1. Consider the vector field $\vec{F}(x,y) = (x+y^2, -y)$ and its associated continuous dynamical system $(W^t, \mathbb{R}^2)$.

   (a) For each of the following functions, show whether or not it is a flow for $(W^t, \mathbb{R}^2)$.
   - i. $\vec{\varphi}(t) = (-e^{-2t}, \sqrt{3}e^{-t})$
   - ii. $\vec{\varphi}(t) = (-e^{-4t}, \sqrt{3}e^{-2t})$
   - iii. $\vec{\varphi}(t) = (\frac{4}{3}e^{t} - \frac{1}{3}e^{-2t}, e^{-t})$
   - iv. $\vec{\varphi}(t) = (e^{t} - e^{-2t}, e^{-t})$

   (b) A *constant flow* is a flow $\vec{\varphi}$ such that $\vec{\varphi}(t_1) = \vec{\varphi}(t_2)$ for all $t_1, t_2 \in \mathbb{R}$. Find all constant flows for $\vec{F}$. (Hint: think about $\vec{\varphi}'$ in this situation.)

   (c) Classify all fixed points of $(W^t, \mathbb{R}^2)$ as stable or unstable.

   (d) Find *all* flows of $(W^t, \mathbb{R}^2)$ that pass through the point $(1,0)$. Remember, if $\vec{\varphi}$ is a flow with this property, it is not a requirement that $\vec{\varphi}(0) = (1,0)$, only that $\vec{\varphi}(t_0) = (1,0)$ for some t_0.

   (e) We call the flows $\vec{\varphi}_1$ and $\vec{\varphi}_2$ time shifts of each other if $\vec{\varphi}_1(t) = \vec{\varphi}_2(t + t_0)$ for some t_0. Show that the flows from part (d) are time shifts of each other.

2. Let's explore the question of whether every discrete dynamical system can be described as the time-1 map of a continuous dynamical system.

   Let (W^t, X) be a continuous dynamical system, and let (T, X) be its time-1 map. That is, $T(x) = W^1(x)$.

   The point $x \in X$ is called *periodic* for (W^t, X) if there exists a $t \neq 0$ so that $W^t(x) = x$ and is called *periodic* for (T, X) if there exists an $i \neq 0$ so that $T^i x = x$. In both cases, the minimum t or i such $W^t(x) = x$ or $T^i(x) = x$ is called the *period* of x.

   (a) Suppose $x \in X$ is a *periodic point* for (T, X). Must x be a periodic point for (W^t, X)?

   (b) Suppose $x \in X$ is a *periodic point* for (W^t, X). Must x be a periodic point for (T, X)?

   (c) Show that if x is a point of period at least 2 for (T, X), then there are infinitely many periodic points for (T, X) *of the same period*.

   (d) Consider the *logistic map* $T : [0,1] \to [0,1]$ defined by $x \mapsto rx(1-x)$. Show that when $r = \frac{19}{6}$ the logistic map has exactly two points of period 2. (Hint: use a computer algebra system to solve any nasty equations you come across!)

   (e) Are all discrete dynamical systems time-1 maps to continuous dynamical systems? Justify your answer.

3. The function $G : \mathbb{R}^n \to \mathbb{R}^m$ is called *affine* if there exists a matrix A and a vector $\vec{p}$ so that $G(\vec{x}) = A\vec{x} + \vec{p}$ for all $\vec{x}$.

   Let $F : \mathbb{R}^n \to \mathbb{R}^m$. A *first-order approximation* to F at the point $\vec{w} \in \mathbb{R}^n$ is an affine function $L : \mathbb{R}^n \to \mathbb{R}^m$ satisfying

   $$\lim_{\|\vec{x}\| \to 0} \frac{F(\vec{w} + \vec{x}) - L(\vec{w} + \vec{x})}{\|\vec{x}\|} = 0.$$

   (a) Let $F : \mathbb{R} \to \mathbb{R}$ be defined by $x \mapsto x^2$. Find a first-order approximation, L_0, to F at 0, and a first-order approximation, L_2, to F at 2.

