Section 2.5

**Improved Euler’s Method**

In the last section, we saw how to use Euler’s Method to approximate the solution to a differential equation. We also saw that Euler’s method produces quite a bit of error, and that we could reduce the error by reducing the step size.

Of course, the smaller the step size, the more calculations we will have to make in order to approximate the curve. This can become a real computational issue.

So instead of using the original Euler’s method with a smaller step size, it is often best to use a different approximation method entirely, preferably one that has better built-in error control. We will look at two such methods, one in this section and one in the following.

**Improved Euler’s Method**

It is possible to tweak some of the calculations in the original Euler’s method and produce a new method with dramatically less error. We illustrate the tweak below.

We’ll begin the approximation process just as we did with the original Euler’s method. Use the available information about \( \frac{dy}{dx} \) to draw the tangent line to the curve at the point \((x_0, y_0)\), and evaluate the tangent line at \( x_1 = x_0 + h \) to get the first Euler point approximation:

![Tangent line](image1)

Calculate the second Euler point approximation the same way:

![Tangent line](image2)

However, neither of the Euler point approximations will be used in the improved Euler method approximation. Instead, we will in a sense average the data from the two steps above to get the first improved Euler point approximation.

Average the slopes of the two tangent lines computed above, and draw the resulting line:
Evaluate this new line at $x_1 = x_0 + h$ to get the first improved Euler point approximation:

Notice that that we have to go through two steps of the original Euler’s method to get one improved Euler’s method approximation; however, the graphic above seems to indicate that the process is far more accurate than is the original Euler’s method. Indeed, zooming in quite close to the actual point $y(x_1)$, we see that the improved Euler’s method did quite a nice job of approximating:

The improved Euler’s method algorithm for approximating a solution curve to the initial value problem $y' = f(x, y), \ y(x_0) = y_0$ is summarized below:
Section 2.5

- Choose step size \( h \)

- Iterate the process of choosing points on the approximation curve:

  1. Calculate intermediate values:
     i. \( k_{n+1} = f(x_n, y_n) \)
     ii. \( u_{n+1} = y_n + h \cdot k_{n+1} \)
     iii. \( j_{n+1} = f(x_{n+1}, u_{n+1}) \)

  2. Calculate the \((n+1)\)st approximation on the approximation curve using the formulae

\[
y_{n+1} = y_n + h \cdot \left( \frac{k_{n+1} + j_{n+1}}{2} \right), \quad x_{n+1} = x_0 + (n+1)h
\]

  3. Connect the points \((x_n, y_n)\) and \((x_{n+1}, y_{n+1})\) with a line segment.

---

**Example.** In the last section, we looked at the initial value problem

\[
y' = 1 + y, \quad y(0) = 1,
\]

whose solution is

\[
y = 2e^x - 1,
\]

graphed below (and in the sequel) in red:

![Solution curve to the differential equation](image)

Using Euler's method with step size \( h = 1/2 \), we produced the approximation curve graphed in blue below:
Let's use the improved Euler method with step size \( h = \frac{1}{2} \) to produce another approximation curve for comparison. Recall that we think of
\[
y' = 1 + y \text{ as } f(x, y) = 1 + y;
\]
in addition, we know that the point \((0, 1)\) is on the actual solution curve, so we let \((x_0, y_0) = (0, 1)\) be the first point on our approximation curve.

We will use the chart below will help us keep track of all of this data:

\[
\begin{array}{c|c|c|c|c|c}
  i & x_i & k_i & u_i & j_i & y_i \\
  \hline
  0 & 0 & & & & 1 \\
\end{array}
\]

Since the process is tedious, we will calculate the first few points on the approximation curve produced by the improved Euler’s method, then use Mathematica to generate more points and a graph for comparison to the actual solution curve.

Since the step size is \( h = \frac{1}{2}, x_1 = \frac{1}{2} \). To get \( y_1 \), we must calculate the intermediate values \( k_1, u_1\), and \( j_1 \):

\[
\begin{align*}
k_1 &= f(x_0, y_0) = 2 \\
u_1 &= y_0 + \frac{1}{2} k_1 = 2 \\
j_1 &= f(x_1, u_1) = 3
\end{align*}
\]

So
\[
y_1 = y_0 + \frac{1}{2} \cdot \frac{k_1 + j_1}{2} = \frac{9}{4},
\]
which means that \((x_1, y_1) = (1/2, 9/4)\) is the first approximation point. We record the data below:

\[
\begin{array}{c|c|c|c|c|c}
  i & x_i & k_i & u_i & j_i & y_i \\
  \hline
  0 & 0 & & & & 1 \\
  1 & 1/2 & 2 & 2 & 3 & 9/4 \\
\end{array}
\]

Next, we must calculate the point \((x_2, y_2);\) clearly \( x_2 = 1 \). To get \( y_2 \), we again calculate the intermediate values \( k_2, u_2\), and \( j_2 \):
\[ k_2 = f(x_1, y_1) = 13/4 \]
\[ u_2 = y_1 + \frac{1}{2} k_2 = 31/8 \]
\[ j_2 = f(x_2, u_2) = 39/8 \]

So
\[ y_2 = y_1 + \frac{1}{2} \cdot \frac{k_2 + j_2}{2} = \frac{137}{32}, \]
which means that \((x_2, y_2) = (1, 137/32)\) is the second approximation point. We record the data below:

<table>
<thead>
<tr>
<th>(i)</th>
<th>(x_i)</th>
<th>(k_i)</th>
<th>(u_i)</th>
<th>(j_i)</th>
<th>(y_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>9/4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>13/4</td>
<td>31/8</td>
<td>39/8</td>
<td>137/32</td>
</tr>
</tbody>
</table>

Using the same process, we may fill in the row \(i = 3\) to see that the point \((x_3, y_3) = (3/2, 1393/128)\) is our third approximation point:

<table>
<thead>
<tr>
<th>(i)</th>
<th>(x_i)</th>
<th>(k_i)</th>
<th>(u_i)</th>
<th>(j_i)</th>
<th>(y_i)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1/2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>9/4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>13/4</td>
<td>31/8</td>
<td>39/8</td>
<td>137/32</td>
</tr>
<tr>
<td>3</td>
<td>3/2</td>
<td>169/32</td>
<td>443/64</td>
<td>507/64</td>
<td>1941/256</td>
</tr>
</tbody>
</table>

The curve produced by 12 iterations of the improved Euler’s method, \(h = 1/2\), is plotted below, along with the curve that we produced in the previous section (which also used 12 iterations and the same step size):

It is clear that the improved version of Euler’s method really did do a significantly better job of approximating the actual solution curve.
As with the original Euler’s method, the improved Euler’s method produces error. However, as you may have guessed, the improved method results in significantly less error. We will not state the theorem precisely, but in most cases, if \( y(x_n) \) is the actual value of the solution curve at \( x = x_n \), and \( y_n \) is the improved Euler’s method estimate, then there is a constant \( C \) so that

\[
|y(x_n) - y_n| \leq C h^2,
\]

i.e. the total error in the \( n \)th step of the improved Euler’s method is no more than a constant multiple of \( h^2 \), the square of the step size. Since we choose \( h \) to be small, and the square of a small number (close to 0) is also small, the improved method gives a much tighter bound on the distance between the actual values \( y(x_n) \) and the approximations \( x_n \).

In both the previous section and this one, we looked at an example with step size \( h = 1/2 \). Using Euler’s method, we know that the error produced is no more than a multiple of \( 1/2 = .5 \). However, using the improved version with the same step size, we know that the error produced is no more than a multiple of \( (1/2)^2 = .25 \).