TMA4205 Numerical Linear Algebra
Fall 2017

Norwegian University of Science and Technology
Department of Mathematical Sciences

Exercise set 1

$\boxed{1}$ 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

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

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) $\|x\|_2 \le \|x\|_1 \le \sqrt{n}\,\|x\|_2$,

b) $\|x\|_\infty \le \|x\|_2 \le \sqrt{n}\,\|x\|_\infty$,

c) $\|x\|_\infty \le \|x\|_1 \le n\|x\|_\infty$.

Possible solution:

a)

We have

$$\begin{aligned}\|x\|_2^2 = \sum_i |x_i|^2 \le \sum_i |x_i|\Big(\sum_j |x_j|\Big) = \Big(\sum_i |x_i|\Big)^2 = \|x\|_1^2 \\ = \sum_{i,j} |x_i||x_j| \le \sum_{i,j} \frac{1}{2}(|x_i|^2 + |x_j|^2) = n\sum_i |x_i|^2.\end{aligned}$$

b)

Here

$$\|x\|_\infty^2 = \max_i |x_i|^2 \le \sum_i |x_i|^2 = \|x\|_2^2 \le \sum_i \max_j |x_j|^2 = n\max_j |x_j|^2 = n\|x\|_\infty.$$

c)

This is essentially the same as in problem b), as

$$\|x\|_\infty = \max_i |x_i| \le \sum_i |x_i| = \|x\|_1 \le \sum_i \max_j |x_j| = n\max_j |x_j| = n\|x\|_\infty.$$

2 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 $\|A\|_{1\infty} > \|A\|_{1\infty,\mathbb{R}}$.

Possible solution:

a) The real matrix norm $\|A\|_{1\infty,\mathbb{R}}$ is in this case defined as

$$\|A\|_{1\infty,\mathbb{R}} = \max_{\|x\|_\infty\le 1} \|Ax\|_1 = \max_{|x_1|,\ |x_2|\le 1} \big(|x_1 - x_2| + |x_1 + x_2|\big).$$

Here the maximum is taken only over real numbers. In order to compute this maximum, we note that (for x_1, $x_2 \in \mathbb{R}$)

$$|x_1 - x_2| + |x_1 + x_2| = 2\max\{|x_1|, |x_2|\}.$$

Thus, actually

$$\|A\|_{1\infty,\mathbb{R}} = \max_{|x_1|,\ |x_2|\le 1} 2\max\{|x_1|, |x_2|\} = 2.$$

b)

We have

$$\left\| \begin{pmatrix} 1+i \\ 1-i \end{pmatrix} \right\|_\infty = \sqrt{2}$$

and

$$\left\| \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1+i \\ 1-i \end{pmatrix} \right\|_1 = \left\| \begin{pmatrix} 2i \\ 2 \end{pmatrix} \right\|_1 = 4.$$

Thus

$$\|A\|_{1\infty} = \max_{x\in\mathbb{C}^m\setminus\{0\}} \frac{\|Ax\|_1}{\|x\|_\infty} \ge \frac{4}{\sqrt{2}} = 2\sqrt{2} > 2 = \|A\|_{1\infty,\mathbb{R}}.$$

3 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 $\|E\|_{\mathrm{F}} = \|u\|_2\|v\|_2$.

Possible solution:

a) We may assume without loss of generality that u and v are different from 0. Because of the submultiplicativity of norms, we always have

$$\|uv^H\|_2 \le \|u\|_2\|v^H\|_2 = \|u\|_2\|v\|_2.$$

On the other hand

$$\|uv^H\|_2 = \max_{\|x\|\in\mathbb{C}^n\setminus\{0\}} \frac{\|uv^H x\|_2}{\|x\|_2} = \max_{\|x\|\in\mathbb{C}^n\setminus\{0\}} \frac{|\langle x, v\rangle|\|u\|_2}{\|v\|_2}.$$

Choosing $x = v$ in this maximum, we obtain that

$$\|uv^H\|_2 \le \frac{|\langle v, v\rangle|\|u\|_2}{\|v\|_2} = \|u\|_2\|v\|_2.$$

b)

