Mandatory Exercise TEK5600
Visualization of vector fields

In this exercise, you will visualize two vector fields using two different methods: 1) by a geometric based method using field lines, and 2) by a texture based method using Line Integral Convolution (LIC). In exercise 1, vector fields are to be visualized using field lines. In exercise 2, you are going to use LIC.

The exercises consist of implementing both techniques, comparing the results and finally to summing up the results in a report.

Exercise 1

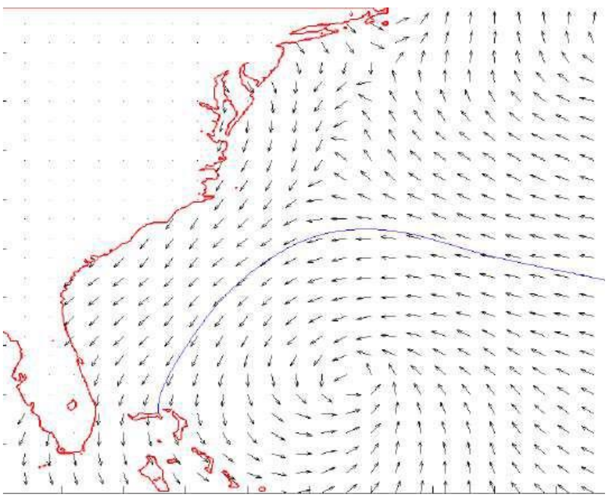
In the first exercise, you are going to do the following tasks:

1. Write a program that reads two vector data sets stored in the HDF5 data format.

2. Make a field line integrator based on both forward Euler and on the 4'th order Runge Kutta integration method. (You will need this for both task 1 and 2). Figure 1 shows a simple field line for one of the data sets.

3. Visualize the two vector fields using field lines. Try out various seed point strategies. Seed points are the starting position or starting point for the field lines to be computed. The visualization result depends strongly on the selection of seed points. Examples of different seeding strategies could be *random based*, *uniform based*, *density based* and *flow feature based*.

   In addition, try out the effect of varying the lengths of the field lines and the effect of changing the accuracy of the integration scheme (first order Euler or fourth order Runge Kutta). The accuracy of the computed field lines will depend on both the integration scheme used as well as the chosen step size. *Hint: The biggest change in result from using integration methods of different accuracy will occur in regions of the flow with high curvature.*



Figur 1: Isabel data visualization using field lines and arrows.

4. Write a report including the following:
    a. Short introduction including the definition of a field line as well as the description of the integration methods.
    b. Describe your selection of seeding strategies.
    c. Compare the selection of seeding strategies and comment on the resulting visualization results.
       What are the differences and which method do you think gives the better result?

Exercise 2

In the second exercise, you are going to visualize the same two vector fields using Line Integration Convolution (LIC). Some of the work done in exercise 1 will be useful for this task. Do the following steps:

1. Introduction - Write a short introduction about LIC by describing the LIC algorithm. Do present the most relevant formulas.

2. Tell short about your LIC implementation.

3. Compare the results obtained by Euler and Runge Kutta integration methods. *(Any thoughts about which method is more sensitive to the accuracy of the integration method – geometric or texture based methods?)*

4. Try out various lengths on the LIC filter core and compare the results. Compare the efficiency.

5. Compare the result from exercise 1 and 2, i.e compare the geometric field lines with LIC textures. Describe the benefits and drawbacks of the two classes of methods.

6. Optional: Try using a higher resolution on the input and output texture rather than just using the same resolution on the textures as is defined by the vector field. *(Line integral Convolution is independent of the resolution of the vector field. As an example, the right image of Figure 2 is created by using a higher resolution of the texture compared to the associated vector field).*

7. Write a short conclusion/summary.

Some information about the data

The two datasets to be used in the exercise is taken from weather simulations. They can be downloaded from the homepage of the course.

The first data set *"Metsim"* is generated by the weather forecast model "Weather Research and Forecasting model (WRF)" developed by the university of Washington, while the second data set *"Isabel"* is a simulation of a tropical hurricane done by "National Center for Athmospheric Research".

Both data sets are stored in the data format HDF5, with the same internal hierarchical

representation:
Velocity (group name)

- X-comp (data array name)
- Y-comp (data array name)

Here, the velocity field is given by V(x,y) = [X-comp(x,y),Y-comp(x,y)]. The resolution of the Metsim data is 127 points in both x- and y- direction. The resolution of the Isabel data is 500 points in both x- and y- direction.

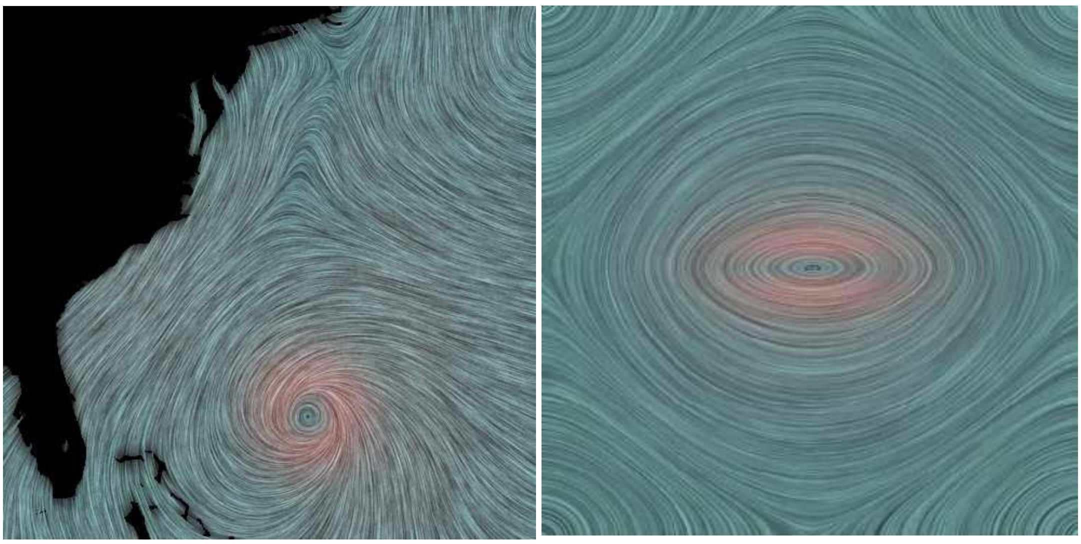
NOTE!!!!!
Both datasets are stored as two-dimensional arrays, one array for each component. However, there is a significant difference in how the data sets are created. For Metsim, the x-index is the first index, while for the Isabel data set the y-index is the first index. *The reason for this is that different program languages may have different ways of storing the data arrays (or matrices) internally.*

You can put the data in a local two-dimensional array as follows:

Isabel:
```
for (int i = 0; i < dim[0]; i++)
 for (int j = 0; j < dim[1]; j++){
   I[i][j] = dataarray[ j + (i * dim[1])];
}
```

metsim:
```
for (int j = 0; j < dim[1]; j++)
 for (int i = 0; i < dim[0]; i++){
   I[i][j] = dataarray[ i + (j *dim[0])];
}
```

Figure 2 shows examples of LIC pictures of the two vector fields. These pictures can be used to check whether you have done the coding right.



Figur 2: LIC-images of Isabel (right) and metsim (left).