Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2022, 10, 865-876
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

A Comparison of Different Numerical Schemes to Solve Nonlinear First Order ODE

Mahjoub A. Elamin[1], Sami H. Altoum[2]

[1]Department of Mathematics, University College of Umluj, Tabuk University, Tabuk, Saudi Arabia
[2]Academy of Engineering Sciences, Umm Al-Qura University, Mecca, Saudi Arabia
Email: Malshaygi@ut.edu.sa, shtoum@uqu.edu.sa

How to cite this paper: Elamin, M.A. and Altoum, S.H. (2022) A Comparison of Different Numerical Schemes to Solve Nonlinear First Order ODE. *Journal of Applied Mathematics and Physics*, **10**, 865-876.
https://doi.org/10.4236/jamp.2022.103059

Received: January 18, 2022
Accepted: March 19, 2022
Published: March 21, 2022

Abstract

In this paper, the nonlinear first order ordinary differential equation will be considered. Three simplest numerical stencils are presented to solve this equation. We deduce that the numerical method of Trapezoidal is a good technique, which helped us to find an approximation of the exact solution with small error.

Keywords

Ordinary Differential Equation, Euler Method, Numerical Approximation

1. Introduction

Consider the nonlinear first order ordinary differential equation

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

Ordinary differential equations occur in many scientific disciplines, including physics, chemistry, biology, and economics. In mathematics, an ordinary differential equation (ODE) is a differential equation containing one or more functions of one independent variable and the derivatives of those functions. Ordinary differential equations (ODEs) arise in many contexts of mathematics and social and natural sciences. Mathematical descriptions of change use differentials and derivatives. Various differentials, derivatives, and functions become related via equations, such that a differential equation is a result that describes dynamically changing phenomena, evolution, and variation. Some ODEs can be solved explicitly in terms of known functions and integrals. When that is not possible, the equation for computing the Taylor series of the solutions may be useful. For applied problems, numerical methods for ordinary differential equations can

supply an approximation of the solution. In addition, some methods in numerical partial differential equations convert the partial differential equation into an ordinary differential equation, which must then be solved. Euler method (also called forward Euler method) is a first-order numerical procedure for solving ordinary differential equations (ODEs) with a given initial value. It is the most basic explicit method for numerical integration of ordinary differential equations and is the simplest Runge-Kutta method [1] [2] [3]. The backward Euler method (or implicit Euler method) is one of the most basic numerical methods for the solution of ordinary differential equations, the backward Euler method has order one. This means that the local truncation error (defined as the error made in one step) is $O\left(h^2\right)$. The error at a specific time t is $O\left(h\right)$. It is similar to the (standard) Euler method, but differs in that it is an implicit method. This differs from the (forward) Euler method in that the latter uses $f\left(t_k, y_k\right)$ in place of $f\left(t_{k+1}, y_{k+1}\right)$. The backward Euler method is an implicit method: the new approximation y_{k+1} appears on both sides of the equation, and thus the method needs to solve an algebraic equation for the unknown y_{k+1}. For non-stiff problems, this can be done with fixed-point iteration [4] [5] [6] [7]. This paper is organized as follows: In Chapter 2 we presented three simplest numerical stencils. In Chapter 3 we introduce a numerical formulation of ODE and apply examples.

2. Three Simplest Numerical Stencils

2.1. Forward Euler Stencil

The purpose of this paper is to derive the three simplest numerical stencils to solve the first order equation

$$D\left(y\right)\left(x\right) = f\left(x, y\left(x\right)\right). \tag{2}$$

In this expression, f is assumed to be a known function of the independent variable x and the function that we are trying to solve for $y\left(x\right)$. The simplest numerical stencils to solve this equation will give us an approximation to y at some point $x = X + h$ given some knowledge of y at $x = X$. All of these stencils are based on the Taylor series approximation for $y\left(x\right)$ about $x = X$ to linear order:

$$y\left(x\right) = y\left(X\right) + D\left(y\right)\left(X\right)\left(x - X\right) + O\left(\left(x - X\right)^2\right). \tag{3}$$

