# **Elements of Mathematics**

Sheet 01

Due date: **XXX** 

Name:		
Matriculation number:		

Task:	1	2	3	4	Total	Grade
Points:						
Total:	6	6	12	10	34	_

Ex 1 Linear Algebra 6

### A matrix as a Linear Function

Let  $A \in \mathbb{F}^{m \times n}$  be a matrix. Then consider the mapping  $f_A \colon \mathbb{F}^n \to \mathbb{F}^m, x \mapsto Ax$ .

1. Show that

$$f_A(\lambda x + y) = \lambda f_A(x) + f_A(y),$$

 $\text{ for all } x,y\in\mathbb{F}^n \text{ and } \lambda\in\mathbb{F}.$ 

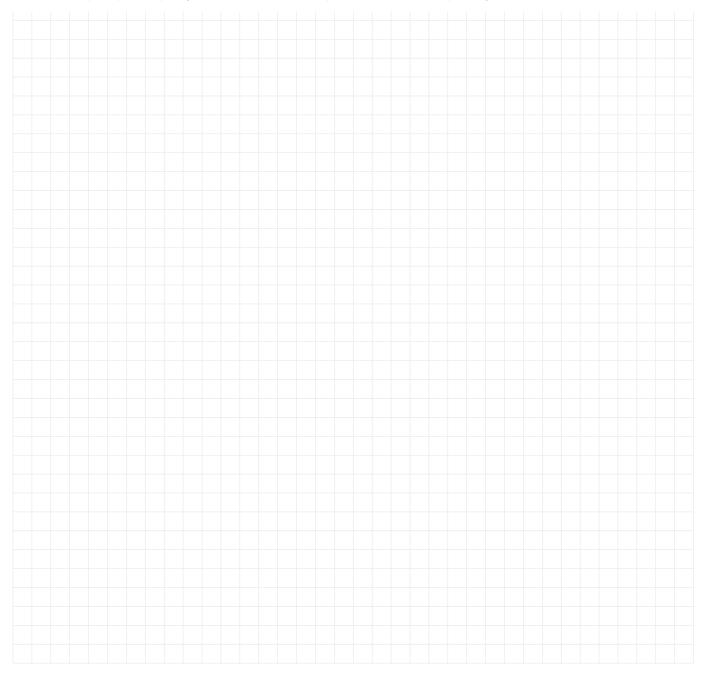
*Hint:* A vector is a matrix with just one column, so you can make use of the computation rules for matrices given in the lecture notes.

Remark: Functions satisfying this property are called linear.

2. Use this fact to show the following equivalence:

$$\ker(A):=\{x\in\mathbb{F}^n\colon Ax=0\}=\{0\}\quad\Leftrightarrow\quad f_A(x)=f_A(y)\text{ implies }x=y,$$
 (i.e.,  $f_A$  is an injective mapping).

*Hint:* Split up the equality  $\Leftrightarrow$  into  $\Rightarrow$  and  $\Leftarrow$  and prove each of them separately.



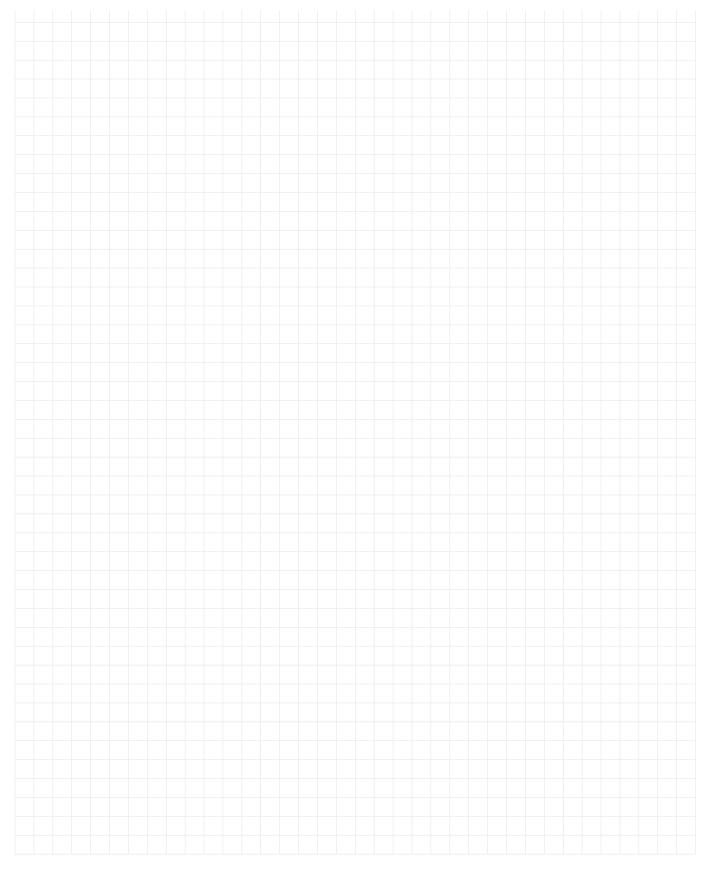


Ex 2 Linear Algebra 6

## The Subspaces Kernel and Image

1. Let  $A \in \mathbb{F}^{m \times n}$ . Show that  $\ker(A)$  and  $\operatorname{Im}(A)$  are subspaces of  $\mathbb{F}^n$  and  $\mathbb{F}^m$ , respectively.

 $2. \ \,$  Construct two example matrices and consider their kernel and image.





