same level of information as in (iv).

(vi) The operation $*$ is compatible with the norm $\|\cdot\|_1$ in the following way: For $x, y \in \ell^1(\mathbb{Z})$ their convolution $x * y$ is an element of $\ell^1(\mathbb{Z})$ and

$$\|x * y\|_1 \leq \|x\|_1 \|y\|_1. \tag{5.32}$$

This shows that $*$ is a *continuous* operation on $\ell^1(\mathbb{Z})$, therefore $\ell^1(\mathbb{Z})$ is also called a *Banach algebra*. Under which additional assumptions on $x, y \in \ell^1(\mathbb{Z})$ does equality hold in (5.32).

Exercise 5.3 (Convolution of finite sequences) (a) For subsets $M \subseteq \mathbb{Z}$, we may define the spaces $\ell^1(M)$ with a norm analogous to (5.29). In order to define the convolution, we identify elements of $\ell^1(M)$ with elements of $\ell^1(\mathbb{Z})$ via *zero padding*, i.e. if $x \in \ell^1(M)$, then $x^{(0)} \in \ell^1(\mathbb{Z})$ with

$$x^{(0)}(n) \coloneqq \begin{cases} x(n) & \text{if } n \in M, \\ 0 & \text{if } n \in \mathbb{Z} \setminus M. \end{cases}$$

This allows us to extend the definition of (5.31) to spaces of finite sequences via

$$*\colon \ell^1(M_1) \times \ell^1(M_2) \to \ell^1(\mathbb{Z}), \quad (x, y) \mapsto x^{(0)} * y^{(0)},$$

where $M_1, M_2 \subseteq \mathbb{Z}$.

(i) Let $M_1 = \{0, 1, 2\}$ and $x = \frac{1}{3}(1, 1, 1)$ and $y \in \ell^1(\mathbb{Z})$. Calculate $(x * y)(k)$ for $k \in \mathbb{Z}$.

(ii) Let $M_1 = \{1, 2, 3\}$ and $x = \frac{1}{3}(1, 1, 1)$ and $y \in \ell^1(\mathbb{Z})$. Calculate $(x * y)(k)$ for $k \in \mathbb{Z}$.

(iii) Let $M_1 = M_2 = M = \{1, 2, 3\}$ and $x, y \in \ell^1(M)$, i.e. $x = (x_1, x_2, x_3)$ and $y = (y_1, y_2, y_3)$. The following table may be helpful:

l	$\cdots$	-4	-3	-2	-1	0	1	2	3	4	5	$\cdots$
$x(l)$	$\cdots$	0	0	0	0	0	x_1	x_2	x_3	0	0	$\cdots$
$y(0-l)$	$\cdots$	0	y_3	y_2	y_1	0	0	0	0	0	0	$\cdots$
$y(1-l)$	$\cdots$	0	0	y_3	y_2	y_1	0	0	0	0	0	$\cdots$
$y(2-l)$	$\cdots$	0	0	0	y_3	y_2	y_1	0	0	0	0	$\cdots$

Calculate $(x * y)(k)$ for $k = 0, 1, \ldots, 5$.

(iv) In part (iii), we saw that for index sets starting with 1, there is an unnatural shift as the resulting sequence starts from 2 instead of 1. Show that the following definition of convolution fixes this problem for sequences that start from 1:

$$\star\colon \ell^1(\mathbb{N}) \times \ell^1(\mathbb{N}) \to \ell^1(\mathbb{N}), \quad (x \star y)(n) \coloneqq \sum_{k=1}^{\infty} x^{(0)}(k)\, y^{(0)}(n-k+1)\,. \tag{5.33}$$

Show that for $n \in \mathbb{N}$, we have $(x \star y)(n) = (x * y)(n+1)$.

(v) Now let $M_1 = \{1, \ldots, l_1\}$ and $M_2 = \{1, \ldots, l_2\}$ be two index sets and $x \in \ell^1(M_1)$ and $y \in \ell^1(M_2)$. Specify a subset $M_3 \subseteq \mathbb{N}$ such that (after removing unnecessary zeros) you have $x \star y \in \ell^1(M_3)$.

(b) Another way to define the convolution of two finite vectors is the so called *circular* convolution. Assume that $M_1 = M_2 = M = \{1, \ldots, l\}$. We define

$$\otimes\colon \ell^1(M) \times \ell^1(M) \to \ell^1(M), \quad (x \otimes y)(n) \coloneqq \sum_{k=1}^{l} x(k)\, \widetilde{y}(n+1-k) \tag{5.34}$$

where $\widetilde{y}$ is a periodic continuation of y as shown in the following table for the case $l = 3$:

$\cdots$	$\widetilde{y}_{-2}$	$\widetilde{y}_{-1}$	$\widetilde{y}_0$	$\widetilde{y}_1$	$\widetilde{y}_2$	$\widetilde{y}_3$	$\widetilde{y}_4$	$\widetilde{y}_5$	$\widetilde{y}_6$	$\cdots$
$\cdots$	y_1	y_2	y_3	y_1	y_2	y_3	y_1	y_2	y_3	$\cdots$

(i) For $l \geq 3$, specify $x = (x_1, x_2, \ldots, x_l)$ such that

$$(x \otimes y)(n) = \frac{1}{3}(\widetilde{y}(n-1) + \widetilde{y}(n) + \widetilde{y}(n+1)) \quad \text{for } n = 1, \ldots, l.$$

(ii) How many zeros do you have to append to y such that $(x \star y)(n) = (x \otimes y)(n)$ for all suitable n?

(c) Implement a function

```
function [z] = compConvolution(x,y,mode)
```

where `x` and `y` are vectors and `mode` is a string containing the boundary condition. Assume that, with the definitions of part (a), for the arguments `x` and `y`, you always have that $M_1 = \{1, 2, \ldots, \texttt{length(x)}\}$ and $M_2 = \{1, 2, \ldots, \texttt{length(y)}\}$.

Depending on the value of `mode`, your function should calculate the convolution of `x` and `y` as in equation (5.33) or equation (5.34).

`mode='zero'`: Suppose that outside of M_2 the vector `y` is extended by zero.

`mode='circular'`: Suppose that $M_1 = M_2$ and outside of M_2 the vector `y` is continued periodically. The OCTAVE function `mod` may be helpful for this implementation.

Exercise 5.4 (Continuous Fourier Transform)
In the lecture, you learned that for functions $f \in \mathrm{L}^1(\mathbb{R})$, the Fourier transform $\hat{f}$ is defined via

$$\hat{f}(z) \coloneqq \frac{1}{(2\pi)^{\frac{1}{2}}} \int_{\mathbb{R}} f(x) \mathrm{e}^{-\mathrm{i}zx} \, \mathrm{d}x, \quad z \in \mathbb{R}.$$

(a) Calculate the Fourier transform of $f \colon \mathbb{R} \to \mathbb{R}$ which is defined via

$$f(x) = \frac{1}{1 + x^2} \, .$$

Hint: Start by calculating the Fourier transform of $\mathrm{e}^{-\lambda|x|}$ *for* $\lambda > 0$.

