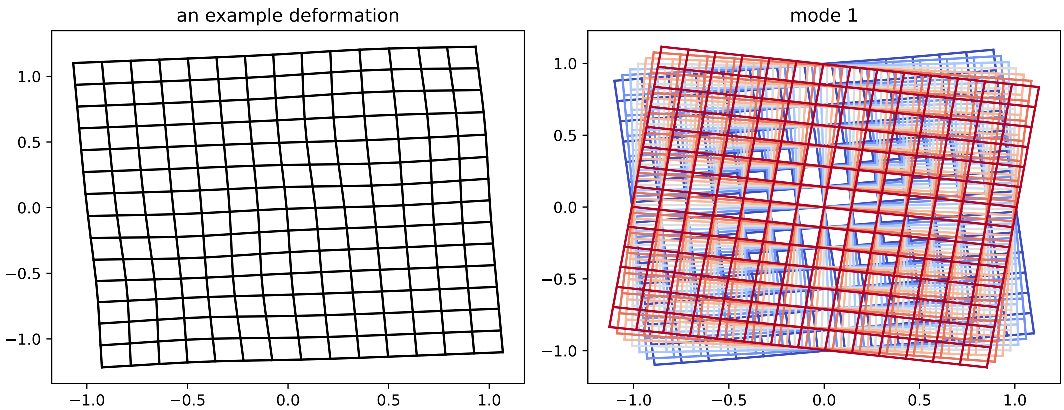
Visualization
Prof. Bernhard Schmitzer, Uni Göttingen, summer term 2023

Problem sheet 4

- *Submission by 2023-07-04 18:00 via StudIP* ***as a single PDF****. Please combine all results into one PDF. If you work in another format (markdown, jupyter notebooks), convert them to PDF before submission.*
- *Use Python 3 for the programming tasks as shown in the lecture. If you cannot install Python on your system, the GWDG jupyter server at* `https://jupyter-cloud.gwdg.de/` *might help. Your submission should contain the final images as well as the code that was used to generate them.*
- *Work in groups of two to three. Clearly indicate names and matrikelnr of all group members at the beginning of the submission.*
- **The third exercise gives bonus points that can help to make it towards exam admission.**

Exercise 4.1: PCA on deformations.
Consider the square $[-1,1] \in \mathbb{R}^2$, discretized by a uniform grid with $\texttt{nPts} = 15$ points along each axis. The file `UVData.npz` contains an array `UVData` with dimensions $\texttt{nMaps} \times 2 \times \texttt{nPts} \times \texttt{nPts}$. For each `i` in `range(nMaps)`, the pair $\texttt{U,V} = \texttt{UVData[i]}$ encodes a deformation of this grid, where `U` contains the horizontal and `V` the vertical coordinates. As reference, `UVData.npz` also contains the arrays `X` and `Y` that contain the original horizontal and vertical positions of the undeformed grid points.

1. Display the first five deformations, e.g. as in the figure below, left. Briefly comment on what types of deformations you can see.
2. Now apply PCA to the collection of deformations. Interpret each pair $(\texttt{U},\texttt{V})$ as a one-dimensional vector of length $2 \cdot \texttt{nPoints}^2$, i.e. interpret `UVData` as two-dimensional array of dimension $\texttt{nMaps} \times (2 \cdot \texttt{nPoints}^2)$ and perform PCA on it, as in shown in the lecture. Visualize the eigenvalues and their cumulative sum. How many modes / eigenvalues are needed to capture at least 99% of the variance?
Hint: Do not forget to center the data before applying PCA.
3. Now try to visualize the 'meaning' of the first three modes, similar to the example from the lecture with the Gaussian curves. I.e. add each eigenvector to the mean, scaled according to the root of the variance that it captures, and visualize the corresponding deformations. For the first mode result could look as in the figure below, right. What is the interpretation of the first three modes?
Hint: In the course of this you need to undo the 'flattening' that was applied before PCA.