   (b) Let $F : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by $\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$. Define $L_{\vec{w}}$ to be the first-order approximation of F at the point $\vec{w}$. Find $L_{\vec{w}}$.

   (c) Let L be an affine function and let $L_{\vec{w}}$ be the first-order approximation to L at $\vec{w}$. Prove that $L = L_{\vec{w}}$ regardless of $\vec{w}$.

(d) Prove that if $(W^t, \mathbb{R}^n)$ is a continuous dynamical system with velocities given by $V(\vec{x}) = A\vec{x}$ for some matrix A, then the point $\vec{x} \in \mathbb{R}^n$ is stable under W^t if and only if the point $\vec{0}$ is stable under W^t.

(e) From calculus, you know that if $f : \mathbb{R} \to \mathbb{R}$ is differentiable, then $L(w + x) = f'(w)(x) + f(w)$ is a first-order approximation to f at w, where f' is the derivative of f. Use this knowledge to find a first-order approximation to $F(x, y) = (x+y^2, -y)$ at the points $(0, 0)$ and $(1, 1)$.

(f) Let $(W^t, \mathbb{R}^2)$ be the continuous dynamical system that flows along $F(x, y) = (x+y^2, -y)$. Classify $(0, 0)$ and $(1, 1)$ as stable or unstable. Justify your answer; in particular, if you use a first-order approximation, you must explain why it is an appropriate approximation to use.

4. Stability/instability describes how points behave under a dynamical system. Let's take a moment to think about how *volumes/areas* behave.

   For this problem, you may use any facts you know about the determinant without justification (so long as they're true…).

   (a) The *trace* of a square matrix X, denoted $\mathrm{Tr}(X)$, is the sum of its diagonal entries. Let $A = [\vec{a}_1|\cdots|\vec{a}_n]$ be a matrix with columns $\vec{a}_1, \ldots, \vec{a}_n$. Let E_i be the identity matrix with the ith column replaced with $\vec{a}_i$. Show that

   $$\mathrm{Tr}(A) = \sum \det(E_i).$$

   (b) Let $(W^t, \mathbb{R}^2)$ be the continuous dynamical system which flows vectors along the vector field given by $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$.

   Write out the limit definition of the derivative $\frac{\partial W^t}{\partial t}$ at time $t = 0$. How does this derivative relate to A?

   (c) Write down a first-order approximation to W^t *with respect to time* at $t = 0$.
   Hint: the hardest part of this question is figuring out what the previous sentence actually means. Don't get discouraged!

   (d) Using a first-order approximation for W^t, estimate the volume of the image of the unit cube after flowing for ε seconds.

   (e) Find the instantaneous rate of change of volume (area) with respect to time for W^t at time 0.

   Hint: Recall that det is a *multi-linear* function of the columns of a matrix. That is, linear relationships like $\det([\vec{c}_1|\vec{c}_2 + \alpha\vec{d}|\vec{c}_3|\cdots]) = \det([\vec{c}_1|\vec{c}_2|\vec{c}_3|\cdots]) + \alpha\det([\vec{c}_1|\vec{d}|\vec{c}_3|\cdots])$ hold for every column.

   (f) Show that if you drop a dab of ink of area α near the origin and let it flow for via W^t for ε seconds, the area of the resulting blob of ink will be approximately $\alpha\varepsilon\,\mathrm{Tr}(A)$.

   (g) Let $(N^t, \mathbb{R}^2)$ be a continuous dynamical system which flows points along the vector field $F : \mathbb{R}^2 \to \mathbb{R}^2$ given by $F(x, y) = (f_x(x, y), f_y(x, y))$.