Let us define h as the difference between x and X, and then get rid of x in the above expression:

$$h = x - X$$

$$y\left(h + X\right) = y\left(X\right) + D\left(y\right)\left(X\right)h + O\left(h^2\right). \tag{4}$$

Now, we can remove the first derivative of y by making use of the differential equation:

$$D\left(y\right)\left(X\right) = f\left(X, y\left(X\right)\right)$$

$$y(X+h)=y(X)+f\left(X,y(X)\right)h+O\left(h^2\right) \tag{5}$$

The forward Euler algorithm involves discarding the terms of order h^2 and higher in this expression and hence obtaining an approximation to $y(X+h)$ in terms of $y(X)$. We can accomplish this by converting the above series into a polynomial using the convert/polynom command. Also, we can take $X=x_i$ as the i^{th} point on an evenly spaced lattice $x_i=x_0+ih$, where h is the lattice spacing and i is an integer; it then follows that "$X+h=x_{i+1}$". Furthermore, we label the numeric approximation of $y(x)$ at $x=x_i$ a "y_i and approx; $y(x_i)$". Implementing these steps and notational changes:

$$y(h+X)=y(X)+f\left(X,y(X)\right)h$$

$$\left[y(h+X)=y_{i+1},y(X)=y_i,X=x_i\right]$$

$$y_{i+1}=y_i+f\left(x_i,y_i\right)h \tag{6}$$

The last expression is the forward Euler stencil for solving ode. It is called an explicit algorithm because the quantity we wish to calculate y_{i+1} is given explicitly in terms of y_i.

2.2. Backward Euler Stencil

The backward Euler stencil is obtained in a similar fashion, except we identify h, y_i, and y_{i+1} differently:

$$h=X-x$$

$$y(-h+X)=y(X)-\left(D(y)\right)(X)h+O\left(\left(-h\right)^2\right)$$

$$y(-h+X)=y(X)-f\left(X,y(X)\right)h+O\left(\left(-h\right)^2\right)$$

$$y(-h+X)=y(X)-f\left(X,y(X)\right)h$$

$$y(-h+X)=y_i,y(X)=y_{i+1},X=x_{i+1}$$

$$y_i=y_{i+1}-f\left(x_{i+1},y_{i+1}\right)h. \tag{7}$$

This stencil is explicit because it will not in general be possible to algebraically isolate what we want to calculate, *i.e.* $y_{i+1}=(x_{i+1})$, except for very specific forms of f. We can rearrange the stencils to better demonstrate their geometric meaning:

$$f\left(x_i,y_i\right)=\frac{-y_i+y_{i+1}}{h}$$

$$f\left(x_{i+1},y_{i+1}\right)=\left(-y_i+y_{i+1}\right)/h. \tag{8}$$

These forms illustrate that the forward Euler stencil (first equation) is a simple approximation of the first derivative of $y(x)$ in the interval $x\in\left[x_i,x_{i+1}\right]$ using ode evaluated at the left hand side of the interval. Conversely, the backward Euler stencil uses the differential equation to evaluate the derivative at the righthand side of the interval [8] [9].

2.3. Trapezoidal Method

There is another stencil we can derive that is the average of these two approaches:

$$\frac{1}{2}f\left(x_i,y_i\right)+\frac{1}{2}f\left(x_{i+1},y_{i+1}\right)=\frac{-y_i+y_{i+1}}{h}. \tag{9}$$

This is called the trapezoidal method, and it uses ode evaluated at both ends of the interval to approximate $y'(x)$. Like the backward Euler stencil, it represents an implicit scheme.

3. Numerical Formulation

As an example, we can look at what these stencils look like for the special case of $f(x,y)=\lambda y$, where λ is a constant:

$$\lambda y_i=\frac{-y_i+y_{i+1}}{h} \text{ Forward Euler}$$

$$\lambda y_{i+1}=\frac{-y_i+y_{i+1}}{h} \text{ Backward Euler}$$

$$\lambda y_i+\frac{1}{2}\lambda y_{i+1}=\frac{-y_i+y_{i+1}}{h} \text{ Trapezoidal.}$$

