



- 1 We say that two norms  $\|\cdot\|_\alpha$  and  $\|\cdot\|_\beta$  are equivalent in  $\mathbb{C}^n$  if there exist positive constants  $c_1$  and  $c_2$  independent of  $x$ , such that

$$c_1\|x\|_\alpha \leq \|x\|_\beta \leq c_2\|x\|_\alpha$$

for all  $x \in \mathbb{C}^n$ . Show that  $\|\cdot\|_1$ ,  $\|\cdot\|_2$  and  $\|\cdot\|_\infty$  are equivalent in  $\mathbb{C}^n$  by showing that for any vector  $x \in \mathbb{C}^n$ , the following inequalities hold:

- a)  $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$ ,
- b)  $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$ ,
- c)  $\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty$ .

- 2 Let  $\|\cdot\|$  be a vector norm on  $\mathbb{C}^n$  and consider the corresponding induced (or natural) matrix norm  $\|\cdot\|$ . Show that  $\rho(A) \leq \|A\|$  where  $\rho(A)$  is the spectral radius of  $A$ .

- 3 In the lecture (and in Saad's book), the matrix norm induced by  $\|\cdot\|_p$  and  $\|\cdot\|_q$  was defined as

$$\|A\|_{pq} := \max_{x \in \mathbb{C}^m \setminus \{0\}} \frac{\|Ax\|_p}{\|x\|_q}.$$

That is, the maximum is taken over all possible *complex* vectors  $x$ .

If  $A \in \mathbb{R}^{n \times m}$ , one can also define its *real matrix norm*

$$\|A\|_{pq, \mathbb{R}} := \max_{x \in \mathbb{R}^m \setminus \{0\}} \frac{\|Ax\|_p}{\|x\|_q}.$$

Consider now specifically the matrix

$$A = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

- a) Compute the real matrix norm  $\|A\|_{1\infty, \mathbb{R}}$ .
  - b) Using the vector  $x = (1 + i, 1 - i)^T$  in the definition of the matrix norm, show that, for this specific matrix, we have  $\|A\|_{1\infty} > \|A\|_{1\infty, \mathbb{R}}$ .
- 4 Suppose that  $E = uv^H$  is the outer product (or tensor product) of two vectors  $u, v \in \mathbb{C}^n$ .

- a) Show that  $\|E\|_2 = \|u\|_2 \|v\|_2$ .
- b) Decide if this also holds for the Frobenius norm, i.e., decide if  $\|E\|_F = \|u\|_2 \|v\|_2$ .

5 What can be said about the eigenvalues of a unitary matrix?

6 We will here investigate how the storage format and structure of a matrix influence the performance of the LU factorization of the matrix. MATLAB has two storage formats for matrices. We can either store them as full matrices, that is, all elements of the matrix are stored, or we can store them as sparse matrices where only non-zero elements and their positions are stored. The commands

```
F = full(S)
S = sparse(F)
```

convert a sparse matrix  $S$  into a full matrix  $F$  and a full matrix  $F$  into a sparse matrix  $S$  respectively. To depict the non-zero elements of  $A$  one may use the following command:

```
spy(A)
```

- a) We consider here the one-dimensional Poisson problem

$$\begin{aligned} -\frac{\partial^2 u}{\partial x^2} &= 4\pi^2 \sin(2\pi x), & x \in [0, 1], \\ u &= 0, & x \in \{0, 1\}. \end{aligned}$$

The MATLAB file `poisson1.m`, which can be fetched from the web page of the course, generates the system of linear equations obtained from discretizing the problem with a finite difference method on a uniform grid. For instance, the command

```
[A, b] = poisson1(n)
```

will return the system of equations with  $n$  unknowns. The matrix  $A$  will here be stored as a full matrix.

- i) For  $n = 900, 1600, 2500, 3600$ , generate the system of linear equations and measure the time it takes to solve the system with Gaussian elimination (i.e. with LU factorization). This can be done, for instance, with:

```
[A, b] = poisson1(n)
tic; [L, U] = lu(A); x = U \ (L \ b); toc
```

- ii) Repeat the experiment above, but convert  $A$  to a sparse matrix before the system is solved. Compare with i) and try to explain the difference.

(Note that it might be necessary to repeat the experiments several times in order to obtain reliable results.)

- b) We now consider the two-dimensional Poisson problem

$$\begin{aligned} -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) &= 5\pi^2 \sin(2\pi x) \sin(\pi y), & (x, y) \in [0, 1] \times [0, 1], \\ u &= 0, & x = 0, x = 1, y = 0, \text{ or } y = 1. \end{aligned}$$

The MATLAB file `poisson2.m` generates the system of linear equations we obtain when we discretize the above problem with a finite difference method. The command

```
[A, b] = poisson2(n)
```

will generate a system with  $N = n^2$  unknowns.

- i) For  $n = 30, 40, 50, 60$ , generate the system of linear equations and measure the time it takes to solve the system with Gaussian elimination. Compare with **a**).
- ii) Repeat step **i**), but convert  $A$  into a sparse matrix before solving the system. Compare with **a**). Check the structure of the matrices before and after Gaussian elimination in **a**) and **b**).

```
spy(A)  
[L, U] = lu(A)  
spy(L); spy(U)
```