(b) Show that for the family of functions $(f_a)_{a>0}$ which is defined via

$$f_a(x) = \frac{1}{a\pi} \frac{1}{1 + \frac{x^2}{a^2}}$$

the following *semigroup property* with respect to convolution holds

$$f_a * f_b = f_{a+b} \quad \text{for } a, b > 0.$$

(c) Let $\lambda > 0$. Determine the Fourier transform of the function

$$g(x) = \begin{cases} x\mathrm{e}^{-\lambda x} & \text{for } x \geq 0, \\ 0 & \text{else.} \end{cases}$$

(d) Use *Parseval's theorem* to derive the following equation

$$\int_0^\infty \frac{\sin^2(a\omega)}{\omega^2} \, \mathrm{d}\omega = \frac{\pi}{2} \, a \quad \text{for } a > 0.$$

Exercise 5.5 (Discrete Fourier Transform)
Let $N \in \mathbb{N}$ and for convenience, let us always assume N to be an even number. Consider the one dimensional signal $f = (f_1, \ldots, f_N) \in \mathbb{C}^N$. Outside of the set $\{1, \ldots, N\}$, we will continue f periodically, as shown in Exercise 5.3.

We define the *discrete Fourier transform* $\mathcal{F}_N\colon \mathbb{C}^N \to \mathbb{C}^N$, $f \mapsto \mathcal{F}_N(f)$ *of* f via

$$\mathcal{F}_N(f)_j \coloneqq \sum_{k=1}^{N} f_k \cdot \mathrm{e}^{-\mathrm{i}\omega_{j-1}(k-1)}, \quad \text{for} \quad j = 1, 2, \ldots, N. \tag{5.35}$$

where $\omega_j \coloneqq 2\pi j/N$.

We define the *inverse discrete Fourier transform of* $F = (F_1, \ldots, F_N) \in \mathbb{C}^N$ via

$$\mathcal{F}_N^{-1}(F)_j \coloneqq \frac{1}{N} \sum_{k=1}^{N} F_k \cdot \mathrm{e}^{\mathrm{i}\omega_{k-1}(j-1)}. \tag{5.36}$$

Note the prefactor $\frac{1}{N}$ to the sum in (5.36).

(a) In the complex plane $\mathbb{C}$ for $N = 4$, sketch the trajectories of $\mathrm{e}^{\mathrm{i}\omega_{j-1}(k-1)}$ for $k = 1, 2, 3, 4$. Make a new sketch for each $j \in 1, 2, 3, 4$.

Hint: Recall Euler's formula $\mathrm{e}^{\mathrm{i}\varphi} = \cos(\varphi) + \mathrm{i}\sin(\varphi)$, $\varphi \in \mathbb{R}$.

(b) Write two Octave/MATLAB functions

`function [F] = dft(f)` and `function [f] = idft(F)`

that calculate the discrete (inverse) Fourier transform of a signal f (Fourier coefficients F) as defined in equation (5.35) and equation (5.36), respectively.

(c) Let $f = \Big(\cos(k\pi/2)\Big)_{k=0}^{3} = (1, 0, -1, 0)$ and $g = (1, 1, 1, 1)$. Then

$$\mathcal{F}_4(f) = (0, 2, 0, 2) \quad \text{and} \quad \mathcal{F}_4(g) = (4, 0, 0, 0)\,.$$

Interpret the results.

Note that $\Big(\cos(k\frac{\pi}{2})\Big)_{k=0}^{3} = \Big(\cos(k\frac{3\pi}{2})\Big)_{k=0}^{3}$ and $\Big(\sin(k\frac{\pi}{2})\Big)_{k=0}^{3} = -\Big(\sin(k\frac{3\pi}{2})\Big)_{k=0}^{3}$.

(d) Prove that $\mathcal{F}_N^{-1}$ is indeed the inversion of $\mathcal{F}_N$, i.e. show that for $f, F \in \mathbb{C}^N$, the equalities $\mathcal{F}_N^{-1}(\mathcal{F}_N(f)) = f$ and $\mathcal{F}_N(\mathcal{F}_N^{-1}(F)) = F$ hold.

(e) Show that the following identity holds

$$\mathcal{F}_N^{-1}(F)_j = \frac{1}{N} \sum_{k=-N/2+1}^{N/2} F_k \cdot \mathrm{e}^{\mathrm{i}\omega_{k-1}(j-1)}\,, \tag{5.37}$$

where $F \in \mathbb{C}^N$ is continued periodically, when necessary. Compare with equation (5.36), and argue which formula is better suited for the interpretation of Fourier coefficients.

(f) Specify an *orthogonal basis* $(b_1, \ldots, b_N)$ of $\mathbb{C}^N$ such that for $f \in \mathbb{C}^N$ we have

$$f = \sum_{k=1}^{N} \mathcal{F}_N(f)_k \, b_k.$$

Hint: $f = \mathcal{F}_N^{-1}(\mathcal{F}_N(f))$.

Note: The complex vector space $\mathbb{C}^N$ *with the* scalar product $\langle u, v\rangle = \sum_{k=1}^{N} u_k \, \overline{v_k}$ *is a* Hilbert space. *In particular, we have that* $\mathcal{F}_N(f)_k = \langle f, b_k\rangle$ *for all* $k = 1, \ldots, N$.

(g) The Fourier coefficients of a given signal $f \in \mathbb{R}^8$ read

$$(\mathcal{F}_8(f)_j)_{j=1}^{8} = (0, 0, -4\mathrm{i}, 0, 0, 0, 4\mathrm{i}, 0) \, .$$

Find $\omega > 0$ such that $f_j = \sin(\omega \frac{j-1}{N})$ for $j = 1, \ldots, N$.

(h) Show that formula (5.35) for the Fourier transform can be extended periodically, i.e. show that

$$\mathcal{F}_N(f)_{j+lN} = \mathcal{F}_N(f)_j$$

holds for all $j = 1, 2, \ldots, N$ and $l \in \mathbb{Z}$.

(i) Show that for $f \in \mathbb{R}^N$, the following relation holds

$$\mathcal{F}_N(f)_{N-j} = \overline{\mathcal{F}_N(f)_j}$$

for all $j \in \mathbb{Z}$, extending $\mathcal{F}$ periodically if necessary. Under which condition does the inverse Fourier transform give a real signal?

(j) Fourier transform and convolution: Show that the discrete analogue of the *convolution theorem* holds, i.e. for $f, g \in \mathbb{C}^N$, we have that for their circular convolution (see (5.34)) the identity $\mathcal{F}_N(f \otimes g) = \mathcal{F}_N(f) \cdot \mathcal{F}_N(g)$ holds.

