COMPUTATIONAL PHYSICS
Second assignment WS 2023/2024

Deadline: 30th of November 2023
Dates for the assessment discussions: 4th/5st of December

The aim of this assignment is to get familiar with Fourier analysis, linear fitting and the solution of eigenvalue equations.

Note: The assignments contain questions and tasks, labeled (a), (b), etc. For some of these tasks, you are supposed to create and hand in one of the following:

- 💻 a script or a part of a script written in the computer language of your choice;
- 📈 a figure with legends and axis titles.

Other questions primarily serve to guide you through the exercises; you do not need to submit the answers in writing. However, you may be asked these or similar questions during the assessment discussions.

Note: Many tasks in the exercises can be performed using an existing library or toolbox. In line with the aim of the course, however, programming your own analysis tools is encouraged. If libraries or toolboxes are used, you are expected to be able to explain in detail how these algorithms work.

Note: When collaborating in groups of up to 3 students, you are allowed to hand in identical code. Please list all collaborators, for example in the header.

Note: The scripts in the programming exercises should be considered to be intended for general use, and a corresponding coding style will be appreciated. For example, using input parameters as variables, clear presentation of input and output, naming of variables and/or comprehensive comments in the script are among the grading criteria.

1 Frequency analysis of a one-dimensional data set

You will implement a discrete Fourier transform and analyze a simple sound file. A test sound file called `single_tone.txt` is provided.[1] The sample rate is 44100 Hz. Note that the file contains a stereo recording; you can use either of the two channels. To actually hear the sound you can use the Matlab function `sound` or play the `.ogg` file using a music player of your choice. Earphones are recommended.

(a) 💻 Write a code that implements the discrete Fourier transform,

$$Y_k = \sum_{j=0}^{n-1} y_j \exp\left(i\,\frac{2\pi kj}{n}\right), \quad k = 0, \ldots, n-1. \tag{1}$$

Your result should be identical to the result of a fast Fourier transform. Test your code by comparing the Fourier transform of the test file with the results of a standard implementation of the fast Fourier transform in Matlab, Python or a C library of your choice.[2] Note that different definitions of the Fourier transform are used for some FFT implementations. In particular, the sign of the exponent, the definition of the frequency and the prefactor may vary.

(b) 📈 Select a segment of the time series of length m, and modify your code to determine the time it takes to compute the discrete Fourier transform of the segment. Make a plot of the computation time as a function of m and interpret your results.

Clearly, the scaling of the computation time makes the discrete Fourier transform prohibitively expensive to use for large data sets. For the rest of the exercise, you may use a package implementation of the FFT.

(c) 📈 The power spectral density of a signal is defined as

$$S_{yy}(k) = \frac{1}{n^2}\left|\sum_{j=0}^{n-1} y_j \exp\left(i\,\frac{2\pi kj}{n}\right)\right|^2, \quad k = 0, \ldots, n-1. \tag{2}$$

Plot the power spectral density as a function of the frequency in Hz, properly labeling the axes.

In addition to a sinusoidal wave at its so-called fundamental frequency, a perfect tone contains waves at a number of higher harmonics, whose frequencies are integer multiples of the fundamental frequency.

[1] The same data is provided in the `.ogg` format. In Matlab, this format can be imported using the `audioread` function, which gives the sample rate as a second output argument.

[2] To enhance performance it is wise to extend the file length with zeros up to a power of two.

(d) The test file is a recording of a single note. What is the fundamental frequency of the note? What is the name of the note?[3]

(e) 💻 What would be the Fourier representation of a perfect tone? How can the frequency spectrum be used to approximate the test file using a limited number of frequencies? Write a script that creates a sound file containing an approximation of the test file using only 3–6 distinct frequencies.[4]

2 Contagious spreading

We consider the dynamics of some contagious substance – a rumor, a virus or a genetic mutation in crops – in a population. The fraction of the population affected as a function of time is denoted $y(t)$. A test data set of a measured $y(t)$ is provided in the file `time_evolution.txt`. The first column contains t_i and the second column contains the corresponding y_i for $i = 1 \dots N$. You will calculate a fit of this data and extract the parameters best describing the curve.

