Data Visualization and Analysis

BINF4245

Prof. Dr. Renato Pajarola

Exercise and Homework Completion Requirements

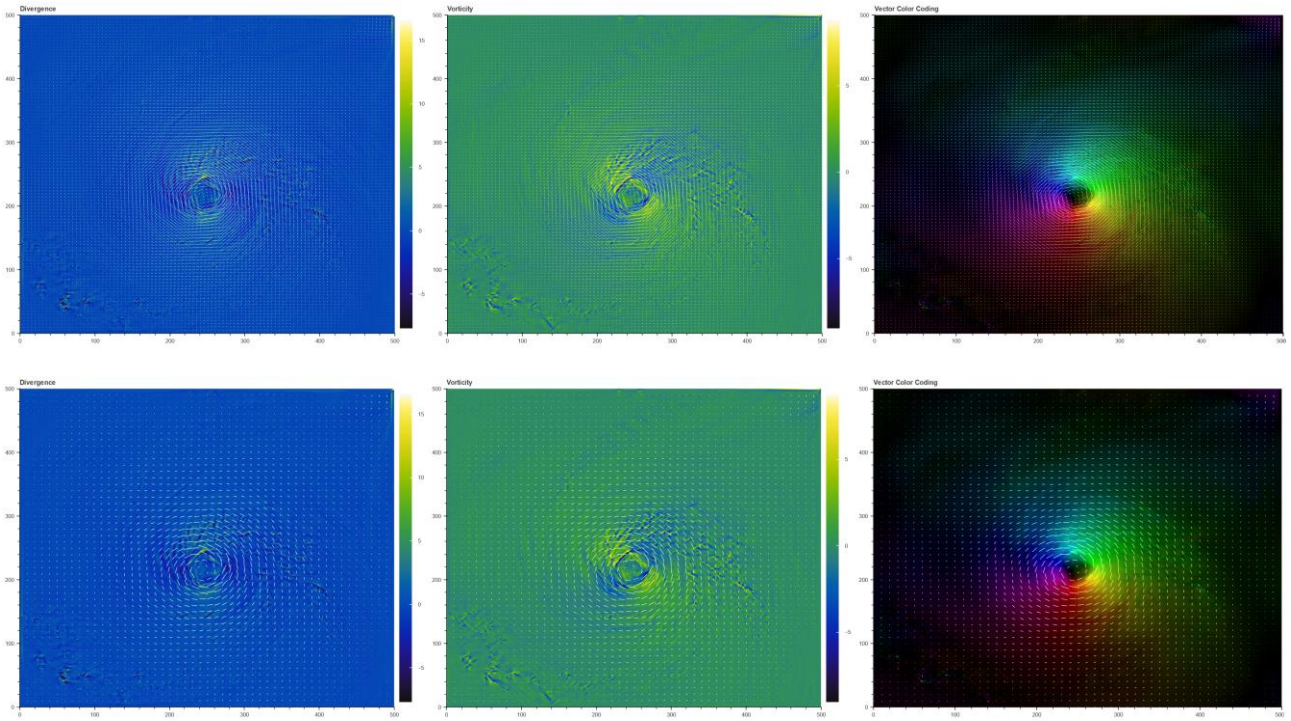
1. Every **solved** exercise will earn you points, up to 15 in total.
   - *Late submissions (up to one day) will result in "-1" point.*
   - *If the solution works as expected, full points are awarded, else partial points are assessed as outlined in the code skeleton.*
2. The four exercises give rise to the following point distribution: 2 – 3 – 5 – 5.
   - *A **minimum of 7 points** from all four exercises must be achieved to pass the module. Failure to achieve this minimum will result in a failing grade for the entire module.*
   - *Thus at least two exercises must be correctly solved, and one must be from the more advanced ones.*
3. We give **bonus points** for students who have completed more than 8 points from all the exercises.
   - *Thus **7 points** from the exercises is required, **8 points** is still normal passing, and **9 and above** would give 1 or more extra bonus points.*
   - *Only the bonus points can and will be added directly to the final exam.*
4. Do not copy assignments, tools to detect copying and plagiarism will be used.
   - *The exercise results are an integral part of the final course grade and therefore the handed in attempts and solutions to the exercises **must be your personal work**.*

Submission Rules

- Submitted code must run without errors using the indicated Python environment, using the included libraries, packages, and frameworks.
- The whole project source code must be zipped and submitted before the given deadline, including the output results (saved in .html file or as a screenshot picture).
- Submit your .zip archive named *dva_ex1_Firstname_Lastname_MATRIKELNUMBER.zip* (e.g. dva_ex1_Hans_Muster_01_234_567.zip) through the OLAT course page.
- You zip file should contain:
  - Your working code as dva_ex4_Hans_Muster.py file
  - A screenshot or export of your tool as .jpg, .png, .pdf or .html
  - An readme.txt if you needed additional libraries and good reasons why you needed them.
- **Deadline is Sunday, 5 June 2022 at 23:59h**

Exercise 4

In this exercise, we will get some practical experience with the processing and visualization of a 2D vector field dataset, which is extracted from a multivariate dataset that simulates a hurricane from the National Center for Atmospheric Research (NCAR) in the United States. The original data consists of several time-varying scalar and vector variables over large dynamic ranges; more details about it can be found from the following link: http://vis.computer.org/vis2004contest/data.html. In this exercise, we will only deal with the "wind speed" in a certain time step and compute the divergence, curl and vector color mapping for a specific height slice using Python.



Two results generated with different hedgehog plot resolutions. Each showing three plots: Divergence, Vorticity and Vector Color Coding, all overlayed with a hedgehog plot.

Task 1: (1 Point) Calculate the divergence of the wind vector field and visualize it in a plot.

Task 2: (1 Point) Calculate the vorticity of the wind vector field and visualize it in a plot. Use the signed value of the magnitude for the visualization, since this allows the expression of the rotation direction. Have a look at the tutorial slides for more information.

Task 3: (1.5 Point) Calculate the Vector Color Coding of the wind vector field. For the HSV to RGB conversion, either use pythons *colorsys* library or implement it on your own using the following formula:

https://www.rapidtables.com/convert/color/hsv-to-rgb.html

Task 4: (1.5 Point) Calculate the start and end locations of the hedgehog line segments and overlay each plot with them.

For all tasks, additional explanations can be found in the lecture slides about vector fields, and in the skeleton comments. Additionally, I added a short presentation with the most important information about how to calculate the divergence and the vorticity.

Remarks:

- Try to make good use of the hints and references provided in the skeleton code. (**very important**)
- Try to google first for any Python related issues/bugs.
- Due to the special situation, we don't arrange in person meeting in this semester. Please contact me (Gaudenz Halter, halter@ifi.uzh.ch) for technical questions regarding the exercise only as a last resort.
- More than one day late submission will not be accepted and graded.
- If the submitted solution works as expected, full points will be rewarded, else, we follow the points indicated in the template
- **Online office hour** will be on **Friday, 27 May 16:00 – 17:00**. The zoom link will be communicated in that week, prior to the office hour, via email.