Exercise 5.6 (Frequency filtering using the discrete Fourier transform)
Usually, when visualizing a discrete Fourier transform (DFT), one wants to localize low frequencies[1] in the middle of the resulting DFT vector. At the same time, this localization makes the application of a frequency filter more intuitive. This localization is accomplished by shifting a vector of discrete Fourier coefficients. In Octave/MATLAB this shift is implemented in the functions `fftshift` and `ifftshift` (for the inversion).

For convenience, we will assume N to be even and continue signals and their Fourier coefficients periodically, if necessary. For even N, the functions `fftshift` and `ifftshift` coincide.

[1] Note that due to formula (5.37), our discrete Fourier transform will only detect (angular) frequencies that are smaller than the so called *Nyquist*-frequency $\Omega_{\mathrm{Nyq}} = \pi$. Therefore, we will identify frequencies $\omega_j > \Omega_{\mathrm{Nyq}}$ with their negative counterpart $\omega_j - 2\pi$.

(a) Let $F_1, \ldots, F_N$ be Fourier coefficients of a signal $f \in \mathbb{C}^N$. Show that the following relation is true

$$(F_1, F_2, \ldots, F_N) = (F_1, F_2, \ldots, F_{N/2}, F_{-N/2+1}, F_{-N/2}, \ldots, F_0) \, .$$

This means that the vector $(F_1, F_2, \ldots, F_N)$ contains the same frequency information as the vector $(F_{-N/2+1}, F_{-N/2}, \ldots, F_0, F_1, \ldots, F_{N/2})$.

(b) Consider the index shift

$$\Big(1, 2, \ldots, \frac{N}{2}, \frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1, N\Big)$$

$$\mapsto \quad \Big(\frac{N}{2}+1, \frac{N}{2}+2, \ldots, N-1, N, 1, 2, \ldots, \frac{N}{2}\Big)$$

as carried out using the OCTAVE/MATLAB function `fftshift`. This shift accomplishes the aforementioned ordering of the Fourier coefficients. Specify the index set $M \subset \{1, \ldots, N\}$ that (in the shifted sequence of Fourier coefficients) corresponds to the Fourier coefficients of the absolute frequencies $\omega_0 = 0, |\omega_1|, \ldots, |\omega_d|$ for $d > 0$.

(c) Based on your findings, implement a simple *low pass filter*

```
function [ f_0 ] = simpleLp(f,d)
```

that for a given signal f and number of frequencies d, returns the filtered signal f_0 that only contains the smallest d frequencies and the frequency ω_0. You may assume in your implementation that f has even length and use the OCTAVE/MATLAB functions `fft` and `ifft` in order to compute the (inverse) discrete Fourier transform, respectively.

(d) Write a script `applySimpleLp.m` that, for the given signal

```
t = 0:(1/64):1 - 1/64; f = sin(2*pi*t) + cos(2*pi*8*t)
```

filters out the high frequency part using the `simpleLp` function from before. Plot both f and f_0 in one figure to verify your result.

Exercise 5.7 (Discrete Fourier Transform in 2D and frequency filtering for images) (a) Derive a formula for the 2D discrete Fourier transform similar to equation (5.35).

Hint: Recall how the 2D continuous *Fourier transform in is "composed" of two 1D continuous Fourier transforms and apply the same construction in the discrete case.*

Note: For the 2D discrete Fourier transform, analogous properties hold as for the 1D discrete Fourier transform.

(b) Let $S_{M,d} \coloneqq \Big[(\frac{M}{2}+1) - d, (\frac{M}{2}+1) + d\Big]^2 \subset \mathbb{R}^2$.

(i) Write an OCTAVE/MATLAB script `visualizeFT.m` that visualizes the absolute value of the 2D Fourier transform of the discrete image

$$u\colon \{1,2,\dots,512\}^2 \to F_{\mathrm{gr}}$$

which is defined via

$$u(x,y) \coloneqq \chi_{S_{512,10}}(x,y) = \begin{cases} 1 & \text{if } (x,y) \in S_{512,10}, \\ 0 & \text{else,} \end{cases}$$

where $\chi_{S_{M,d}}$ denotes the *characteristic function* of $S_{M,d}$. To this end, use the OCTAVE/MATLAB command `fft2` to compute the 2D Fourier transform and the commands `meshgrid` and `surf` for visualizing, e.g.

```
surf(X,Y,log(1+abs(fftshift(F))),'EdgeColor','none')
```
(5.38)

if your grid is generated from `X` and `Y` and `F` denotes the Fourier transform of u. If the Fourier transform is not dominated by very bright values in the center, the scaling `log(1 + ...)` may not be necessary. Also visualize the FFT using the following command:

```
imagesc(log(1+abs(fftshift(F)))); colormap(gray)
```
(5.39)

(ii) Explain what happens if you leave out the command `fftshift` in command (5.38) or (5.39).

(c) Low-Pass Filter Design: Write an OCTAVE/MATLAB function

```
function [B] = applyLp(A,d)
```

which, given a matrix `A` that represents an 8-bit grayscale image and a parameter `d`, applies an ideal low-pass filter to the Fourier transform of `A` (i.e. in the frequency domain) and transforms the resulting image back to the spatial domain. In the frequency domain, a continuous ideal low-pass filter with parameter d is given by $\chi_{S_{M,d}}$. In the continuous case, the procedure could be represented by the *basic filtering equation*

$$g(x,y) = \mathrm{Re}\left[\mathcal{F}^{-1}(\mathcal{F}(u)\cdot\chi_U)(x,y)\right],$$

where Re denotes the real part of a complex number and $U \subseteq \mathbb{R}^2$ is a (point) symmetric domain in $\mathbb{R}^2$. For your implementation, convert your input image to double using the command `im2double` before applying the Fourier transform `fft2`. Convert your final result back to 8-bit grayscale using the command `im2uint8`. Note that this will truncate negative values to 0.

(d) Removing periodic noise: Write a script `removePNoise.m` in which you use the function `applyLp` from the previous part in order to remove the periodic pattern from the image `mandril_pnoise.png`. [2] In order to find a suitable size of your ideal low-pass filter, try to spot the corresponding frequencies in the FFT output of the noisy image. To this end, use the command (5.39) to visualize the FT of `mandril_pnoise.png`.

[2] You find this image in the uploaded material to this exercise or directly at https://collaborating.tuhh.de/e-10/teaching/mathematical-image-processing/lecture-notes/-/tree/master/example_img

Exercise 5.8 (Heat Equation)
A prototype of *parabolic PDEs* on $\mathbb{R}^d$ is the *heat equation*

$$\begin{cases} \partial_t u(x,t) = \Delta u(x,t) & (x,t) \in \mathbb{R}^d \times (0,T], \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d, \end{cases} \tag{5.40}$$

where $T > 0$ and the *initial value* $u_0 \colon \mathbb{R}^n \to \mathbb{R}$ is given. Here, $\Delta = \sum_{k=1}^{d} \frac{\partial^2}{\partial x_k^2}$ denotes the *Laplacian*.