A general linear fitting routine uses a sum of basis functions $\phi_j(t)$,

$$f(t; \boldsymbol{a}) = \sum_{j=1}^{m} a_j \phi_j(t), \tag{3}$$

with a_j being the fit parameters and $f(t; \boldsymbol{a})$ being the fit function. The minimization of the mean square error can be written as the set of linear equations $A\boldsymbol{a} = \beta$, where the matrix A is a function of the basis functions at the abscissae t_i of the data set, and β being a vector depending on the ordinates y_i.

(a) Write down the elements of the matrix A and the vector β as a function of y_i and $\phi_j(t_i)$.

(b) 💻 The equation $A\boldsymbol{a} = \beta$ can be solved using LU or QR decomposition or Gauss-Seidel iteration. Write a code that uses either of these algorithms to calculate the vector $\boldsymbol{a}$ for an $m \times m$ matrix A and a corresponding vector β.

(c) 💻 Use a polynomial fit as a first guess. To do so, use the basis functions $\phi_j(t) = t^{j-1}$. Write a fitting routine to fit the test data with a polynomial with m terms.

(d) 📈 How many terms do you need to capture the shape of the curve? Make a plot using a few different values of m and interpret your results.

To improve the fit to the data, we can write down a differential equation that describes the underlying process,

$$\frac{dy}{dt} = \alpha \left(1 - y(t)\right) y(t), \tag{4}$$

with $\alpha < 1$ being a real-valued constant.

[3] See a reference table for the frequencies of notes: `http://pages.mtu.edu/~suits/notefreqs.html`

[4] Useful functions in Matlab: `fft`, `ifft`, `audiowrite`.

(e) Verify that the solution to Eq. (4) is given by

$$y(t) = \frac{1}{1 + c\exp(-\alpha t)}, \tag{5}$$

with c being an integration constant.

(f) Eq. (5) exhibits a nonlinear dependence on the fitting parameters α and c. Show that with a transformation of the input data $y(t)$, the equation can be approximated as a linear combination of the basis functions $\phi_j(t) = 1 - \exp(-t/j)$. How many basis functions do you need, and for which values of j? Express the fit parameters $\boldsymbol{a}$ in terms of α and c.[5]

(g) 💻 Modify your fitting routine to use the new set of basis functions. Determine the parameters α and c including their standard deviation. For the calculation of the covariance matrix you are allowed to use a package implementation of matrix inversion.

(h) 📈 Make a plot of the fit function together with the measured data $y(t)$.

3 Normal mode analysis of a DNA molecule

Given is the equilibrium structure $\boldsymbol{R}^0$ of a molecule as obtained from an x-ray diffraction experiment, consisting of the positions of n atoms. The displacement of the atoms from this equilibrium structure, $\boldsymbol{R} - \boldsymbol{R}^0$ is denoted by the generalized coordinates $\boldsymbol{q} = \{q_1, q_2 \dots q_n\}^T$, with q_i being a vector and i being the index of the atom. In this exercise, we use the structure of a DNA molecule, as shown in Fig. 1. The aim of the exercise is to estimate the dynamics of this system at different frequencies by approximating the interaction potential between the atoms by harmonic functions. To this end, we expand the interaction potential around the equilibrium structure

$$V(\boldsymbol{q}) = V(0) + \sum_i \left(\frac{\partial V}{\partial q_i}\right)_0 q_i + \frac{1}{2}\sum_{ij}\left(\frac{\partial^2 V}{\partial q_i \partial q_j}\right)_0 q_i q_j + \dots \tag{6}$$

The first term of this expression is a constant that may be set to zero and the second term equals zero at equilibrium. Therefore, to second order, the potential energy is given by the sum of pairwise potentials,

$$\begin{aligned} V(\boldsymbol{q}) &= \frac{1}{2}\sum_{ij} H_{ij} q_i q_j \\ &= \frac{1}{2}\boldsymbol{q}^T H \boldsymbol{q}, \end{aligned} \tag{7}$$

[5] Hint: expand the functions $f(t)$ and $y(t)$ and equate the different orders of t. Furthermore, because $\alpha < 1$, $\alpha > 1/j$ for some threshold value of $j > 1$.

