# *Exercise 3*

## *Linear algebra*

*IAML 2020*

This week, the focus is on getting you up to speed working with linear algebra. You will both be performing pen-and-paper exercises and coding exercises using Numpy. We believe that proficiency in both gives you the best opportunities for understanding and applying essential linear algebra skills in the course and elsewhere.

Before starting on the mathematics however, we provide a more in-depth introduction to Matplotlib, which is indispensable when working with data in Python.

**Exercise 3.1**

### *(**Jupyter**) Introduction to Matplotlib*

*This exercise is written as a Jupyter notebook. It is saved as* `Matplotlib.ipynb` *in the exercise material.*

This exercise gives you a thorough introduction to **Matplotlib**, which is a library you will be using throughout the course. It is the standard for high-quality plots in Python and has a very stable API. Later, we will also introduce **Seaborn**, a library that exposes a more high-level plotting API for easy creation of certain plot types. Matplotlib supports several desktop GUI environments as well as Jupyter notebooks.

We cover the following topics in this exercise:

- Basic plotting of functions and data points.
- Styling of plots.
- Combining multiple plots.
- Saving plots to disk.

**Exercise 3.2**

### *Pen and paper exercises*

These exercises should be completed using only pen and paper. The goal is to test some of the basic linear algebra skills needed for proper understanding of the course material.

Task 1: *Book exercises*

Solve the following exercises from the **[ST]** book:

*Chapter 1* **1.1.1**, **1.1.2**, **1.2.1**, **1.2.2** (skip the question on Schwartz inequalities), **1.2.8** (skip c)

*Chapter 2* **2.4.1**, **2.4.3**, **2.4.15**, **2.5.15**$^{extra}$, **2.7.2**

*Chapter 3* **3.4.10** (only do a, b, f), **3.4.1**, **3.4.11**, **3.4.18**

Task 2: *Multiplication with vectors and matrices*

Calculate the following expressions by hand:

(a) $\begin{bmatrix} 2 & 4 \\ 9 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = ?$

(b) $\begin{bmatrix} 1.1 & 1.2 \\ 1.3 & 1.4 \\ 1.5 & 1.6 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \end{bmatrix} = ?$

(c) $\begin{bmatrix} 1 & 0 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = ?$

(d) $\begin{bmatrix} 5 & 5 \end{bmatrix} \begin{bmatrix} 3 \\ 6 \end{bmatrix} = ?$

(e) $\begin{bmatrix} 5 \\ 5 \end{bmatrix} \begin{bmatrix} 3 & 6 \end{bmatrix} = ?$

**Exercise 3.3**

## *(**Jupyter**) Introduction to linear algebra in Numpy*

*This exercise is written as a Jupyter notebook. It is saved as* `LinearAlgebra.ipynb` *in the exercise material.*

This exercise introduces basic linear algebra operations in Numpy as well as how to use it to solve systems of linear equations and for performing linear regression using least squares. Your goal should be to familiarise yourself with the theoretical linear algebra concepts and learn some standard applications in Numpy. We cover the following topics:

- Basic matrix operations (elementwise operations, transpose, multiplication, inverse)
- Properties of matrix multiplication and inversion
- Linear equations in matrix form
- Solving linear equations using matrix inverses