(a) Show that the d-dimensional Gauß kernel

$$G_s^{(d)}(x) \coloneqq G_s^{(d)}(x_1, \dots, x_d) \coloneqq \frac{1}{(2\pi s^2)^{d/2}} \mathrm{e}^{-\frac{|x|^2}{2s^2}}$$

with $s = \sqrt{2t}$ solves the equation $\partial_t u(x,t) = \Delta u(x,t)$ on $\mathbb{R}^d \times \mathbb{R}$.
Here $|x| \coloneqq (x_1^2 + x_2^2 + \cdots + x_d^2)^{1/2}$ denotes the *euclidean norm of* x.

(b) Let $u_0 \colon \mathbb{R}^n \to \mathbb{R}$ be given. Argue that a solution to (5.40) is given via $u(t,x) \coloneqq (G_{\sqrt{2t}} * u_0)(x)$. Does u_0 have to be continuous? Does the *smoothness*, i.e. the differentiability, of the solution $u(t,x)$ depend on the smoothness of the initial value u_0? Justify your answers briefly, a full proof is not required.

Exercise 5.9 (Multiscale Analysis and Contrast Invariance)
Scale-space theory or multiscale theory takes an axiomatic approach to image processing by postulating specific properties to characterize and derive image processing methods. One central object in scale-space theory is the *multiscale analysis*.

Definition. A *multiscale analysis* is a family of transformations $(\mathcal{T}_t)_{t \geq 0}$ with the mapping property

$$\mathcal{T}_t \colon \mathcal{BUC}(\mathbb{R}^d) \to \mathcal{BUC}(\mathbb{R}^d) \quad \text{for all } t \geq 0,$$

where

$$\mathcal{BUC}(\mathbb{R}^d) \coloneqq \{u \in \mathrm{C}(\mathbb{R}^d) \colon u \text{ uniformly continuous and bounded}\}.$$

The idea behind this definition is that for a given image $u \colon \Omega \to \mathbb{R}$, the *scale parameter* t describes representations of u at coarser scales as t gets bigger.

The image analysis must be independent of the (arbitrary) gray-level scale. Therefore an important axiom is

$$\mathcal{T}_t(0) = 0, \quad \mathcal{T}_t(u + c) = \mathcal{T}_t(u) + c \quad \text{for all images } u \text{ and all constants } c. \tag{GLSI}$$

A stronger version of this axiom is

$$\mathcal{T}_t(h(u)) = h(\mathcal{T}_t(u)) \quad \text{for all images } u \text{ and } t \geq 0, \text{ where } h \text{ is any non-decreasing function.} \tag{GLI}$$

You find further information on multiscale analysis in [2] and [4].

(a) Prove that a multiscale analysis that fulfills axiom (GLI) necessarily fulfills axiom (GLSI).

(b) Let $(\mathcal{T}_t)_{t\geq 0}$ be the multiscale analysis coming from the heat equation, i.e. for an initial value u_0 we have that $u(t,\cdot) \coloneqq \mathcal{T}_t(u_0)$ for all $t > 0$ fulfills the differential equation

$$\partial_t u = \Delta u.$$

Show that, for general h, the multiscale analysis $(\mathcal{T}_t)$ fulfills axiom (GLI) but not axiom (GLSI). Under which additional condition is (GLSI) fulfilled? Do you know examples for linear and non-linear grayscale transformations? (*Hint: Exercise 3.2*)

Note: This shows that only a non-linear multiscale analysis (e.g. solutions to the Perona-Malik equation) will achieve (GLI).

(c) In Exercise 3.2, you already saw different versions of h. Show that (b) makes sense for the linear(!) transformation contrast stretching, i.e. it makes no difference if you first enhance the contrast or smooth and later enhance the contrast.

Exercise 5.10 (Nonlinear Diffusion)
The idea of Perona and Malik [21] was to overcome the inherent problems of the heat equation. The *Perona-Malik-equation* on $\mathbb{R}^d$ reads

$$\begin{cases} \partial_t u(x,t) = \operatorname{div}(g(|\nabla u(x,t)|)\nabla u(x,t)) & (x,t) \in \mathbb{R}^d \times (0,T], \\ u(x,0) = u_0(x) & x \in \mathbb{R}^d, \end{cases} \tag{5.41}$$

where $g\colon [0,\infty) \to [0,\infty)$ is a smooth function. Common examples for g are the functions

$$g_1(s) = \frac{1}{1+\frac{s^2}{\kappa^2}}, \qquad g_2(s) = \mathrm{e}^{-\frac{s^2}{2\kappa^2}}, \tag{5.42}$$

where $\kappa > 0$ is a parameter that *controls* how fast the functions go to zero for $s \to \infty$, see Figure 5.2.

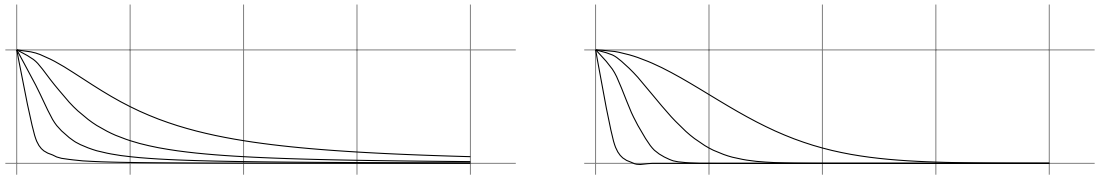

Figure 5.2: Left: $g_1(s)$, right: $g_2(s)$ for $\kappa = 1, \frac{1}{2}, \frac{1}{4}, \frac{1}{16}$ (in each plot from right to left).

(a) Perona-Malik and Edges: We want to understand what Perona-Malik does to *edges*. To this end, we switch to the 1D setting and consider only the 1D version of equation (5.41)

$$\partial_t u = (g(|u'|)u')' \tag{5.43}$$

Let us first define what we mean when we talk about an *edge*.

Definition. We say that $u\colon \mathbb{R} \to \mathbb{R}$ has an *edge* in $x_0 \in \mathbb{R}$ if the following conditions hold

1. $u'(x_0) > 0$
2. $u''(x) = 0$
3. $u'''(x_0) < 0$.

The first condition ensures that our image is not flat, the second and third guarantee certain type of inflection point. Figure 5.3 shows an example curve for each of the two different types of inflection points.