with H being the Hessian matrix, containing all second derivatives of the potential. Similarly, the kinetic energy can be written as

$$T(\dot{\boldsymbol{q}}) = \frac{1}{2}\dot{\boldsymbol{q}}^T M \dot{\boldsymbol{q}}, \tag{8}$$

with M being the diagonal matrix containing the masses of the atoms. The equations of motion are then given by

$$M\ddot{\boldsymbol{q}} + H\boldsymbol{q} = 0. \tag{9}$$

By transforming to mass-weighted coordinates $\boldsymbol{r}$ and the stiffness matrix K,

$$\boldsymbol{r} = M^{\frac{1}{2}}\boldsymbol{q} \quad \text{and} \quad K = M^{-\frac{1}{2}} H M^{-\frac{1}{2}}, \tag{10}$$

we find the equation

$$\ddot{\boldsymbol{r}} = -K\boldsymbol{r}. \tag{11}$$

The solution is given by

$$\boldsymbol{r}(t) = \boldsymbol{v}\exp(-i\omega t), \tag{12}$$

where $\boldsymbol{v}$ and ω satisfy the eigenvalue equation $K\boldsymbol{v} = \omega^2\boldsymbol{v}$.

Gaussian network model

We will construct an isotropic network model, which will allow us to analyze the relative motion of the atoms, but not the directions. To construct this so-called Gaussian network model, the atomic pair potential is given by

$$V(\boldsymbol{q}) = \begin{cases} \sum_{ij} \frac{k}{2}(q_i - q_j)^2 & \text{for} \quad |R_i^0 - R_j^0| < R_{cut} \\ 0 \quad \text{otherwise.} \end{cases} \tag{13}$$

Inserting this expression into the equation for the Hessian matrix H yields a contribution of k to H_{ii} and H_{jj}, and a contribution of $-k$ to H_{ij} and H_{ji}, if the distance between the equilibrium positions of atom i and atom j is less than R_{cut}.

(a) 💻 Import the coordinates $\boldsymbol{R}^0$ and atomic masses m of the double-stranded DNA molecule from the file `xyzm_dna.txt`. The file is a comma-separated table containing the coordinates $R_i^0 = x_i, y_i, z_i$ in Å and mass m_i of each atom i. The hydrogen atoms are not included. The first half of the coordinates belongs to the first DNA strand, the second half to the other strand. Write a code to construct the Hessian matrix H using a cutoff distance of $R_{cut} = 5$ Å and a spring constant $k = 1$.

Power method[a]

We assume that the matrix A has a dominant eigenvalue with corresponding dominant eigenvectors. Then we choose an initial approximation $\boldsymbol{x}^{(0)}$ of one of the dominant eigenvectors of A. This initial approximation must be a nonzero real-valued vector. Finally, we form the sequence given by

$$\boldsymbol{x}^{(p)} = A\boldsymbol{x}^{(p-1)} \tag{14}$$

for $p = 1 \ldots n$. For large values of n, this sequence converges to the dominant eigenvector of A. The corresponding eigenvalue can be found using the Rayleigh quotient

$$\lambda = \frac{\boldsymbol{x}^{\dagger(n)} A \boldsymbol{x}^{(n)}}{\boldsymbol{x}^{\dagger(n)} \boldsymbol{x}^{(n)}} \tag{15}$$

Details can be found in the lecture notes.

[a] Also known as von Mises iteration. Richard Edler von Mises (1883 - 1953) was an Austrian mathematician.

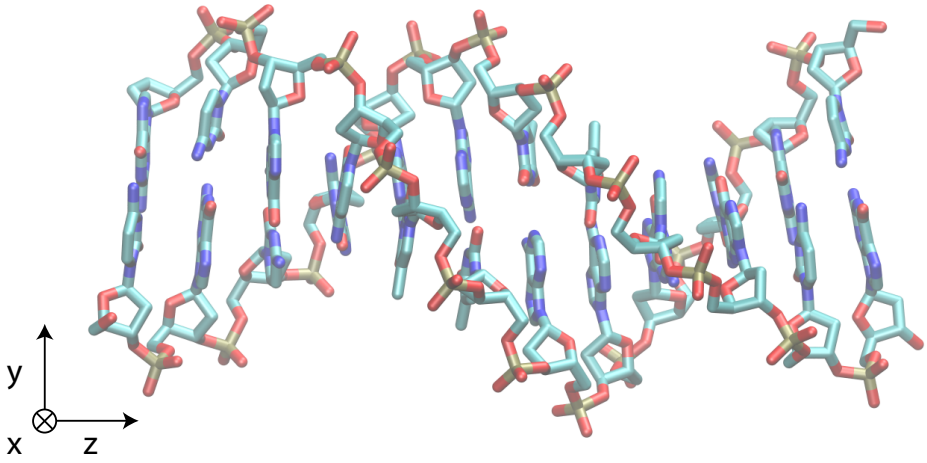

Figure 1: Crystal structure of a piece of double-stranded DNA.

