Lab 4

In this lab, we will explore different ways to solve the differential equation

$$y' = f(x, y) \tag{1}$$

using Mathematica. Specifically, we look into the following methods:

- Solve some differential equations using `DSolve`,
- Plot the direction field and solution curves of (1),
- Use Euler's method to solve (1) approximately,
- Use contours to visualize solution curves of a separable equation.

1 Solve some differential equations with DSolve

(1) Many differential equations can be solved using the command `DSolve`. For example, consider the equation $y' + y = e^x$. Try the following:

```
Clear[x, y]
DSolve[y'[x] + y[x] == Exp[x], y[x], x]
```

The command `Clear[y]` is to erase from the memory the variables x and y in case they were previously used. The double equal sign is used to indicate an *equation*. A single equal sign is used when you want to *assign a value to a variable*.

(2) If the initial value is given, say $y(0) = 1$, then adjust the command as follows.

```
Clear[x, y]
DSolve[{y'[x] + y[x] == Exp[x], y[0] == 1}, y[x], x]
```

(3) Graph the solution with $y(0) = 1$ and the solution with $y(0) = -1$ on the same plot.

(4) Solve the initial value problem $y' = (x^2 + 1)(y^2 + 1)$, $y(0) = 0$. *Be careful:* in `DSolve`, make sure that you write `y[x]` instead of `y`. Plot the solution.

(5) There are also many differential equations that `DSolve` cannot solve. For example, try solving the equation $y' = x + y^3$ with `DSolve` and see what you get. *Be careful:* in `DSolve`, you need to write `y[x]` instead of `y`.

2 Direction fields

For each point (a, b) on the plane, the curve $y = y(x)$ that satisfies (1) and passes through (a, b) will have a slope of $y'(a) = f(a, y(a)) = f(a, b)$ at $x = a$. If we place at each point (a, b) on the plane a tiny line segment of slope $f(a, b)$, we will get a *direction field*. On Mathematica, you can draw a direction field using the command **VectorPlot**. The syntax of this command is

```
VectorPlot[{1,f[x,y]}, {x,xmin,xmax}, {y,ymin,ymax}]
```

(6) For example, consider the differential equation $y' = \frac{x^2}{y^2+1}$. Enter the following:

```
f[x_, y_] := x^2/(y^2 + 1)
VectorPlot[{1, f[x, y]}, {x, -5, 5}, {y, -5, 5}]
```

You see a map of arrows. The arrows are of the same length but with different colors. The lighter color, the longer the actual length of the arrow.

(7) The colors are not important for our purposes. You can add `VectorColorFunction ->None` to the **VectorPlot** command to remove the color.

```
VectorPlot[{1, f[x, y]}, {x, -5, 5}, {y, -5, 5}, VectorColorFunction ->None]
```

(8) To increase the number of arrows, you can add the option `VectorPoints->n` to the command **VectorPlot**. Here n is a number of your choice, representing the number of arrows a each vertical line. The default value is 15. Try again the command in the previous exercise with $n = 20, 30, 50, 100$.

(9) To "link" these arrows into a curve (called *solution curve*), we use the command **StreamPlot**. Try the following:

```
StreamPlot[{1, f[x, y]}, {x, -5, 5}, {y, -5, 5}]
```

You can remove the color by adding the option `StreamColorFunction -> None`. From the picture, what is behavior of all the solution curves as $x \to \infty$?

(10) To single out the solution curve that passes through a given the point, you add an option `StreamPoints` to the above command. For example, to highlight the solution curve that passes through the point $(1, -1)$, which is the curve satisfying $y(1) = -1$, try the following:

```
StreamPlot[{1, f[x, y]}, {x, -5, 5}, {y, -5, 5},
                StreamColorFunction -> None,
                   StreamPoints -> {{{{1, -1}, Red}, Automatic}}]
```

Sorry, lots of curly brackets! The part `{{1, -1}, Red}` is to specify that we want the solution curve passing through $(1, -1)$ to be red. And `Automatic` is to specify that all other solution curves have a default color (blue).

(11) To highlight one more solution curve, for example the curve that passes through the point $(0, 1)$, put `{{0, 1}, Green}` in front of `Automatic`.