(i) Let u be a sufficiently smooth solution to (5.43). Furthermore, let $u(\cdot, t_0)$ have an edge in x_0 with $u''''(x_0, t_0) = 0$. Last, let us define the *flux* f via $f(s) := sg(s)$. Then the following conditions hold in (x_0, t_0):

1. $\partial_t u' = f'(u')u'''$
2. $\partial_t u'' = 0$
3. $\partial_t u''' = 3f''(u')(u''')^2 + f'(u')u''''$.

Prove statement 1. Start by rewriting (5.43) in terms of the flux f using the *chain rule*.

Note: The first condition tells us whether edges become steeper or more flat with time. The second condition tells us that edges remain inflection points, see 2. in the definition. The last condition tells us if an inflection point is trying to change its kind, see Figure 5.3.

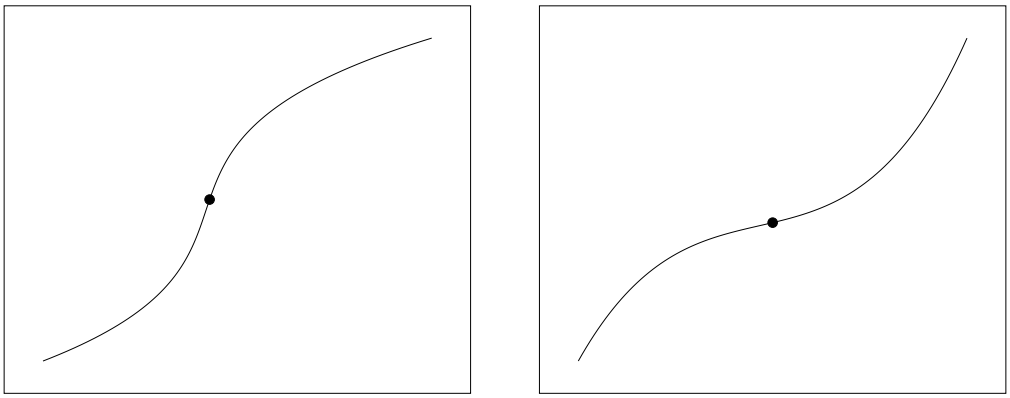

Figure 5.3: Left: Inflection point of the first kind (an *edge*), right: inflection point of the second kind (not an *edge*).

(ii) Now, in addition to (i), the diffusion coefficient g is g_1 from equation (5.42). Furthermore, assume that in the edge (x_0, t_0) we have that $u'''' > 0$.

Prove that in (x_0, t_0) the following holds:

1. $\partial_t u' > 0$ if $u' > \kappa$ and $\partial_t u' < 0$ if $u' < \kappa$
2. $\partial_t u'' = 0$
3. $\partial_t u''' < 0$ if $\kappa < u' < \sqrt{3}\kappa$.

Note: The first point tells us that steep edges will get steeper and flat edges will get more flat. The third point tells us that if an edge is steeper than $\sqrt{3}\kappa$ it will try to change its type. Sometimes this will lead to a so-called staircasing-effect.

(b) Let $\Omega \subseteq \mathbb{R}^d$, $u_0 \in \mathrm{L}^2(\Omega)$, $g\colon [0,\infty) \to [0,\infty)$ continuous and u a solution to the Perona-Malik initial-boundary-value problem

$$\begin{cases} \partial_t u = \operatorname{div}(g(|\nabla u|)\nabla u) & \text{in } \Omega \times [0,\infty), \\ \partial_\nu u = 0 & \text{on } \partial\Omega \times [0,\infty), \\ u(x,0) = u_0(x) & \text{for } x \in \Omega. \end{cases} \tag{5.44}$$

(i) Show that

$$\mu(t) \coloneqq \int_\Omega u(x,t)\,\mathrm{d}x \tag{5.45}$$

is constant.

(ii) Show that

$$h(t) \coloneqq \int_\Omega u(x,t)^2\,\mathrm{d}x \tag{5.46}$$

is monotone decreasing.

Hint: For part (i), recall that a function with vanishing derivative is constant. For part (ii) use Leibniz's rule for differentiation *under the integral sign and a similar argument. In order to apply the boundary conditions, use the following theorem.*

Theorem. (Gauß-Green Theorem, Integration-by-parts [11, C.2 Theorem 1])

Let Ω be a bounded, open subset of $\mathbb{R}^d$ with smooth boundary and outer normal vector ν. For $f, g \in \mathrm{C}^1(\Omega)$ and $i = 1, \dots, d$, we have that

(i) $\int_\Omega \partial_i f\,\mathrm{d}x = \int_{\partial\Omega} f\nu_i\,\mathrm{d}S$

Here $\partial\Omega$ denotes the *boundary* of the set Ω and $\int_{\partial\Omega} \cdot\,\mathrm{d}S$ denotes the respective integral along $\partial\Omega$. If $F \in \mathrm{C}^1(\Omega, \mathbb{R}^d)$, we get that

(i') $\int_\Omega \operatorname{div}(F)\,\mathrm{d}x = \int_{\partial\Omega} \langle F, \nu\rangle\,\mathrm{d}S$

Furthermore, it holds that $\int_\Omega (\partial_i f) g\,\mathrm{d}x = -\int_\Omega f\partial_i g\,\mathrm{d}x + \int_{\partial\Omega} fg\nu_i\,\mathrm{d}S$

Note: The function μ from equation (5.45) measures the mass *of the image and the function h from equation (5.46) quantifies the energy of the image.*

(c) Calculate the solution to equation (5.44) for $d = 2$, $g = g_1$ and $\kappa =$ `kappa` via an *explicit Euler scheme*. To this end, write an Octave function

```
function [U] = peronaMalikSteps(U_0,kappa,nsteps,dt)
```

that gets an initial 8-bit grayscale image `U_0` and calculates `nsteps` time steps of size `dt` of the explicit Euler scheme. At the beginning, convert the image to `double` using the Octave function `im2double`. In the end you should return the result as 8-bit image; use `im2uint8` for the conversion. Use discrete versions of the differential operators as seen in the lecture. To this end the command `filter2` may be helpful.
Note: Due to rounding errors from the conversion from `double` *to* `uint8`*, it will be difficult to see any changes if the value for* `nsteps` *is chosen too small.*

(d) Write an OCTAVE script `denoisePeppersPM.m` that accomplishes the following:

- Reading in the image `peppers_gray.png`. [3]
- Adding noise to the image using the command `imnoise` with the parameters `'gaussian'`, `mean = 0`, `variance = 0.01`.
- Denoising the image by repeatedly calling `peronaMalikSteps` (e.g. with parameters `nsteps = 10`, `dt = 0.1` and `kappa = 0.05`).
- Visualizing the result after each termination of the function `peronaMalikSteps`.
  Note: Consider adding a `pause(0.1)` *after each visualization in order to let the figure refresh.*

Exercise 5.11 (Semigroup Theory)
A prototype of *parabolic PDEs* is the *heat equation* with *Neumann* boundary conditions

$$\begin{cases} \partial_t u(x,t) = \Delta u(x,t) & (x,t) \in \Omega \times (0,T], \\ \partial_\nu u(x,t) = 0 & (x,t) \in \partial\Omega \times (0,T], \\ u(x,0) = f(x) & x \in \Omega, \end{cases} \tag{5.47}$$

where $\Omega \subseteq \mathbb{R}^d$ is a bounded smooth domain, $T > 0$ and the *initial value* $f \colon \Omega \to \mathbb{R}$ is given. Here, $\partial_\nu u(x)$ denotes the *normal derivative* of u at the boundary $\partial\Omega$ and Δ is the *Laplacian*.