Deflation

Let B be a normal $n \times n$ matrix with eigenvalues $|\lambda_1| < |\lambda_2| < \cdots < |\lambda_n|$ and associated eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots \boldsymbol{v}_n$. Then the matrix

$$C = B - \lambda_n \boldsymbol{v}_n \boldsymbol{v}_n^\dagger \tag{16}$$

has eigenvalues $0, \lambda_1, \ldots, \lambda_{n-1}$ with associated eigenvectors $\boldsymbol{v}_n, \boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_{n-1}$.

(b) 💻 Construct the stiffness matrix K. Which methods can be used to calculate the eigenvalues of K? Use the power method to determine the 10 largest eigenvalues of K and their associated eigenvectors. Note that after having calculated the largest eigenvalue and its eigenvector, deflation is used to get the second largest eigenvector, *etc.* Deflation transforms the n-th eigenvalue to zero.

(c) 📈 Plot the eigenvectors associated with the largest eigenvalues as a function of z for the two DNA strands. Interpret the results.

We are particularly interested in the dynamics at low frequencies, so we are looking for the smallest eigenvalues of K. The eigenvalue of a matrix A which is the closest to a value μ can be calculated by applying the power method to the matrix $(A - \mu I)^{-1}$. We will use a Neumann series to calculate the inverse.

For the following, we need a matrix which has eigenvalues $\lambda = \omega^2$ for which $|\lambda| < 1$. Because the spring constant k has an arbitrary value in our case, we can multiply K by any value to achieve this.

(d) Determine the minimum and the maximum eigenvalues of K using Gershgorin's theorem (for details, see the lecture notes appendix B).

(e) 💻 Implement a function to construct a matrix K' which has $|\lambda| < 1$. Take care that the inequality is strict: $|\lambda| = 1$ is not allowed. Note that you do not need to calculate any eigenvalues to achieve this.

(f) Verify that the smallest eigenvalue of K' can be determined by applying the power method to the matrix $(K' + I)^{-1}$.

(g) 💻 Prove the following equality,

$$\lim_{n\to\infty} \sum_{p=0}^{n} (-K')^p \, (K' + I) = I, \tag{17}$$

and implement a script to calculate $(K' + I)^{-1}$. Note that you can stop the sum if the maximum element (absolute value) of $(K')^p$ is less than a cutoff value, for which you can take 10^{-8}.

(h) 💻 Use the power method to determine the 10 smallest eigenvalues of K' and the associated eigenvectors. Is K' invertible?

(i) 📈 Out of these 10 smallest eigenvalues, plot the eigenvectors associated with the nonzero eigenvalues as a function of the position z along the DNA molecule. Use a separate (sub)plot for every eigenvector and plot the two DNA strands in two distinct colors. Interpret your results in terms of the relative motion of the atoms.

4 Bonus task: anisotropic normal mode analysis

The Hessian matrix can also be constructed with the three cartesian directions separated, while still using harmonic potentials between the atoms. In that case, typically referred to as the anisotropic elastic network model, the matrix for the interaction between the atoms i and j is given by

$$H_{ij} = \begin{bmatrix} \frac{\partial^2 V_{ij}}{\partial x_i \partial x_j} & \frac{\partial^2 V_{ij}}{\partial x_i \partial y_j} & \frac{\partial^2 V_{ij}}{\partial x_i \partial z_j} \\ \frac{\partial^2 V_{ij}}{\partial y_i \partial x_j} & \frac{\partial^2 V_{ij}}{\partial y_i \partial y_j} & \frac{\partial^2 V_{ij}}{\partial y_i \partial z_j} \\ \frac{\partial^2 V_{ij}}{\partial z_i \partial x_j} & \frac{\partial^2 V_{ij}}{\partial z_i \partial y_j} & \frac{\partial^2 V_{ij}}{\partial z_i \partial z_j} \end{bmatrix} . \tag{18}$$

The complete $3n \times 3n$ Hessian is composed of these super-elements.

(a) 💻 Using power iteration, implement a solution algorithm for the 3 largest and 3 smallest eigenvalues of the anisotropic elastic network model.

(b) 📈 Make an animation of the calculated eigenmodes for an arbitrary small-amplitude deviation from the equilibrium configuration of the DNA molecule.