Training Assignment 06

NUMA01: Computational Programming with Python

Malin Christersson, Robert Klöfkorn

This exercise trains knowledge about vectors and arrays, i.e. the basics from Chapter 4 and 5 in the course book.
This assignment has 8 tasks.

Warming-up Exercises

Task 1

Read Chapter 4 of the course book.

Task 2

Implement the following code:

```
from numpy import *
v = array([1., 2., 3])
M = array([[1., 2, 3.],
           [4., 5., 6],
           [7., 8., 9]])

# what does this line do?
z = M @ v
print(z)
```

Exercises

Task 3

Consider a square matrix A (number of rows is the same as columns).

A is symmetric $\iff A = A^T$, i.e. if $a_{ij} = a_{ji}$ for indices i, j. Accordingly, A is skew-symmetric $\iff -A = A^T$.

Write a function, which takes a matrix as parameter. It should check if this matrix is symmetric. The function should return 1 for a symmetric matrix, -1 for a skew-symmetric matrix and 0 otherwise.
Test your function.

Task 4

Write a function, which takes two vectors as parameters. It should check if these vectors are orthogonal. If they are orthogonal it should return `True`, otherwise `False`.
Don't forget to provide your function with a `docstring`. If you don't know what a `docstring` is, consult the book.

Test your function.

Task 5

Write a function, which takes a vector as parameter and which returns the corresponding normalized vector, i.e. $\frac{x}{\|x\|}$. Write two variants of this program: one in which you compute the norm (use the 2-norm) of the vector by yourself and another, which uses the function `norm` from the module `numpy.linalg`.

Recall that to be able to use Numpy's norm-function you must have the line

```
from numpy.linalg import norm
```

at the start of your program. Note that (of course) you cannot then call your own function `norm` too. If this is inconvenient, one can use

```
from numpy.linalg import norm as scnorm
```

to give Numpy's `norm`-function the new name `npnorm`.

Task 6

Show experimentally that the inverse of a rotation matrix is its transpose.
Hint: B is the inverse of A if $AB = BA = \mathrm{I}$, the identity matrix.

Note, in 2D a rotation matrix has the form

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}$$

where α can be any angle.

Extra Task 7 (in case you want an extra challenge)

(If you don't know eigenvalues (yet) skip this task for now) Construct a 20×20 matrix with the value 4 on its diagonal and the value 1 on its sub- and super-diagonal. The rest of the matrix is zero. Compute its eigenvalues. (Use the function `eig` from the module `numpy.linalg`). You might also want to check the function `diag` for this task

Extra Task 8 (in case you want an extra challenge)

(If you don't know eigenvalues (yet) skip this task for now) Change in the above task the matrix in such a way that all the elements of its subdiagonal instead have the value -1. How are the eigenvalues affected by this change?