(a) Calculate the solution to equation (5.47) for $d = 2$ via an *explicit Euler scheme*. To this end, write an OCTAVE function

```
function [U] = heatEquationStep(U_0,t)
```

that gets an initial value `U_0` and calculates one time step of size `t` of the explicit Euler scheme. Here `U_0` and `U` should be assumed to be 8-bit grayscale image matrices. Use the OCTAVE function `filter2` in order to calculate the gradients. Express the gradients of second order by calculating the first derivative using a *forward difference* and then using a *backward difference*. Carry out the necessary modifications in order to correctly implement the Neumann boundary condition.

(b) Write an OCTAVE script `denoisePeppersHE.m` that accomplishes the following:

- Reading in the image `peppers_gray.png`. [4]
- Adding noise to the image using the command `imnoise` with the parameters `'gaussian'`, `mean = 0`, `variance = 0.01`.
- Denoising the image by calculating 100 steps of the denoising process via `heatEquationStep` time step size `dt = 0.1`.

[3] You find this image in the uploaded material to this exercise or directly at https://collaborating.tuhh.de/e-10/teaching/mathematical-image-processing/lecture-notes/-/tree/master/example_img

[4] You find this image in the uploaded material to this exercise or directly at https://collaborating.tuhh.de/e-10/teaching/mathematical-image-processing/lecture-notes/-/tree/master/example_img

- Visualizing the result after each termination of the function `heatEquationStep`.
  Note: Consider adding a `pause(0.1)` *after each visualization in order to let the figure refresh.*

Now, consider the heat equation on $\mathbb{R}^d$

$$\begin{cases} \partial_t u(x,t) = \Delta u(x,t) & (x,t) \in \mathbb{R}^d \times (0,T], \\ u(x,0) = f & x \in \mathbb{R}^d, \end{cases} \tag{5.48}$$

A solution to the PDE (5.48) is given via convolution with the so-called *heat-kernel*

$$G_{\sqrt{2t}}(x) = (4\pi t^2)^{-\frac{d}{2}} \mathrm{e}^{-\frac{|x|^2}{4t}}.$$

Given a (sufficiently regular) initial value u_0, the solution to (5.48) at $t \in (0,T]$ reads

$$u(x,t) = T_t u_0(x), \quad x \in \mathbb{R}^d,$$

where

$$T_t u_0(x) \coloneqq (G_t * u_0)(x).$$

Consider the *Schwartz space* $\mathcal{S}(\mathbb{R}^d) \subset \mathbb{C}^\infty(\mathbb{R}^d), p \in [1,\infty]$.[5] This space is left invariant under T_t, i.e. for $u_0 \in \mathcal{S}(\mathbb{R}^d)$ we have that $T_t u_0 \in \mathcal{S}(\mathbb{R}^d)$ again. Another important property is that $\mathcal{S}(\mathbb{R}^d) \subseteq \mathrm{L}^1(\mathbb{R}^d)$ which allows us to define the Fourier transform of Schwartz-functions $f \in \mathcal{S}(\mathbb{R}^d)$. Like for the space $\mathrm{L}^2(\mathbb{R}^d)$, the Fourier transform leaves the space $\mathcal{S}(\mathbb{R}^d)$ invariant.

(c) Show that the Fourier image of the operator T_t, i.e. for fixed $u_0 \in \mathcal{S}(\mathbb{R}^d)$ is a *multiplication operator* on $\mathcal{S}(\mathbb{R}^d)$, i.e. show that

$$\mathcal{F}(T_t u) = M_f(\mathcal{F}(u))$$

for some $f \in \mathcal{S}(\mathbb{R}^d)$, where we define the multiplication operator M_f on the space $\mathcal{S}(\mathbb{R}^d)$ via

$$M_f(u) = f \cdot u$$

for all $u \in \mathcal{S}$. More on the subject of operators and associated semigroups can be found at [10, Chapter II].

Exercise 5.12 (The Sobolev Space $\mathrm{H}^1(\mathbb{R}^d)$ and Weak Solutions)
Let $\Omega \subseteq \mathbb{R}^d$ be open. In this exercise, we want to get a first feel for functions $f \in \mathrm{H}^1(\Omega)$. We will see that weakly differentiable functions behave like continuously differentiable functions when considered under an integral sign.

[5] For this exercise, it is not required that you know this space in detail. Just think of a set of very special smooth functions.

Notation. (Smooth functions with compact support)
The set of real-valued functions that are infinitely differentiable will be denoted by $\mathrm{C}^\infty(\mathbb{R}^d)$. For a continuous function f we define its support $\operatorname{supp}(f)$ via

$$\operatorname{supp}(f) \coloneqq \overline{\{x \in \mathbb{R}^d \colon f(x) \neq 0\}},$$

where $\overline{\{\dots\}}$ denotes the *closure* of the set $\{\dots\}$, e.g.

$$\overline{(0,1]} = [0,1] \quad \text{or} \quad \overline{\{x \in \mathbb{R}^n \colon |x| < 1\}} = \{x \in \mathbb{R}^n \colon |x| \leq 1\}\,.$$

More general, if $\Omega \subseteq \mathbb{R}^n$ is open, then $\overline{\Omega} = \Omega \cup \partial\Omega$.

Compare the functions

$$f(x) = \mathrm{e}^x \quad \text{and} \quad g(x) = \begin{cases} \mathrm{e}^{-\frac{1}{1-x^2}} & \text{for } x \in (-1,1) \\ 0 & \text{else} \end{cases}.$$

From Figure 5.4 it is clear what the supports of the functions f and g are, namely

$$\operatorname{supp}(f) = \mathbb{R} \quad \text{and} \quad \operatorname{supp}(g) = [-1,1]\,.$$

For the function g, we see that the support is a *bounded* set and therefore *compact*. The set of all functions that are infinitely differentiable and have *compact support* will be denoted by $\mathrm{C}_\mathrm{c}^\infty(\mathbb{R}^n)$. Sometimes, we only consider functions that live on an open set $\Omega \subseteq \mathbb{R}^d$. In this case, infinitely differentiable compactly supported functions are collected in the set $\mathrm{C}_\mathrm{c}^\infty(\Omega)$. Note that since Ω is open, for $f \in \mathrm{C}_\mathrm{c}^\infty(\Omega)$ we always have that $f(x) = 0$ for $x \in \partial\Omega$.