   The first-order approximation to F at a point (x_0, y_0) is given by the *Jacobian* of F at that point. That is, if we define

   $$J(F)\big|_{(x_0,y_0)} = \begin{bmatrix} \frac{\partial f_x}{\partial x}(x_0, y_0) & \frac{\partial f_x}{\partial y}(x_0, y_0) \\ \frac{\partial f_y}{\partial x}(x_0, y_0) & \frac{\partial f_y}{\partial y}(x_0, y_0) \end{bmatrix},$$

   then

   $$F\big((x_0, y_0) + \vec{w}\big) \approx J(F)\big|_{(x_0,y_0)}\vec{w} + F(x_0, y_0).$$

A vector field is called *incompressible* if $\nabla \cdot F = \frac{\partial f_x}{\partial x} + \frac{\partial f_y}{\partial y} = 0$ at all points in its domain (i.e., the divergence is zero everywhere).

Is the term incompressible warranted? Explain using what you've learned from this problem.

Programming Problems

For the programming problems, please use the Jupyter notebook available at

https://utoronto.syzygy.ca/jupyter/user-redirect/git-pull?repo=https://github.com/siefkenj/2020-MAT-335-webpage&subPath=homework/homework2-exercises.ipynb

Make sure to comment your code and use "Markdown" style cells to explain your answers.

1. We're going to make some beautiful simulations of continuous dynamical systems!

   The function `plot_vectorfield` takes in a matplotlib *axis* and a function from $\mathbb{R}^2$ to $\mathbb{R}^2$ and plots the corresponding vector field. The functions `V1`, `V2`, `V3`, and `V4` from class have already been defined for you.

   Use `plot_vectorfield` to create a plot that contains 4 subplots. Plot `V1`, `V2`, `V3`, and `V4` in these subplots. The extents should be the rectangle $[-2, 2]^2$ and you should plot a 12×12 grid of vectors (i.e., $N = 12$).

   Using `ax.set_title()`, label each of your subplots appropriately.

2. The simplest way to simulate a flow along a vector field is called Euler's method.[1] Let $\vec{F} : \mathbb{R}^n \to \mathbb{R}^n$ be a vector field. Euler's method says to simulate a flow along $\vec{F}$ starting at $\vec{v}$ perform a series of computations

$$\begin{aligned}
\vec{v}_1 &= \vec{v} + \varepsilon \vec{F}(\vec{v}) \\
\vec{v}_2 &= \vec{v}_1 + \varepsilon \vec{F}(\vec{v}_1) \\
\vec{v}_3 &= \vec{v}_2 + \varepsilon \vec{F}(\vec{v}_2) \\
&\vdots \\
\vec{v}_n &= \vec{v}_{n-1} + \varepsilon \vec{F}(\vec{v}_{n-1})
\end{aligned}$$

   Then, $\vec{v}_n$ will be the approximate result of flowing along $\vec{F}$ for $n\varepsilon$ seconds.

   Write a function `euler` that takes in a starting x-coordinate, y-coordinate, a number of steps and an epsilon and performs Euler's method for `V1` for the specified number of steps with the specified epsilon.

3. Let's make some pictures! Along they way, we'll make a more general method.

   Create two functions `flow` and `orbit`. They each take in a starting x, y, a function that defines a vector field, a time, and an epsilon.

   `flow` will use Euler's method with the specified parameters and output the result of flowing along the vector field for `t` seconds (well, the approximate result).

   `orbit` will perform the same operations as `flow` except it will return a tuple `(xs, ys)` where `xs` and `ys` are Numpy arrays with the x and y coordinates of every step of Euler's method. For example, if you did two steps of Euler's with $\vec{v} = (1, 0)$, $\vec{v}_1 = (1, 0.01)$, and $\vec{v}_2 = (0.99, 0.02)$, then `orbit` would return `xs=(1,1,0.99)` and `ys=(0,0.01,0.02)`.