Exercise 4.2: damped pendulum.
Consider a mathematically idealized damped pendulum with angle θ and angular velocity ω with the equation of motion given by

$$\begin{pmatrix} \partial_t \theta(t) \\ \partial_t \omega(t) \end{pmatrix} = F \begin{pmatrix} \theta(t) \\ \omega(t) \end{pmatrix} = \begin{pmatrix} \omega(t) \\ -\alpha \cdot \omega(t) - \beta \cdot \sin(\theta(t)) \end{pmatrix}$$

which are to be solved for some initial $(\theta(0), \omega(0))$. Here, $\alpha \geq 0$ is a coefficient describing the strength of friction; β is a coefficient describing the length of the pendulum and the strength of gravity. We set $\alpha = 0.15$, $\beta = 5$.

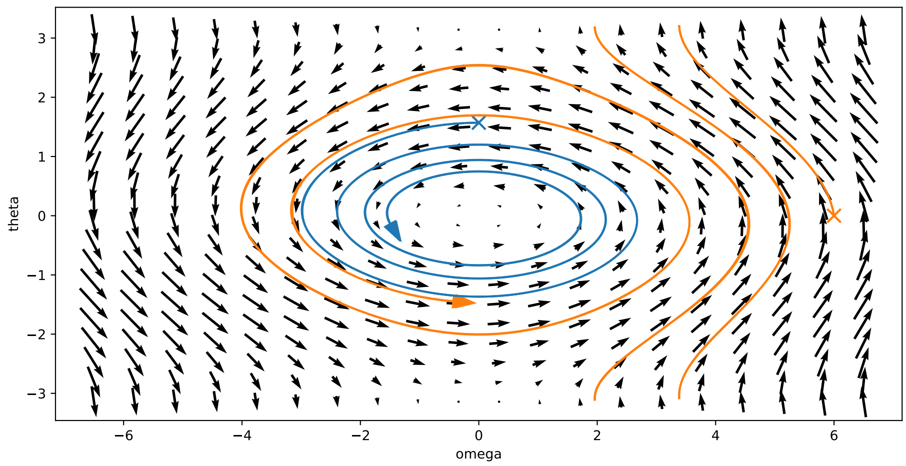
1. Create a quiver plot of the vector field F on $(\theta, \omega) \in [-\pi, \pi] \times [-6.5, 6.5]$.

2. Instead of merely using the streamplot function of matplotlib, we will create our own trajectory plot. For initial configurations

$$\begin{pmatrix} \theta(0) \\ \omega(0) \end{pmatrix} = \begin{pmatrix} \pi/2 \\ 0 \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} \theta(0) \\ \omega(0) \end{pmatrix} = \begin{pmatrix} 0 \\ 6 \end{pmatrix}$$

solve the above ODE numerically on $t \in [0, 10]$. This can be done quickly with the function `scipy.integrate.odeint` (see its documentation for the pendulum example) or `scipy.integrate.solve_ivp` and add these trajectories to the quiver plot. Recall what we learned about trajectory plots.
Hint: In the second configuration you must be careful about the cyclic structure of θ.

The result could look as follows:



Here the two colored lines indicate the two trajectories, initial points given by crosses and final positions by dots.

3. Briefly answer the following questions: What is the major difference between the two trajectories and what does it mean for the pendulum's behaviour in each case? Qualitatively, how would the trajectories change if α were increased?

Exercise 4.3: multi-dimensional scaling (7 bonus points). The file `mds.npz` contains the array `d2` of dimension `nPts` $\times$ `nPts` giving the matrix of squared distances of a point cloud of `nPts` $= 500$ points, originally embedded in $\mathbb{R}^3$.

1. Use MDS as in the lecture to reconstruct the embedding. Visualize the best approximate embeddings into 2 and 3 dimensions. Describe in a few words the 'shape' of the 3d embedding.

2. How can you tell that the original point cloud must probably have been three dimensional? The original point cloud had features on three different length scales (things like 'length' or 'period of some oscillation'). What are these features? Which feature is missing in the approximate 2-dimensional embedding?

3. Use `sklearn.manifold.MDS` to get approximate 2- and 3-dimensional embeddings. Briefly comment on whether they are comparable.
   Hints: Use the keyword `dissimilarity='precomputed'` and use the (entry-wise) square root of `d2` as proxy for dissimilarity.