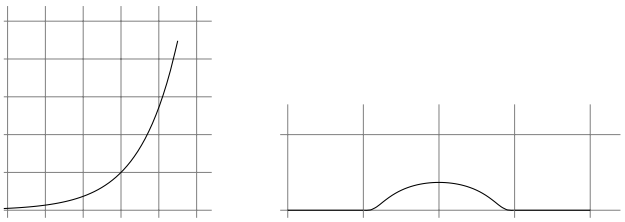

Figure 5.4: Left: function plot of f. Right: function plot of g.

(a) Fix $f \in \mathrm{C}^1(\Omega)$. Show that for all $\varphi \in \mathrm{C}_\mathrm{c}^\infty(\Omega)$ the following identity holds

$$\int_\Omega f(x) \frac{\partial \varphi}{\partial x_i}(x)\,\mathrm{d}x = -\int_\Omega \frac{\partial f}{\partial x_i}(x)\varphi(x)\,\mathrm{d}x. \tag{5.49}$$

The idea of Sobolev functions in comparison to functions in $\mathrm{C}^1(\Omega)$ is now to collect all functions f that have the following properties:

- $f \in \mathrm{L}^2(\Omega)$, i.e. $\|f\|_2 < \infty$.

- For all $i = 1, \dots, d$ there exists $v \in \mathrm{L}^2(\Omega)$ such that for all $\varphi \in \mathrm{C}_\mathrm{c}^\infty(\Omega)$

$$\int_\Omega f(x) \frac{\partial \varphi}{\partial x_i}(x)\,\mathrm{d}x = -\int_\Omega v(x)\varphi(x)\,\mathrm{d}x. \tag{5.50}$$

In this case, we write $f \in \mathrm{H}^1(\Omega)$ and we call v the *weak partial derivative of f in direction of x_i*. In order to ease notation, we will write $v = \frac{\partial f}{\partial x_i}$. In the same way, we will understand the *weak gradient* ∇f for some $f \in \mathrm{H}^1(\Omega)$. You find more on Sobolev spaces in the lecture notes [22]. We will come back to Sobolev spaces when we discuss distributional derivatives in the last chapter of the lecture.

Now, we will consider partial differential equations and see how this new *notion of differentiability* gives rise to a new *notion of solvability* of partial differential equations. Consider the *Poisson equation*

$$-\Delta u = f \quad \text{on } \mathbb{R}^d. \tag{5.51}$$

Given some $f \in \mathrm{L}^2(\mathbb{R}^d)$, we call $u \in \mathrm{H}^1(\mathbb{R}^d)$ a *weak solution* to (5.51) if for all $\varphi \in \mathrm{H}^1(\mathbb{R}^d)$ the following equation is fulfilled

$$\sum_{i=1}^{d} \int_{\mathbb{R}^d} \frac{\partial u}{\partial x_i}(x) \frac{\partial \varphi}{\partial x_i}(x)\,\mathrm{d}x = \int_{\mathbb{R}^d} f(x)\varphi(x)\,\mathrm{d}x. \tag{5.52}$$

We will also call (5.52) a *weak formulation* or *weak version* of the Poisson equation (5.51).

(b) Derive equation (5.52) from equation (5.51) in the following way:

  (i) Multiply both sides of (5.51) with some $\varphi \in \mathrm{H}^1(\mathbb{R}^d)$.

  (ii) Integrate both sides of the resulting equation over the domain $\mathbb{R}^d$. Then use "integration by parts" (5.50) in order to derive an equation that only involves first derivatives.

(c) Show that for a given $f \in \mathrm{L}^2(\mathbb{R}^d)$, there exists at most one weak solution to equation (5.51). Proceed as follows:

  (i) Suppose you have two solutions $u, v \in \mathrm{H}^1(\mathbb{R}^d)$. Their difference $u - v$ is a weak solution of which equation?

  (ii) Based on (i), show that $u - v$ has to be locally constant.
  Hint 1: A function $u \in \mathrm{H}^1(\mathbb{R}^d)$ is locally constant if the weak gradient $\nabla u \in (\mathrm{L}^2(\mathbb{R}^d))^d$ equals 0.
  Hint 2: In every normed vector space X, we now that $\|x\| = 0$ implies $x = 0$. Use this fact for the space $\mathrm{L}^2(\mathbb{R}^d)$.

  (iii) Now conclude, that based on the fact $\mathrm{H}^1(\mathbb{R}^d) \subseteq \mathrm{L}^2(\mathbb{R}^d)$, the difference $u - v$ needs to be equal to zero.

Exercise 5.13 (Variational Calculus and L^2-H^1-denoising – direct approach)
Let $u^0 \in L^2(\mathbb{R}^d)$ a noisy image. For $\lambda > 0$ consider the functional

$$\phi\colon H^1(\mathbb{R}^d) \to \mathbb{R}, \quad u \mapsto \frac{1}{2}\int_{\mathbb{R}^d} |u^0(x) - u(x)|^2 \,\mathrm{d}x + \frac{\lambda}{2}\int_{\mathbb{R}^d} |\nabla u(x)|^2 \,\mathrm{d}x \tag{5.53}$$

In order to denoise u^0, we want to find out if there exists $u^* \in H^1(\mathbb{R}^d)$ such that

$$\phi(u^*) = \min_{u \in H^1(\mathbb{R}^d)} \phi(u). \tag{5.54}$$

In order to solve this problem by a direct calculation, we proceed step by step:

(a) Use Parseval's theorem and further properties of the Fourier transform to transform the minimization problem (5.54) such that under the integral sign only appear Fourier transforms of $L^2(\mathbb{R}^d)$ functions.

Now assume that it suffices to find a function $\mathcal{F}(u)$ which minimizes the integrand

$$\frac{1}{2}|\mathcal{F}(u^0)(\xi) - \mathcal{F}(u)(\xi)|^2 + \frac{\lambda}{2}|\xi|^2|\mathcal{F}(u)(\xi)|^2$$

pointwise, i.e. in each point $\xi \in \mathbb{R}^d$. The unknown value of $\mathcal{F}(u)(\xi)$ will now be denoted as some $z \in \mathbb{C}$. Accordingly, the minimizer u^* will be given via

$$\mathcal{F}(u^*)(\xi) = \arg\min_{z \in \mathbb{C}} \frac{1}{2}|\mathcal{F}(u^0)(\xi) - z|^2 + \frac{\lambda}{2}|\xi|^2|z|^2, \qquad \xi \in \mathbb{R}^d.$$

In order to locate the minimum of the real valued function

$$\frac{1}{2}|\mathcal{F}(u^0)(\xi) - z|^2 + \frac{\lambda}{2}|\xi|^2|z|^2, \tag{5.55}$$

we use the decomposition $z = |z|\operatorname{sgn}(z)$, where sgn denotes the *complex signum function*. The purpose of this decomposition is that for (5.55) to be minimized we can first minimize w.r.t. $\operatorname{sgn}(z)$ and then w.r.t. $|z|$.

