Interaction of point vortices

Abstract

This report is a continuation of the work done as a part of Assignment 1. A few modifications have been made to the code to improve the post processing function of computing stream lines. Previously, stream lines always originate from the same set of points no matter what the time step is. This leads to inconclusive behavior of stream lines. This problem is tackled by trying to make the initializing points of the stream lines somehow attached to the vortices. Also, simulation results of three vortex simulations have been included in this report.

Structure of the code

The python code is as general as possible. It was tried not to hard wire the code to a specific problem. The general problem in mind is interaction of n vortices. All of the processing part of the code is same as that in the previous assignment. Only the post processing part is changed a little to generalize it. Also, a new function named *linComb* is defined so as to provide initializing values for stream lines for each time step. The structure of the code is as follows:

$$X[i] = \begin{pmatrix} x_1[i] & y_1[i] \\ x_2[i] & y_2[i] \\ \vdots & \vdots \\ x_n[i] & y_n[i] \end{pmatrix} \tag{1}$$

$$U(\gamma, X[i], n) = \begin{pmatrix} u_x^1[i] & u_y^1[i] \\ u_x^2[i] & u_y^2[i] \\ \vdots & \vdots \\ u_x^n[i] & u_y^n[i] \end{pmatrix} \tag{2}$$

Euler scheme of integration will now be of the form:

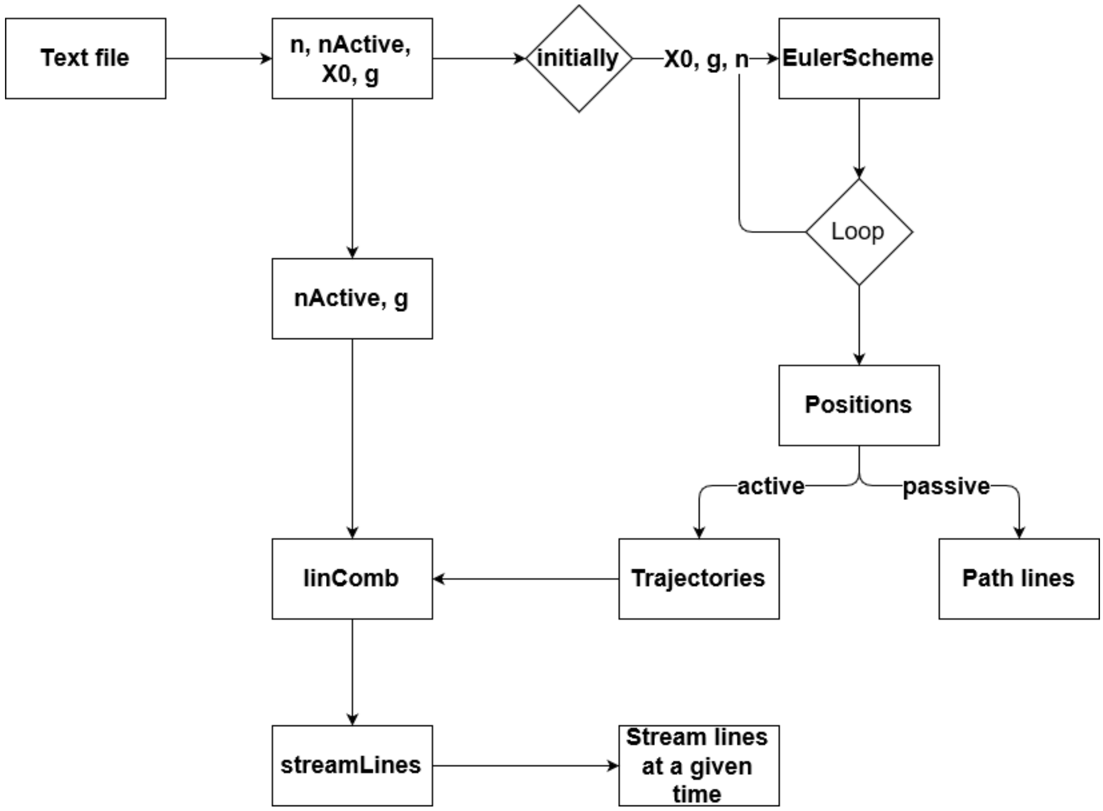
$$X[i+1] = X[i] + \delta t U(\gamma, X[i], n) \tag{3}$$

Stream lines are given by:

$$\frac{\delta y}{u_y} = \frac{\delta x}{u_x} = \delta t \tag{4}$$

$$y[i+1] = y[i] + u_y(x[i], y[i])\delta t \tag{5}$$

$$x[i+1] = x[i] + u_x(x[i], y[i])\delta t \tag{6}$$



Functions defined in the code

The following are the functions defined in the code. A short description of each function is given below.