For this special case, we see that it is possible to isolate y_{i+1} for each stencil. Let's do this using a loop:

$$y_{i+1}=y_i\left(h\lambda+1\right) \text{ Forward Euler}$$

$$y_{i+1}=-\frac{y_i}{h\lambda-1} \text{ Backward Euler}$$

$$y_{i+1}=-\frac{y_i\left(h\lambda+2\right)}{h\lambda-2} \text{ Trapezoidal.}$$

Now, lets go back to the general case by undefined f:

$$f\left(x_i,y_i\right)=\left(-y_i+y_{i+1}\right)/h$$

$$f\left(x_{i+1},y_{i+1}\right)=\left(-y_i+y_{i+1}\right)/h$$

$$(1/2)f\left(x_i,y_i\right)+(1/2)f\left(x_{i+1},y_{i+1}\right)=\left(-y_i+y_{i+1}\right)/h.$$

We now want to determine what the error is in each stencil. To do this, we need to rewrite each of them in the standard form

$$y_{i+1}-y_i-h\Phi\left(x_i,x_{i+1},y_i,y_{i+1}\right)=0$$

therefore,

$$y_{i+1}-y_i-f\left(x_i,y_i\right)h=0$$

$$y_{i+1}-f\left(x_{i+1},y_{i+1}\right)h-y_i=0$$

$$y_{i+1}-h\left((1/2)*f\left(x_i,y_i\right)+(1/2)f\left(x_{i+1},y_{i+1}\right)\right)-y_i=0.$$

Note that these relations define our numeric stencils in these sense that they give exact relations between the approximations y_i and y_{i+1}. But if we replace

the approximations with the exact values of $y(x)$ they are supposed to represent, the above will represent only approximate equalities. That is, if we make the changes $x_i \to x$, $x_{i+1} \to x+h$, $y_i \to y(x)$, $y_{i+1} \to y(x+h)$, the left hand sides of the above equations will only be approximately equal to zero. We call the magnitude of the discrepancy the "one step error" or "local error" in the stencil. Hence, the one step error in the various stencils will be:

$$y_i = y(x), y_{i+1} = y(x+h), x_i = x, x_{i+1} = x+h$$

$$\text{Error Forward Euler} = y(x+h) - y(x) - f(x, y(x))h$$

$$\text{Error Backward Euler} = y(x+h) - y(x) - f(x, y(x))h$$

$$\text{Error Trapezoidal} = y(x+h) - f(x+h, y(x+h))h - y(x)$$

To estimate the magnitudes of these errors, we expand each of the above expression as a power series about $h=0$ (since we are implicitly assuming h is a small quantity):

$$\text{Error Forward Euler} = \left(-f(x, y(x)) + D(y)(x)\right)h + 1/2\left(D^{(2)}\right)(y)(x)h^2 + 1/6\left(D^{(3)}\right)(y)(x)h^3 + O\left(h^4\right)$$

$$\text{Error Backward Euler} = \left(-f(x, y(x)) + D(y)(x)\right)h + (\alpha_1)h^2 + \alpha_2 - \alpha_3 + O\left(h^4\right)$$

where,

$$\alpha_1 = 1/2\left(D^{(2)}\right)(y)(x) - D_1(f)(x, y(x)) - D_2(f)(x, y(x))D(y)(x)$$

$$\alpha_2 = 1/6\left(D^{(3)}\right)(y)(x) - 1/2\left(D(y)(x)\right)^2(D_{2,2})(f)(x, y(x)) - 1/2\, D_2(f)(x, y(x))\left(D^{(2)}\right)(y)(x)$$

$$\alpha_3 = (D_{1,2})(f)(x, y(x))D(y)(x) - 1/2(D_{1,1})(f)(x, y((x)))h^3$$

$$\text{Error Trapezoidal} = (\beta_1)h + (\beta_2)h^2 + (\beta_3 - \beta_4)h^3 + O\left(h^4\right)$$

where,

$$\beta_1 = -f(x, y(x)) + D(y)(x)$$

$$\beta_2 = 1/2\left(D^{(2)}\right)(y)(x) - 1/2\, D_1(f)(x, y(x)) - 1/2\, D_2(f)(x, y(x))D(y)(x)$$

$$\beta_3 = 1/6\left(D^{(3)}\right)(y)(x) - 1/4\left(D(y)(x)\right)^2(D_{2,2})(f)(x, y(x))$$

$$\beta_4 = 1/4\, D_2(f)(x, y(x))\left(D^{(2)}\right)(y)(x) - 1/2(D_{1,2})(f)(x, y(x))D(y)(x) - 1/4(D_{1,1})(f)(x, y(x)).$$