[1] If you want to play with more advanced methods, you can look up the *Runge-Kutta-Fehlberg* methods. http://web.cs.ucdavis.edu/~ma/ECS177/papers/particle_tracing.pdf

4. For V_1, we know that all vectors flow in a circle with period 2π seconds. Let's see what our simulations tell us.

   Make a plot with the vector field V_1 and the $t = 4\pi$ orbits[2] of the point $(1, 0)$ simulated with a steps size (epsilon) of 0.1, 0.01, and 0.001. Are your simulated orbits periodic? Explain.

5. We are going to try to guess the period of orbits from our simulations. The idea we'll use is as follows: We compute and see which points along the orbit are close to the starting point. We then use this information to compute the "time" that the orbit was closest to the starting point.

   We'll implement an algorithm to do this in steps.

   (a) A *run* of integers is a sequence of integers $x_1, x_2, \ldots$ where $x_i = x_{i-1}+1$. Create a function `split_runs` which takes a list of integers and returns a list containing all maximal runs contained within the original sequence.

   (b) Create a function `indices_close_to_min` which inputs a 1D Numpy array and a tolerance and returns a list containing the indices of that array which are within the specified tolerance of the array's minimum value.

   (c) Create a function `guess_minimums` that inputs a Numpy array `seq`, an epsilon, and a tolerance and outputs a guess of where the minimums of `seq` are (in the form of a list); instead of returning the index of the minimum, your function should return epsilon×index of minimums.

   Some things to keep in mind:

   - `seq` may get close to the its minimum several times (and in our case, it will likely *start* at its minimum).
   - If `seq` is close to its minimum at the *run* of indices (a, b, c), the minimum is probably closest to occurring at index b.
   - If the input to your function came from an Euler approximation with step size epsilon, this function will output an estimate of what *time* the minimums occur.

   (d) We know that the period of points flowing along V_1 is exactly 2π (excluding the origin). Use `orbit` and `guess_minimums` to estimate the period of flowing along V_1 starting at $\vec{v}_0 = (-1, 0)$. Estimate the period using step sizes of 0.01 and 0.001. How well do you approximate 2π?

6. Let's analyze the mysterious V_3.

   (a) Plot the vector field V_3 along with the orbits of $(-1, 0)$, $(-1.1, 0)$, and $(-1.2, 0)$. Do you think flows starting at these points are periodic? Why or why not?

   (b) Estimate the periods of the flows starting at $(-1, 0)$, $(-1.1, 0)$, and $(-1.2, 0)$. Can you say with confidence whether or not the periods are the same?

   (c) Once and for all, argue whether or not flows along V_3 are stable.

7. **Flows of regions**. It's time to drop some ink in the current! We are going to flow polygonal regions along our vector fields.

   (a) Create the functions `make_perimiter_circle` and `make_perimiter_square`. `make_perimiter_circle` inputs a center and radius and number of points and outputs points along the specified circle (in counter-clockwise order). `make_perimiter_square` inputs a lower-left corner of a square, its width, and a number and outputs that many points along the edges of the square (again in counter-clockwise order).

   Your functions should return a tuple `(xs, ys)` with the coordinates.

[2] The orbit restricted to times between 0 and 4π.

(b) We can graph polygonal regions in Matplotlib using the `Polygon` and `PatchCollection` functions. But, they require as inputs a list of ordered pairs (whereas most other functions want the xs and ys as separate arrays). The `zip_coords` utility function will take a tuple of x coordinates and y coordinates and turn it into a list of coordinate pairs.[3]

Read and execute the notebook cell that graphs a square and a circle.

(c) By using the trusty `np.vectorize`, we can turn our `flow` function into one that accepts lists of coordinates instead of single coordinates. Create `vec_flow`, the vectorized version of `flow`.

Now, plots the vector field V_3 along with the square with lower-left corner $(-1, 0)$ and width $1/2$ and the image of this square after it has flowed for 1 second.

[3]This is a very similar operation to the transpose!