U

Takes in the current positions of the particles and the strengths of each to output a list of velocities induced on each vortex.

EulerScheme

Takes in the current positions and the current velocities of the vortices to compute the new position of each vortex.

Velocity

Takes the coordinates of a point and the current positions of the vortices and outputs the velocity induced by the vortices at a point.

linComb

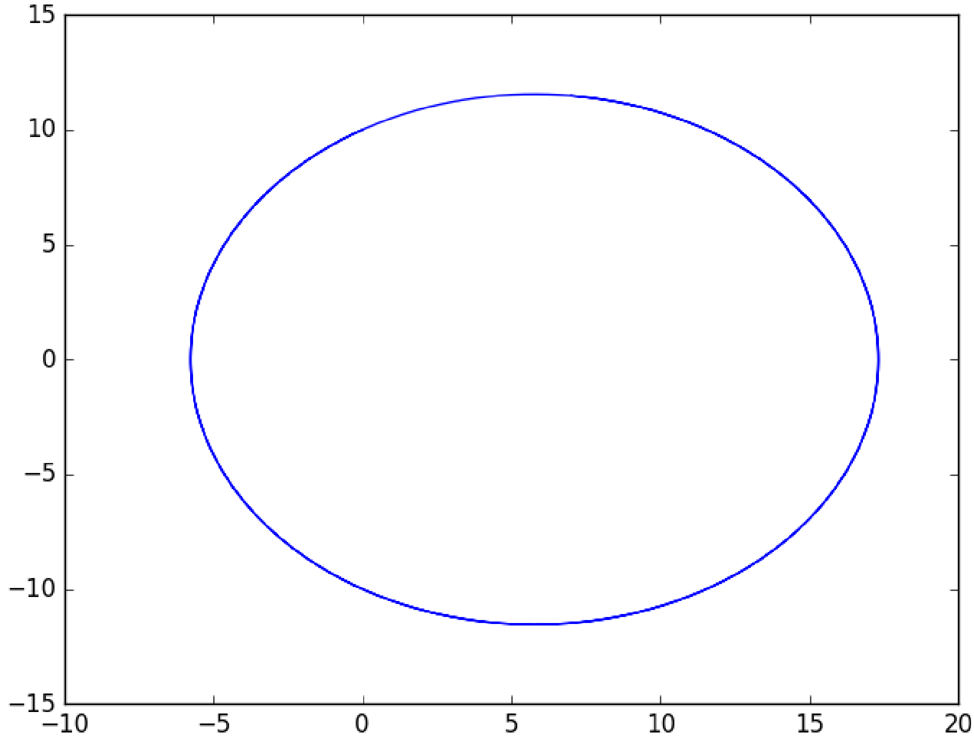
Takes the current positions of the vortices and outputs a linear combination of the positions of the vortices. This is a crude way to make sure that the initializing point of the vortex moves along with the vortices. This makes sure that the stream lines obtained are similar at each time step.