(b) Show that the term term in (5.55) is equal to

$$\frac{1}{2}(1 + \lambda|\xi|^2)|z|^2 + \frac{1}{2}|\mathcal{F}(u^0)(\xi)|^2 - |z|\operatorname{Re}\big(\operatorname{sgn}(z)\overline{\mathcal{F}(u^0)(\xi)}\big). \tag{5.56}$$

If we minimize (5.55) w.r.t. $\operatorname{sgn} z$ we get the necessary condition

$$\operatorname{sgn}(z) = \operatorname{sgn}(\mathcal{F}(u^0)(\xi)).$$

(c) For this value of $\operatorname{sgn}(z)$, you should now find an appropriate $|z|$ that minimizes (5.55).

(d) Now find u^* by using the inverse Fourier transform. Use that for

$$P_\lambda = \frac{|x|^{1-d/2}}{(2\pi)^{d-1}\lambda^{(d+2)/2}}\,\mathrm{K}_{d/2-1}\Big(\frac{2\pi|x|}{\sqrt{\lambda}}\Big),$$

where $\mathrm{K}_{\mathrm{d}/2-1}$ denotes a *modified Bessel function of the second kind*, we have that

$$\mathcal{F}(P_\lambda)(\xi) = \frac{(2\pi)^{-d/2}}{1+\lambda|\xi|^2}.$$

Exercise 5.14 (Bilateral Filters)
Consider the d-dimensional Gauß kernel from Exercise 5.8

$$G_s^{(d)}(x) \coloneqq G_s^{(d)}(x_1,\dots,x_d) \coloneqq \frac{1}{(2\pi s^2)^{d/2}}\mathrm{e}^{-\frac{|x|^2}{2s^2}}$$

and note that

$$G_s^{(d)}(x) = \prod_{i=1}^{d}\frac{1}{(2\pi s^2)^{1/2}}\mathrm{e}^{-\frac{x_i^2}{2s^2}} = \prod_{i=1}^{d} G_s^{(1)}(x_i).$$

Given an image $u^{(0)}\colon \Omega \to F$, $\Omega = \{1,\dots,N\}\times\{1,\dots,M\} \subseteq \mathbb{R}^2$, a bilateral filter BF could be defined as follows (cf. [24])

$$\mathrm{BF}(u^{(0)})_{i,j} = \frac{1}{W_{i,j}}\sum_{(k,l)\in\Omega} G_{s_1}^{(2)}\Big(\begin{bmatrix}i\\j\end{bmatrix}-\begin{bmatrix}k\\l\end{bmatrix}\Big)\cdot G_{s_2}^{(1)}\Big(u_{i,j}^{(0)}-u_{k,l}^{(0)}\Big)\cdot u_{k,l}^{(0)}\,, \tag{5.57}$$

where

$$W_{i,j} = \sum_{(k,l)\in\Omega} G_{s_1}^{(2)}\Big(\begin{bmatrix}i\\j\end{bmatrix}-\begin{bmatrix}k\\l\end{bmatrix}\Big)\cdot G_{s_2}^{(1)}\Big(u_{i,j}^{(0)}-u_{k,l}^{(0)}\Big)\,. \tag{5.58}$$

You find further information on bilateral filtering in the survey [20].

(a) Write a *brute-force* implementation of the Gaussian bilateral filter using the above formulas.

To this end, write an Octave function

```
function [B] = applyBLFilter(A,s1,s2,r)
```

that receives a matrix representation `A` of an 8-bit grayscale image and applies the bilateral filter in a quadratic neighborhood with diameter $2r$, i.e. in equations (5.57) and (5.58) the summation reads $\sum_{(k,l)\in N_{i,j}}$, where

$$N_{i,j} = \{(k,l)\colon \max\{|i-k|,|j-l|\} \le r\}.$$

If the last argument is not passed, your algorithm should sum over all pixels. You can check the number of passed arguments via the built-in Octave function `nargin`.

(b) Write a function

```
function [B] = cartoonizeImg(A,s1,nIter)
```

that receives a matrix representation `A` of an 8-bit color image and applies the bilateral filter `nIter` times. The parameters s_1 and s_2 of the involved Gauß kernels should be taken from the input argument `s1` and the calculation

$$s_2 \coloneqq s_2^{(k)} \coloneqq \frac{1}{10}\Big(\max_{(i,j)\in\Omega} u_{i,j}^{(k)} - \min_{(i,j)\in\Omega} u_{i,j}^{(k)}\Big), \quad k \in \mathbb{N}_0 \tag{5.59}$$

respectively. The function $u^{(n)}$ in equation (5.59) denotes the value of the filtered image after the first n iterations.

In contrast to a brute force implementation, we want to look at a more efficient method to apply the bilateral Gauss filter now. To this end, we want to write equations (5.57) and (5.58) as a convolution. As equations (5.57) and (5.58) are non-linear, this is not directly possible. We will proceed according to [19]:

(c) Multiply equation (5.57) with $W_{i,j}$ and for each pixel $(i,j) \in \Omega$ write the resulting system of equations in vector form

$$\begin{pmatrix} W_{i,j} \cdot \mathrm{BF}(u)_{i,j} \\ W_{i,j} \end{pmatrix} = \dots$$

(d) The resulting filter weights still depend on the image u and therefore the filter is still non-linear. We introduce a further (artificial) summation over the color space F by deploying the characteristic function of the graph $\Gamma(u) \coloneqq \{(i,j,\xi) \in \Omega \times F \colon u_{i,j} = \xi\}$. Rewrite the result from part (c) in the form

$$F(i,j,u_{i,j}) \coloneqq \begin{pmatrix} W_{ij} \cdot \mathrm{BF}(u)_{i,j} \\ W_{ij} \end{pmatrix} = \sum_{(k,l,\xi)\in\Omega\times F} K\Big(\begin{bmatrix} i \\ j \\ u_{i,j} \end{bmatrix} - \begin{bmatrix} k \\ l \\ \xi \end{bmatrix}\Big) \cdot \begin{pmatrix} f_1(k,l,\xi) \\ f_2(k,l,\xi) \end{pmatrix} \tag{5.60}$$

for suitable functions K and (f_1, f_2).

(e) What is the advantage of the derived form (5.60) of the bilateral filter? How could you use this advantage in a numerical implementation? (you are not supposed to carry out the implementation!) Why would you expect this new implementation to be *more* efficient than the implementation from part (a)? Why would you expect this new implementation to be *less* efficient than the implementation from part (a)?