(12) Sketch the direction field of the differential equation $y' = \frac{x}{y^2}$. Use both **VectorPlot** and **StreamPlot**.

(13) Highlight the solution curves satisfying the initial conditions $y(0) = 4$, 3, 2, 1, 0.5.

(14) Mathematica will fail to draw the solution curve satisfying $y(0) = 0$ because the function $\frac{x}{y^2}$ is not well-defined when $y = 0$. However, by looking at the five curves in the previous exercise, can you guess what the curve satisfying $y(0) = 0$ will look like?

(15) Pick a differential equation of your choice (but still of the form $y' = f(x, y)$). Show the direction field and show the solution curve satisfying $y(1) = 0$.

3 Euler's method

Consider an initial value problem

$$y' = f(x, y), \quad y(x_0) = y_0.$$

Suppose that we want to solve for $y = y(x)$ on the interval $x \in [x_0, x_0 + T]$. With step size h, Euler's method allows us to solve *approximately* the value of y at x_1, x_2, x_3, x_4,,... where

$$\begin{aligned} x_1 &= x_0 + h \\ x_2 &= x_1 + h \\ x_3 &= x_2 + h \\ &\dots \end{aligned}$$

More specifically, $y(x_n)$ is approximated by y_n, where y_n is found from the recursive formula

$$y_{n+1} = y_n + hf(x_n, y_n)$$

(16) To execute this recursive formula on the computer, we need to write a *loop*. In Mathematica, the simplest loops are perhaps the `Do` loop and the `For` loop. The syntax of the `Do` loop is as follows:

```
Do[expr, {i,imin,imax}]
```

Mathematica will execute *expr* with the variable *i* successively taking on the values *imin* through *imax*. Notice the comma (not semicolon) between `expr` and `{i,imin,imax}`. If `expr` should contain multiple commands, separate those commands from each other by semicolons. For example, to find the sum and product of natural numbers from 1 to 100, we do the following:

```
sum = 0;
prod = 1;
Do[
    sum = sum + i;
    prod = prod*i,
   {i, 1, 100}
   ]
sum
prod
```

Note that the `expr` in this code has two commands: one is `sum = sum + i` and the other is `prod = prod*i`. They are separated from each other by a semicolon.

(17) Use a single `Do` loop to find two sums:

$$\frac{2^1}{1^2} + \frac{2^2}{2^2} + \frac{2^3}{3^2} + \dots + \frac{2^{100}}{100^2}$$
$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{100}$$

(18) Let us consider the differential equation $y' = x + y$ with an initial condition $y(0) = 1$. We want to solve for $y = y(x)$, $x \in [0, 3]$. To prepare for the Euler's method, we enter the known parameters:

```
Clear[x, y, f]
f[x_, y_] := x + y;
x0 = 0;
y0 = 1;
T = 3;
```

(19) Let us take the step size $h = 0.1$. The number of mesh points $x_1, x_2, x_3, ...$ on the interval $[x_0, x_0 + T]$ is $M = T/h$. We enter these parameters:

```
h = 0.1;
M = T/h;
```

(20) Now we are ready to execute the recursive formula:

```
x[0] = x0;
y[0] = y0;
Do[
      x[n + 1] = x[n] + h;
      y[n + 1] = y[n] + h*f[x[n], y[n]],
      {n, 0, M}
   ]
```

For explanation sake, the full code has been split into 3 blocks in the above three exercises. It is better if you combine them into one single code block so that all of them can be executed at once.

(21) To put all the points (x_n, y_n) on the coordinate system, we use the command **ListPlot**.

```
A = Table[{x[n], y[n]}, {n, 0, M}];
ListPlot[A, PlotRange -> Full]
```

To change the color of the points to red, you can add the option `PlotStyle -> Red` inside the command **ListPlot**.

