CPSC 446/546
Assignment 5
Due 11/27/2023, 11:59 pm

Upload to Canvas as a zip file named *yourfirstname_yourlastname*_5.zip. Include both the contents of the original assignment 5 zip file and your solution files in your submission.

This assignment requires you to develop visualizations using D3. Do your own coding using code provided with the assignment and examples given in the Scott Murray textbook. **Do not use any code from the internet that you may find that creates visualizations similar to those required in problems. Do not ask any AI or code co-pilot to write some or all of the code.** If we find that code you use for a solution is taken from an internet source or has been automatically generated, you will receive a zero for the entire assignment.

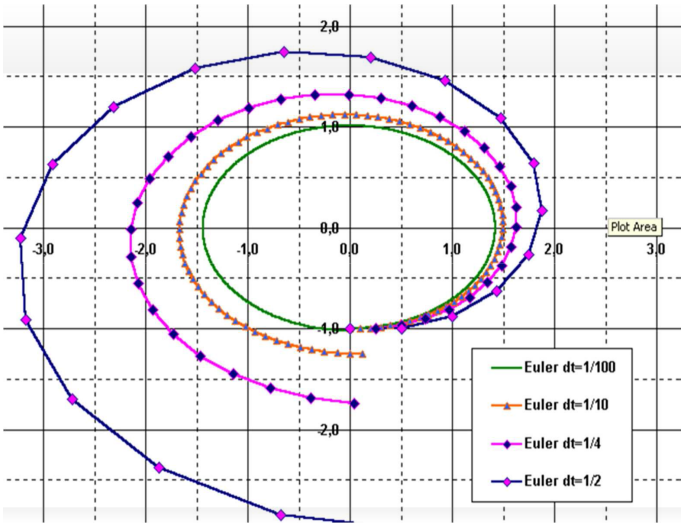
Be sure to reference d3.js in the same directory, i.e. use this in your html:

```
<script type="text/javascript" src="d3.js"></script>
```

As in previous assignments you may use https://developer.mozilla.org/en-US/docs/Web/JavaScript or https://javascript.info/ to do tutorials and look up syntax.

Note that some parts of the questions are for all students, and some are additional work for **CPSC 546** only.

1. Arrange Spatial Data -- Vectors (35 pts)
a. Produce a plot of Euler integration for the velocity field defined as [-y, (1/2)x] at each location [x,y]. Allow the user to enter a step size between 0 and 1.0 using a slider. Show the result as a path with the computed points marked on it as in the example from class shown below. As in the example, start the path at (0, -1), and show the field for x and y ranging from [-3,-3] to [3,3]. Unlike the example you only need to show one path at a time. You may just show the labeled axes rather than the full background grid.



(546 ONLY) b. Produce the same plot for the same field as in part a.) but use 4th order Runge-Kutta integration that improves the estimate of velocity along the step.

$$[x(t+dt), y(t+dt)] = [x(t), y(t)] + 1/6*dt* (\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

where:
$\mathbf{k}_1 = v[x(t),y(t)]$ // velocity at start point
$\mathbf{k}_2 = v\ [[x(t),y(t)]+ \mathbf{k}_1*dt/2\ ]$ // velocity after ½ step at start point velocity
$\mathbf{k}_3 = v\ [[x(t),y(t)]+ \mathbf{k}_2*dt/2\ ]$ // velocity after ½ step at refined velocity
$\mathbf{k}_4 = v\ [[(x(t),y(t)] + \mathbf{k}_3*dt]$ // velocity after step at further refined velocity

2. Arrange network data (35 pts)\
A chord diagram shows relationships between categories that are represented as angles on a circle with "ribbons" that connect the categories. The width of the ribbons can represent the magnitude of interaction between the categories.

Sample code for creating chord diagrams is given in
https://www.d3-graph-gallery.com/chord.html

Create a chord diagram for this table of migration data (each row is migration from the "Country", i.e. 2,021 people migrated from Austria to Belgium). Each chord should be colored according to the "source" country, and the countries should be clearly labeled on the diagram. The ribbons should be asymmetric (as in the example referenced above.) Your visualization does not need to be interactive.

Country	Austria	Belgium	France	Germany
Austria	0	2 201	7 665	191 269
Belgium	2 540	0	139 529	39 225
France	11 610	141 648	0	215 982
Germany	249 001	33 783	140 948	0

Also see:
The D3 API: https://d3js.org/d3-chord
Example by Mike Bostock: https://observablehq.com/@d3/chord-diagram

3. Geographic visualization (30 pts)

The map below is from the Murray book chapter_14/06_points.html plotted with points that represent (approximately) each states centroid (from statecentroids.csv that comes with this assignment), instead of cities. Using 06_points.html and the centroids as a starting point, plot two values for each state (one encoded with color, one with the size of circle) from www.census.gov. You can choose any data for each state that you like (population, housing etc.) One possibility is taxes collected:
https://www.census.gov/data/tables/2022/econ/stc/2021-annual.html
Some of the centroids aren't very good (such as Michigan and Florida), and you can update these manually if you would like.

(546 ONLY) Create a second map of the US, but instead of a circle create a glyph for each state that encodes 2 or more data values (e.g., for tax data each state could be colored by total taxes, and the glyph could represent income from types of taxes such as property and income tax.)