VRIJE UNIVERSITEIT BRUSSEL

Third assignment: matrices

Basic Techniques for Computer Simulations Simulations WPO | 25 March 2025

1 Instructions

In order to solve the assignment, you have to answer all questions included in the problem set.

Provide your answers as **one Python file** called `wpo3.py` that has to print all solutions to the questions and produce all plots requested in the problem set. The file has to be uploaded on Canvas before the deadline of the assignment.

Solutions to the questions have to be defined in the file as variables named `sol_x` where `x` is the number of the question. The solution file has to comply with the following format:

```python

import numpy as np

def main():
    """ Your names """
    sol_1 = square(2)
    print(sol_1)
    sol_2a, sol_2b = logarithm(1000)
    print(sol_2a, sol_2b)

def square(x):
    """ Ex1 : Square of x """
    return x**2

def logarithm(x):
    """ Ex2 : Calculate log base ten and log base e """
    return np.log10(x), np.log(x)

if __name__=='__main__':
    main()
```

Listing 1: Example solution file format

2 Problem sets

Three objects of mass m_i are connected with four springs with spring constant k_i between two fixed walls. The following homogeneous set of equations can be derived for this system:

$$
\begin{aligned}
\left(\frac{2k}{m_1} - \omega^2\right) X_1 - \frac{k}{m_1} X_2 &= 0 \\
-\frac{k}{m_2} X_1 + \left(\frac{2k}{m_2} - \omega^2\right) X_2 - \frac{k}{m_2} X_3 &= 0 \\
-\frac{k}{m_3} X_2 + \left(\frac{2k}{m_3} - \omega^2\right) X_3 &= 0,
\end{aligned}
$$

where X_i is the amplitude of the oscillation of object i. This is an eigenvalue problem of the form $(A - \lambda I)x = 0$, where the eigenvalue λ corresponds to the angular frequency ω^2.

If all masses $m = 1\,\mathrm{kg}$ and all the spring constants $k = 20\,\mathrm{N/m}$, form matrix A and:

1. Approximate and print the highest eigenvalue and corresponding eigenvector of A, using your own code for the *power method*;

2. Approximate and print the lowest eigenvalue and corresponding eigenvector of A with the *inverse power method*, using the Python function you wrote in Question 1;

3. By using *numpy.linalg.eig*, calculate and print the U, Σ, and V matrices of the *Singular Value Decomposition* (SVD) of A;

4. Approximate and print all eigenvalues of A with your own code for the QR method (*QR decomposition*).