Numerical Approximation of First-Order Differential Equations: An Exploration

Cole Johnson

April 2017

Introduction

Differential equations are mathematical equations that involve the derivatives of an unknown function. Differential equations of the first order are equations where the dependent variable is dependent both on the the independent variable and on its own derivative (or rate of change).

Differential equations are commonplace in the world and are extremely helpful in the describing physical relationships in a broad variety of fields. While the solutions to these equations can be found, the methods are often tedious and involve quite a bit of computation. It is convenient then to use numerical methods that can approximate the solution to the differential equation without the need to explicitly solve it.

One of the most basic methods devised to do this is Euler's Method. Euler's method seeks to find the solution at a point given the differential equation and an initial value. It does this by approximating the value of the equation at many points between the initial point and the point of interest. The more of these points used, the closer the approximation will be to the exact solution.

To approximate each of the points, Euler's Method takes the slope from the previous point and uses it to calculate the next point. It then does this again for the next point and the next point and so on. Euler's Method is thus described as recursive since it uses previously calculated values in the calculation of current values.

Examination of Approximation Methods

Euler's Method is not the sole method of approximating differential equations and it is by no means the best. It has its flaws and limitations. For one, the process described is only useful for first-order differential equations where the differential term can be isolated from the rest of the terms in the equation. Euler's Method also suffers from poor accuracy if a sufficient amount of steps is not taken or the point of interest is significantly far away from the initial point.

Better numerical methods have since emerged that have improved on Euler's Method but still have the same basic idea. They calculate the value of the differential term from previously calculated values. The ones examined herein face the same requirement of isolation of the differential term, but show marked improvement in accuracy and convergence speed. (Convergence speed is the rate at which the approximation approaches the exact as the step size approaches zero.)

In addition to Euler's Method, this exploration will also examine the Improved Euler's Method and the Classical Fourth-Order Runge-Kutta Method. The recursive formulae for each of these methods are outlined below:

Given:

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

Euler's Method can be defined by the two recursive formulas:

$$x_{n+1} = x_n + h,$$

$$y_{n+1} = y_n + hf(x_n, y_n),$$
$$n = 0, 1, 2, ...$$

Similarly, the Improved Euler's Method can be represented:

$$x_{n+1} = x_n + h,$$

$$y_{n+1} = y_n + \frac{h}{2}[f(x_n + h, y_n + hf(x_n, y_n)],$$
$$n = 0, 1, 2, ...$$

Lastly, the Classical Fourth-Order Runge-Kutta Method can be expressed by:

$$x_{n+1} = x_n + h,$$
$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$
$$n = 0, 1, 2, ...$$

where

$$k_1 = hf(x_n, y_n),$$
$$k_2 = hf(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}),$$
$$k_3 = hf(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}),$$
$$k_4 = hf(x_n + h, y_n + k_3)$$

Examining the differences

Perhaps the most basic and classic example of a differential equation is $y' = y$. As such, it seems suitable that this be the first differential equation that is examined herein.

To begin, first examine the analytical solution to the differential equation $y' = y$.

$$y' = y$$

can be written as

$$\frac{dy}{dx} = y$$

using separation of variables gives

$$\frac{1}{y}dy = dx$$

integrating both sides yields

$$\ln y = x$$

and finally solving for y gives

$$y = e^x$$

Now, to examine the numerical approximations through the different methods presented.

Initial values of $x = 0, y = 0$, a final value of $x = 1$ and 1000 iterations will be used to compare the outcomes of the three separate methods to the exact answer.

$x =$	$y=$	Euler's Method	Improved Euler's	Runge-Kutta
0	1	1	1	1
1	e	2.71692393224	2.71828137575	2.71828182846
% Error		.049954	1.6654 x 10^{-5}	3.6788 x 10^{-11}

$e = 2.718281828549$

While all of the methods are able to approximate the answer to less than half a percent, it is apparent that the accuracy improvement for each subsequent method is greater by several orders of magnitude.

Additional to these methods, the program also has code to approximate values based on the desired level of accuracy. Using the same equation and initial values as before but instead specifying an accuracy of .0001 yields the following results:

	Euler's Method	Improved Euler's	Runge-Kutta
Iterations	16384	128	8

Here the speed of convergence of each method is illustratively shown by the step size necessary for the desired accuracy level. While this gives a good visualization is it not perfectly accurate. Euler's Method actually has a speed of convergence of (step size)^1. The Improved Euler's Method has a speed of convergence of (step size)^2, and the Classical Fourth-Order Runge-Kutta Method has a speed of convergence of (step size)^4.

Lastly, the extrapolation accuracy of each method can be evaluated. For this calculation the same initial values will be used with 1000 iterations but the final value will instead be $x = 100$.

