Numerical Methods Exercises

Exercise 1:

Consider the differential equation $\frac{dy}{dt} = y^2$.

Note: $y = 0$ is a fixed point, i.e. the derivative there is 0. This means if y is ever 0, then $y$would stay 0 forever. Thus if y is ever negative, then y stays negative for all time. Likewise, if y is ever positive, then y stays positive for all time.

Analytically solve this differential equation using the method of separation of variables subject to the initial condition $y(0) = -1$. Note that the solution has a singularity at $t = -1$.

What is the value of $y(1)$?

Thing to consider: Can every differential equation be solved this way? Can the Hodgkin-Huxley equations?

Exercise 2:

Use the Explicit Euler method to numerically solve the differential equation $\frac{dy}{dt} = y^2$ subject to the initial condition $y(0) = -1$.

Implement the algorithm yourself; do not use an ode solver library.

Graphically compare the solutions you obtain from several different choices of timestep with each other and with the exact solution you found in exercise 1. Plot the error at $t = 1$ versus the time step. Describe the relationship between the two.

Bonus challenge: implement the algorithm in a way that allows easily switching to solve a different differential equation.

Bonus challenge: implement the algorithm in a way that solves systems of differential equations. Test with $\frac{dx}{dt} = -y$ and $\frac{dy}{dt} = x$ where $x(0) = 1$ and $y(0) = 0$. For this example, plot x and $y$versus t. In a separate graph, plot y versus x. Describe the shape of the trajectory in this second plot.

Exercise 3:

Repeat the main part of exercise 2 (not the bonus challenges) using the Implicit Euler method instead.

Hint: Your code will have to solve a quadratic. Remember that the solution to $ax^2 + bx + c = 0$ is $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$.

Hint: To decide whether to add or subtract, recall the observation in exercise 1 that $y < 0$ for all time.

Bonus challenge: Repeat the same exercise except with the Implicit Midpoint Method: $y_{n+1} = y_n + dt\, f(\frac{1}{2}(y_n + y_{n+1}))$. Although the formula and process are similar to Implicit Euler, the error convergence is different. How might this difference in error convergence rates affect your choice of integration method?

Exercise 4:

A major component of exercise 3 was using the quadratic formula to solve for y or for Δy. Most algebraic functions, however, do not have closed form solutions.

In particular, consider the function: $f(x) = e^x - x - 2$

Since $f'(x) = e^x - 1$ is positive for $x > 0$, f is monotonic on that domain, so it has at most one root (place where it is 0) there. Further as f is continuous, $f(0) = -1$, and $\lim_{x \to \infty} f(x) = \infty$, by the intermediate value theorem there must be at least one root on $x > 0$. Therefore, there is exactly one root on $x > 0$. A similar argument shows there is exactly one root on $x < 0$.

Plot the graph of f on an appropriate domain to convince yourself that there are in fact exactly two roots and to identify their approximate locations.

Newton's Method is one strategy for locating roots. Starting with a point x_0, we can define a sequence $x_{n+1} = x_n - f(x_n) \,/\, f'(x_n)$. Intuitively, this algorithm chooses the next point to be where the function would be 0 based on its slope if it were a linear function. Under appropriate conditions on f, the sequence will converge to a root.

Use Newton's Method to approximate *both* roots of $f(x) = e^x - x - 2$.

Exercise 5:

Newton's Method required us to know the derivative of the function. If that information is not available, it may be approximated by the forward difference:

$$f'(x) \approx \frac{f(x+dx) - f(x)}{dx}$$

or by the central difference

$$f'(x) \approx \frac{f(x+dx) - f(x-dx)}{2\,dx}$$

for small values of dx.

For both approximations, estimate the value of $f'(1)$ for $f(x) = e^x - x - 2$ for several choices of dx. How do the errors compare for the two approaches as $dx \to 0$?

Modify your Newton's Method code from exercise 4 to work for an arbitrary function by approximating the derivative using the approximation from this exercise that converges most quickly.

Exercise 6:

We cannot simply choose arbitrarily small choices of dx to get better approximations. While this would work in exact arithmetic, computers have limited precision due to how they represent numbers internally (usually IEEE 754).

Consider the following Python session:

```
>>> 1 + 1e-10 == 1
False
>>> 1 + 1e-100 == 1
True
```

Clearly in both cases the two quantities are not equal, but in the second case we have numerical equality.

Find the value of *machine epsilon*, which is defined as the smallest number that when added to one is not numerically equal to 1. (Hint: this is a power of 2 because computers use binary.)

Find two positive numbers a and b with b less than machine epsilon, but $a + b \neq a$ numerically.