Here we have

$$\|uv^H\|_2^2 = \sum_{ij}|u_i\overline{v}_j|^2 = \Big(\sum_i |u_i|^2\Big)\Big(\sum_j |v_j|^2\Big) = \|u\|_2^2\|v\|_2^2.$$

$\boxed{4}$ **a)** Assume that $A = uv^H$ is the outer product of two vectors u, $v \in \mathbb{C}^n$. Find the non-zero eigenvalues and corresponding eigenvectors of A. In addition, find the Jordan normal form of A.

b) Assume that $A \in \mathbb{C}^{n\times n}$ is an arbitrary quadratic matrix What is the relation between the eigenvalues and eigenvectors of A and the matrix $A + \lambda\,\mathrm{Id}$, where $\lambda \in \mathbb{C}$ and Id denotes the n-dimensional identity matrix?

c) Find the eigenvalues and an eigenbasis of the matrix

$$A = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix}.$$

Possible solution:

a)

We assume without loss of generality that u, $v \ne 0$; else the matrix A is trivial.

The vector $w \ne 0$ is an eigenvector of A, if and only if $Aw = \lambda w$ for some $\lambda \in \mathbb{C}^n$. However,

$$Aw = uv^H w = \langle w, v\rangle u.$$

Thus the only possible non-zero eigenvectors of A are the multiples of u, with eigenvalue $\lambda = \langle u, v\rangle$. If $\langle u, v\rangle = 0$, the matrix A does not have any non-zero eigenvalues. (We also note that each vector orthogonal to v is an eigenvector of A with eigenvalue 0.)

In order to determine the Jordan normal form of A, we have to distinguish between the two cases $\langle u, v\rangle = 0$ and $\langle u, v\rangle \neq 0$. Moreover, we note that the matrix A has rank one (its image is just the span of u), and thus also its Jordan normal form necessarily has rank 1.

- If $\lambda = \langle u, v\rangle \neq 0$, then the only possible rank one matrix in Jordan normal form where one eigenvalue is equal to λ is the matrix
$$J = \begin{pmatrix} \lambda & 0 & \dots & \dots & 0 \\ 0 & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & \dots & 0 & 0 \end{pmatrix},$$
which therefore has to be the Jordan normal form of A.
- If $\lambda = \langle u, v\rangle = 0$, then the Jordan normal form has to be a rank one matrix where all diagonal elements are equal to 0. The only possibility is the matrix
$$J = \begin{pmatrix} 0 & 1 & 0 & \dots & \dots & 0 \\ 0 & 0 & 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & 0 & 0 & 0 \end{pmatrix}.$$

b) A vector w is an eigenvalue of a matrix A with eigenvector μ, if and only if $Aw = \mu w$. This, however, implies that
$$(A + \lambda\,\mathrm{Id})w = Aw + \lambda w = (\mu + \lambda)w,$$
that is, w is also an eigenvector of $(A + \lambda\,\mathrm{Id})$ with eigenvalue $\mu + \lambda$.

c)

We note that we can write
$$A = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} = \mathrm{Id} + vv^T$$
with
$$v = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}.$$

As a consequence, the matrix A has one eigenvalue $1 + \langle v, v \rangle = 5$, and a triple eigenvalue 1. Moreover, v is an eigenvector for the eigenvalue 5, and all vectors normal to v are eigenvectors for the eigenvalue 1. As an example, we therefore obtain the eigenbasis

$$w_1 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \qquad w_2 = \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \qquad w_3 = \begin{pmatrix} 0 \\ 1 \\ -1 \\ 0 \end{pmatrix}, \qquad w_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ -1 \end{pmatrix}.$$

(Obviously, the vectors w_2, w_3, and w_4 are linearly independent.)

(If one actually wants an orthogonal eigenbasis, then one could choose $w_3 = (1, 1, -1, -1)^T$ instead.)

$\boxed{5}$ 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

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

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)
```