Introduction: Euler’s Method – A Numeric Solution to a differential equation \( \frac{dy}{dx} = g(x, y) \)
given an initial condition \( y_0 = f(x_0) \). Since \( \frac{\Delta y}{\Delta x} = \frac{dy}{dx} \) for small values of \( \Delta x \) it follows that
\[
\Delta y \approx \frac{dy}{dx} \Delta x \text{ or }
\]
\[
y(x + \Delta x) = y(x) + \frac{dy}{dx} \Delta x = y(x) + g(x, y) \Delta x
\]
This can be leveraged into a series of equations that be used to generate values for the underlying function \( y = f(x) \). Specifically
\[
y_1 = y_0 + g(x_0, y_0) \Delta x
\]
\[
y_2 = y_1 + g(x_1, y_1) \Delta x
\]
\[
y_n = y_{n-1} + g(x_{n-1}, y_{n-1}) \Delta x
\]
where \( x_i = x_{i-1} \Delta x \) for \( i = 1, 2, ..., n \). This is Euler’s Method.

In this lab we will implement two versions of Euler’s Method, one that given any differential equation of the form \( \frac{dy}{dx} = g(x, y) \) and an initial value \( y_0 = f(x_0) \) will approximate some \( y_i = f(x_i) \) and a second that will generate a list of intermediate points for the between \( y_0 = f(x_0) \) and \( y_i = f(x_i) \)

Code for Euler: Note that \( g[x, y] \) is the function for the differential equation

```math
Euler[x1_, g_, x0_, y0_, n_] := Module[{x, y, k, deltax},
  (deltax = (x1 - x0)/n;
   x = x0;
   y = y0;
   For[k = 0, k < n, k++,
     y = y + g[x, y] * deltax;
     x = x + deltax];
   y)
]
```
Code for EulerTable

\[
\text{EulerTable}[x1_, \ g\_, \ x0\_, \ y0\_, \ n\_] \ := \ \text{Module}[\{x, \ y, \ k, \ \text{deltax}, \ pts\}, \\
(\text{deltax} \ = \ (x1 - x0)/n; \ \\
x \ = \ x0; \ \\
y \ = \ y0; \ \\
pts \ = \ \{(x, \ y)\}; \ \\
\text{For}[k \ = \ 0, \ k \ < \ n, \ k++, \ \\
y \ = \ y + g[x, \ y]\ \text{\ast} \ \text{deltax}; \ \\
x \ = \ x + \ \text{deltax}; \ \\
\text{AppendTo}[pts, \ \{x, \ y\}] \\
]\];
\]

**Part 1:** Test these two modules using \(g[x, y] = y\) with initial value \((0, 1)\). What is \(f(1)\)?

**Part 2:** Euler solution to \(\frac{dQ}{dt} = 0.12Q\) with \(Q(0) = 1000.00\). Reproduce the results in Table 11.4

**Part 3.** Newton Cooling: Graph Euler solutions to \(\frac{dH}{dt} = -k(H - 20)\) with \(H(0) = 100\) for three values of \(k\): 0.2, 0.3, and 0.4. Create a combined plot. Using trial and error: when does the soup reach 25 degrees Celsius?

**Part 4:** In Lab 11 we modeled data on sheep population (see Problem 3 on page 469) using a logistic model. Graph the Euler Solution to \(\frac{dP}{dt} = \frac{\text{r} \text{(M - P)} \text{P}}{\text{M}}\) for \(P(0) = 125\) for the next 60 years where \(M = 1800\) and \(r = 0.0000660633\). Note that \(M\) and \(r\) were valued obtained from Lab 11. To compare with the original data, do a combined plot

Note: Let 1814 be base year 0. Use the data

\[
\text{year} \ = \ \{0, \ 10, \ 20, \ 30, \ 40, \ 50\} \\
\text{sheet} = \ \{125, \ 275, \ 830, \ 1200, \ 1750, \ 1650\}
\]