(22) You can also draw the table of values. This table has 3 columns: one for n, one for x_n, and one for y_n.

```
B = Table[{n, x[n], y[n]}, {n, 0, M}];
TableForm[B, TableHeadings -> {None, {"n", "xn", "yn"}}]
```

(23) The initial value problem $y' = x+y$, $y(0) = 1$ is simple enough to solve by hand (or by `DSolve`). The exact solution is $y = -1-x+2e^x$. You can graph the exact function and the points (x_n, y_n) on the same plot to see how well the approximation is.

```
p1 = Plot[-1 - x + 2*E^x, {x, 0, 3}, PlotStyle -> Red];
p2 = ListPlot[A, PlotRange -> Full];
Show[p1, p2]
```

What do you observe? What part of the interval $[0, 3]$ is the approximation good, and what part of it is the approximation not so good? Give an explanation. What is the largest error between the exact solution (the curve) and approximate solution (the dots)?

(24) Reduce the step size to $h = 0.05$ and see if the approximation gets better or worse.

(25) Repeat Exercises 18 through 23 for the differential equation $y' = xy^2$, $y(-4) = -0.1$, on the interval $x \in [-4, 4]$.

4 Solve separable equations using the contour method

In general, a separable equation only gives solutions in an implicit form.

(26) For example, to solve the equation

$$y' = \frac{x^3+1}{y^3+1}$$

as normally done by hand, we separate y and x as

$$(y^3+1)dy = (x^3+1)dx$$

and then integrate each side:

```
Integrate[y^3+1, y]
Integrate[x^3+1, x]
```

You get an implicit formula $\frac{y^4}{4} + y = \frac{x^4}{4} + x + C$, or equivalently

$$\frac{y^4}{4} + y - \frac{x^4}{4} - x = C.$$

It is difficult, sometimes impossible, to derive an explicit formula for y from the implicit formula.

(27) Suppose the initial condition is $y(0) = 1$. Use the command `DSolve` to see if Mathematica is able to find an explicit formula for the solution. *Keep in mind:* in `DSolve`, make sure that you write `y[x]` instead of `y` alone. If it is taking too long, press the combination *Alt + .* to terminate the evaluation.

Whether you get an explicit formula or not, you can always visualize the solution from the implicit formula. The command **ContourPlot** plots all the points (x, y) satisfying a given equation.

(28) Suppose the initial condition is $y(0) = 1$. In this case, the constant $C = \frac{5}{4}$. To draw the collection of all the points (x, y) satisfying the equation $\frac{y^4}{4} + y - \frac{x^4}{4} - x = \frac{5}{4}$, try the following:

```
ContourPlot[y^4/4 + y - x^4/4 - x == 5/4, {x, -3, 3}, {y, -3, 3},
            ContourStyle -> {Thick, Red}]
```

What you see is the graph of y as a function of x (without knowing an explicit formula of y).

(29) Let us compare the solutions with different initial conditions: $y(0) = 1$ and $y(0) = 2$. In the first case, $C = \frac{5}{4}$. In the second case, $C = 6$. You will plot each solution using `ContourPlot` and then "combine" them into one plot.

```
p1 = ContourPlot[y^4/4 + y - x^4/4 - x == 5/4, {x, -3, 3}, {y, -3, 3},
        ContourStyle -> {Thick, Red}]
p2 = ContourPlot[y^4/4 + y - x^4/4 - x == 6, {x, -3, 3}, {y, -3, 3},
        ContourStyle -> {Thick, Blue}]
Show[p1, p2]
```

(30) Follow Exercise 26 to derive an implicit solution formula for the separable equation

$$y' = \frac{y(5x-2)}{x(1-3y)}$$

given that both x and y are positive. This differential equation comes from the predator-prey model by Lotka and Volterra.

(31) Suppose the initial condition is $y(1) = 1$. Use `ContourPlot` to visualize the solution.

(32) You can see that the picture shown in the previous exercise is not a graph of a function (Vertical Line Test fails). Can you identify from the picture the largest interval of x on which y is a function of x ? (This interval is called the maximal interval of existence of the solution.)
Hint: look at the differential equation as well as the graph and see for what value(s) of x is the slope of the curve equal to infinity.

(33) Show the solution with the initial condition $y(1) = 1$ and the solution with the initial condition $y(1) = 1/2$ on the same plot. How are they compared to each other?

5 To turn in

Submit your implementation of Exercises (1) - (33) as a single pdf file.