$x =$	$y=$	Euler's Method	Improved Euler's	Runge-Kutta
0	1	1	1	1
100	e^{100}	2.46993 x 10^{41}	2.30292 x 10^{43}	2.68791 x 10^{43}
% Error		99.081	14.330	.0076675

$e^{100} = 2.688117142$ x 10^{43}

Once more, the accuracy and versatility of these methods is clearly displayed. It is for this reason that the method's described are intended to be utilized for relatively near values of x and display improved accuracy as the iteration count increases.

Test Cases

To further apply the methods and test the program, a few test cases are displayed here.

1. $y' - \frac{y}{x} = 2x + 1$

The first test case will be $y' - \frac{y}{x} = 2x+1$. The initial values will be $x = 1, y = 1$ and a final value of $x = 10$.

First, the analytical solution.

$$y' - \frac{y}{x} = 2x + 1$$

Gives as the exact solution

$$y = 2x^2 + x \ln|x| + Cx$$

Which with the initial values gives

$$y = 2x^2 + x \ln|x| - x$$

Inputting the same values and the rearranged equation $y' = \frac{y}{x} + 2x + 1$ and using 1000 iterations gives:

$x =$	$y=$	Euler's Method	Improved Euler's	Runge-Kutta
1	1	1	1	1
10	$190 + 10 \ln 10$	212.571680235	213.024992484	213.0258509
	% Error	.2132	.000403	0.00

$190 + 10 \ln 10 = 213.0258508$

2. $\frac{dy}{dx} = (x^2y - y)$

For the second case where $y' - \frac{y}{x} = 2x + 1$, initial values will be $x = 1, y = 1$ and a final value of $x = 5$.

Analytically,

$$y' - \frac{y}{x} = 2x + 1$$

Gives the exact solution

$$y = e^{\frac{1}{3}x^3 - x + C}$$

Where the initial values give

$$y = e^{\frac{1}{3}x^3 - x + \frac{2}{3}}$$

Inputting the same values and equation into the program and using 1000 iterations gives:

$x =$	$y=$	Euler's Method	Improved Euler's	Runge-Kutta
1	1	1	1	1
5	$e^{\frac{112}{3}}$	5.49579 x 10^{15}	1.59722 x 10^{16}	1.63553 x 10^{16}
% Error		66.3976	2.34313	.000690

$e^{\frac{112}{3}}$ = 1.63554 x 10^16

3. $y' = \sqrt{xy}$

The last test case will be $y' = \sqrt{xy}$. Initial values will again be $x = 1, y = 1$ but this time the final value will be $x = 10$.

First, the analytical solution.

$$y' = \sqrt{xy}$$

Gives as the exact solution

$$y = \frac{1}{9}x^3 + \frac{2}{3}x^{\frac{3}{2}}C + C^2$$

With the initial values gives

$$y = \frac{1}{9}x^3 + \frac{4}{9}x^{\frac{3}{2}} + \frac{4}{9}$$

Inputting the same values and equation into the program and using 1000 iterations gives:

$x =$	$y=$	Euler's Method	Improved Euler's	Runge-Kutta
1	1	1	1	1
10	$\frac{1}{9}(1000 + 4(1 + 10^{\frac{3}{2}}))$	125.0237	125.6098	125.6101
% Error		.28377	.0002663	0.00

$\frac{1000+4(1+10^{\frac{3}{2}})}{9} = 125.6101$

Conclusion

As a whole, numerical approximation is a useful technique when seeking the solution to a differential equation. With more powerful methods such as the Classical Fourth-Order Runge-Kutta Method, extremely accurate values can be obtained and frequently so can exact values. These methods are not without their limits though. Again, they are only useful for differential equations where the differential term is able to be isolated. Additionally, they lose accuracy when the point of interest is not close to the initial point. Furthermore, they are limited by the kinds of functions they can handle.

In comparison of the three methods, several observations are of note. Quite clearly, the Classical Fourth-Order Runge-Kutta Method gives the most accurate value and has the fastest speed of convergence. However, this method is not useful to be approximated by hand. Even the Improved Euler's Method would be immensely time consuming to approximate without technological aid. In this light, Euler's Method is actually superior. When access to a program that can run thousands of iterations or quickly perform the more complicated methods, it is the only realistically viable method.

Moreover, these methods also have an additional advantage over analytical solution. Certain differential equations are not simple to solve analytically. Even with multiple techniques to solve such equations, some equations simply are too complicated. Numerical approximation triumphs in this regard because it is not concerned with actually finding the equation of the solution but rather just the numerical solution at a point. This is perhaps the most important reason for developing numerical approximations. They allow for relatively quick solutions at a point, even when equations, initial values, points of interest, and desired accuracy change.