Exercise Sheet 2 (Chopping and Nodding)

Markus Haltmeier

Exercise 4. Let n, h be natural numbers and define the mapping $\mathbf{L}_{n,h} : \mathbb{R}^n \to \mathbb{R}^n$ by

$$\big(\mathbf{L}_{n,h}\, f\big)(i) := 2f(i) - f(i-h) - f(i+h)\,, \quad i \in \{1, \dots, n\}\ .$$

(Here $f(i) := 0$ if $i \notin \{1, \dots, n\}$.)
After discretization, the problem of chopping and nodding in one spatial consist in reconstructing a vector $f := \big(f(i)\big)_{i=1}^n \in \mathbb{R}^n$ from data $g := \big(\big(\mathbf{L}_{n,h}\, f\big)(i)\big)_{i=1}^n \in \mathbb{R}^n$.

(a) Illustrate the definition of $\mathbf{L}_{n,h}$.

(b) Is $\mathbf{L}_{n,h}$ linear, injective, surjective and/or bijective?

(c) Find the transformation matrix of $\mathbf{L}_{n,h}$ (with respect to the standard basis) and implement it in Matlab. (Use the functions `diag` or `spdiags`) .

(d) Write a Matlab function `chop(f,n,h)` that computes $\mathbf{L}_{n,h}\, f$. Apply the function `chop(f,n,h)` to each column of

```
u = double(imread('chop.bmp')).
```

Exercise 5. We define the operator $\mathbf{I}_n \otimes \mathbf{L}_{n,h} : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ by

$$\big(\mathbf{I}_n \otimes \mathbf{L}_{n,h}\, u\big)(i,j) := 2u(i,j) - u(i,j-h) - u(i,j+h)\,, \quad i,j \in \{1, \dots, n\}$$

Here, again, we set $u(i,j) := 0$ if $(i,j) \notin \{1, \dots, n\}^2$.

(a) Write a Matlab function `chop2d(u,n,h)` (using Exercise 4) that computes $\mathbf{I}_n \otimes \mathbf{L}_{n,h}\, u$ for an image $u \in \mathbb{R}^{n\times n}$.

(b) Write a Matlab function `ichop2d(v,n,h)` that reconstructs the image u from data $v := \mathbf{I}_n \otimes \mathbf{L}_{n,h}\, u$ by applying the inverse matrix of $\mathbf{L}_{n,h}$. (For example, use the Matlab function `inv(A)`.)

(c) Test `ichop2d(v,n,h)` with

```
u = double(imread('chop.bmp')); v = chop2d(u,n,h).
```

What happens if you use

```
unoisy = u + errpc * randn(size(u)) * max(u(:));
```

(with `errpc = 0.01` for example) instead of `u`. How does the result depend on h. Why?