streamLines

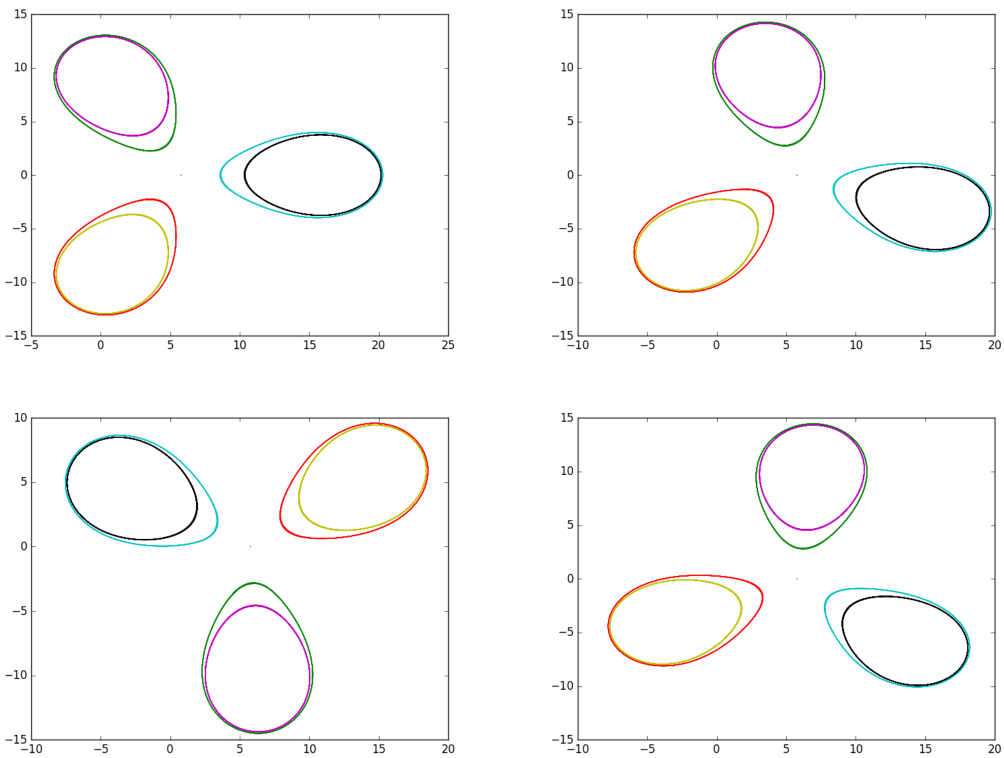
Takes in the current time, positions of vortices, initializing points and plots the stream lines at that time step.

Results of 3 vortices

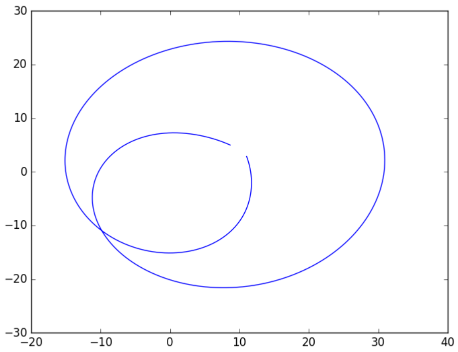
The vortices were initially located at$(0, 10.0)$, $(0, -10.0)$ and $(17.3205, 0)$. They constiute an equilateral triangle. Due to this symmetry, all the three vortices had the same closed trajectory which was:

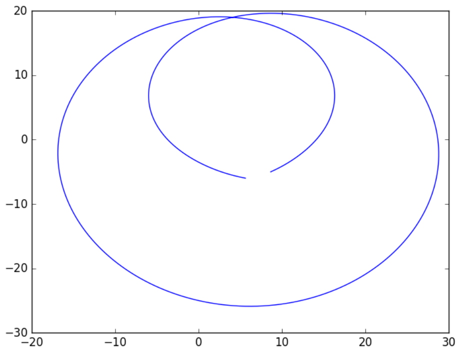


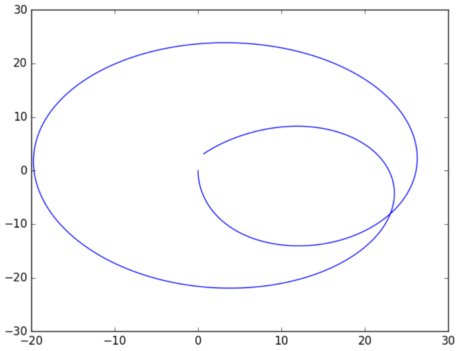
The stream lines plotted at times $t = 0$, $t = 500$, $t = 750$ and $t = 1000$ respectively are as follows:



The path lines of points starting form $(8.66025, 5.0)$, $(8.66025 - 5.0)$ and $(0.0, 0.0)$ are as follows:


30
20
10
0
−10
−20
−30
−20
−10
0
10
20
30
40


20
10
0
−10
−20
−30
−20
−10
0
10
20
30


30
20
10
0
−10
−20
−30
−20
−10
0
10
20
30