Ex 3 Linear Algebra 12

## Rank/Image and Nullity/Kernel

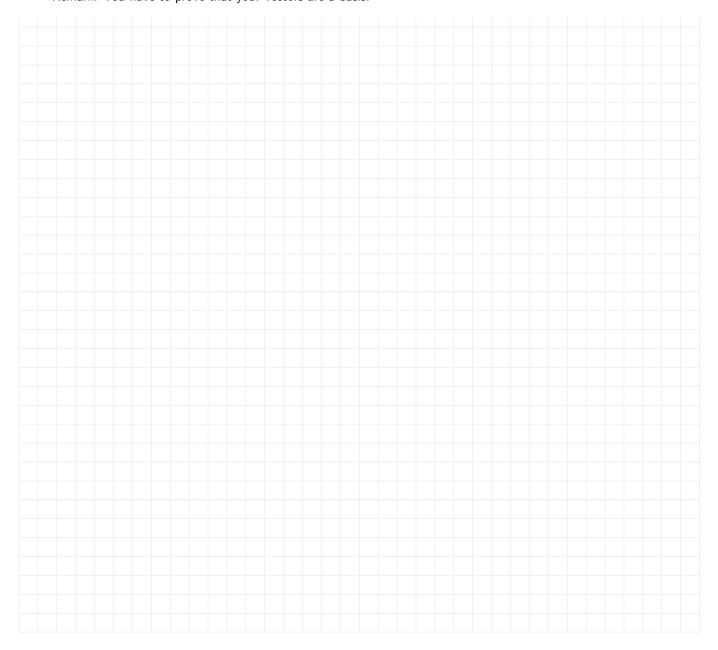
Consider the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$

the column vector  $\mathbf{1} := \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$  (i.e., a  $(3 \times 1)$  matrix) and the row vector  $\tilde{\mathbf{1}} := (1 \ 1 \ 1)$  (i.e., a  $(1 \times 3)$  matrix).

- 1. Show that  $A = 1\tilde{1}$ .
- 2. Find two distinct nonzero vectors x and y, so that Ax = 0 and Ay = 0.
- 3. How does the image Im(A) look like? First draw a picture. Then find a basis of Im(A) to determine the rank of the matrix.
- 4. How does the kernel  $\ker(A)$  look like? First draw a picture. Then find a basis of  $\ker(A)$  to determine the nullity of the matrix.

Remark: You have to prove that your vectors are a basis.





#### The Matrix-Vector Product

Implement a function that takes as input a matrix  $A \in \mathbb{R}^{m \times n}$  and a vector  $x \in \mathbb{R}^n$  and then returns the matrix-vector product Ax.

- 1. Implement the following four ways:
  - a) **Dense:** Input expected as numpy.ndarray:

Assume that the matrix and the vector are delivered to your function as numpy.ndarray.

- i. Implement the matrix-vector product "by hand" using for loops, i.e., without using numpy functions/methods.
- ii. Implement the matrix-vector product using A.dot(x), A@x, numpy.matmul(A,x) or numpy.dot(A,x).
- b) **Sparse:** Matrix expected in CSR format:

Assume that the matrix is delivered to your function as  $scipy.sparse.csr\_matrix$  object. The vector x can either be expected as numpy.ndarray or simply as a Python list.

- i. Access the three CSR lists via A.data, A.indptr, A.indices and implement the matrix-vector product "by hand" using for loops.
- ii. Implement the matrix-vector product using A.dot(x) or A@x .
- 2. **Test** your four different routines from above on the following matrix  $A \in \mathbb{R}^{n \times n}$  with constant diagonals given by

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & -1 & 2 & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix} \in \mathbb{R}^{n \times n}$$

and the input vector

$$x = (1, \dots, 1)^{\top} \in \mathbb{R}^n$$
 (you can use: x = numpy.ones(n)).

- a) Determine how  $b:=A\cdot x\in\mathbb{R}^n$  looks like in this example in order to facilitate a test.
- b) Test whether your four routines compute the matrix-vector product correctly by checking  $A \cdot x = b$ .
- c) Use different values for the dimension n (especially large  $n \geq 10^5$  note that you may exceed your hardware capacities for the dense computations).

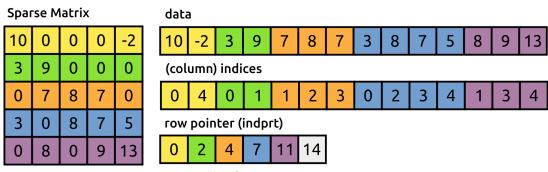
Remark: The matrix has "2" on the main diagonal and "-1" on the first off-diagonals.

- 3. For all cases:
  - a) **Memory:** What is the number of Gbytes needed to store an  $m \times n$  array of floats? Print the number of Gbytes which are needed to store the matrix in all cases.

*Hint:* A number implemented as float in Python implements double precision and therefore needs 64 Bits of storage. For a numpy.ndarray you can type A.nbytes and for the scipy.sparse.csr\_matrix you can type A.data.nbytes + A.indptr.nbytes + A.indices.nbytes.

b) **Computation times:** Measure the time which is needed in each case to compute the matrix-vector product for a random input vector x = numpy.random.rand(n).

*Hint:* In the IPython shell you can simply use the *magic function* %timeit to measure the time for a certain operation. For example, you can type %timeit pythonfunction(x). Alternatively you can use the package timeit.



Example of a Matrix in CSR Format