Notice that first, second and third derivatives of y appear explicitly in these expressions. The first derivative terms can be removed by making use of ode. The second derivative can also be removed by examining the derivative of ode:

$$\left(D^{(2)}\right)(y)(x) = D_1(f)(x, y(x)) + D_2(f)(x, y(x))\frac{\mathrm{d}}{\mathrm{d}x}y(x)$$

and

$$\frac{\mathrm{d}}{\mathrm{d}x}y(x)=f\left(x,y(x)\right)$$

$$\left(D^{(2)}\right)(y)(x)=D_1(f)\left(x,y(x)\right)+D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right).$$

Similarly, the third derivative can be removed

$$\begin{aligned}\left(D^{(3)}\right)(y)(x)&=\left(D_{1,1}\right)(f)\left(x,y(x)\right)+\left(D_{1,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)\\&+\left(\left(D_{1,2}\right)(f)\left(x,y(x)\right)+\left(D_{2,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)f\left(x,y(x)\right)\\&+D_2(f)\left(x,y(x)\right)\left(D_1(f)\left(x,y(x)\right)+D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right).\end{aligned}$$

We now substitute our formulae for the derivatives of y into the error expressions:

$$\text{Error Forward Euler}=\gamma_1 O\left(h^2\right)+\left(\gamma_2+\gamma_3+\gamma_4\right)O\left(h^3\right)+O\left(h^4\right) \tag{10}$$

where,

$$\gamma_1=\left(1/2\,D_1(f)\left(x,y(x)\right)+1/2\,D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)$$

$$\gamma_2=1/6\left(D_{1,1}\right)(f)\left(x,y(x)\right)+1/6\left(D_{1,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)$$

$$\gamma_3=1/6\left(\left(D_{1,2}\right)(f)\left(x,y(x)\right)+\left(D_{2,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)f\left(x,y(x)\right)$$

$$\gamma_4=1/6\,D_2(f)\left(x,y(x)\right)\left(D_1(f)\left(x,y(x)\right)+D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)$$

$$\text{Error Backward Euler}=\gamma_5\left(h^2\right)+\left(\gamma_6+\gamma_7-\gamma_8-\gamma_9\right)h^3+O\left(h^4\right) \tag{11}$$

where,

$$\gamma_5=-\left(1/2\,D_1(f)\left(x,y(x)\right)-1/2\,D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)$$

$$\gamma_6=-1/3\left(D_{1,1}\right)(f)\left(x,y(x)\right)-5/6\left(D_{1,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)$$

$$\gamma_7=1/6\left(\left(D_{1,2}\right)(f)\left(x,y(x)\right)+\left(D_{2,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)f\left(x,y(x)\right)$$

$$\gamma_8=1/3\,D_2(f)\left(x,y(x)\right)\left(D_1(f)\left(x,y(x)\right)+D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)$$

$$\gamma_9=1/2\left(f\left(x,y(x)\right)\right)^2\left(D_{2,2}\right)(f)\left(x,y(x)\right)$$

$$\text{Error Trapezoidal}=\left(\gamma_{10}+\gamma_{11}-\gamma_{12}-\gamma_{13}\right)h^3+O\left(h^4\right) \tag{12}$$

where,

$$\gamma_{10}=-1/12\left(D_{1,1}\right)(f)\left(x,y(x)\right)-1/3\left(D_{1,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)$$

$$\gamma_{11}=1/6\left(\left(D_{1,2}\right)(f)\left(x,y(x)\right)+\left(D_{2,2}\right)(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)f\left(x,y(x)\right)$$

$$\gamma_{12}=1/12\,D_2(f)\left(x,y(x)\right)\left(D_1(f)\left(x,y(x)\right)+D_2(f)\left(x,y(x)\right)f\left(x,y(x)\right)\right)$$

$$\gamma_{13}=1/4\left(f\left(x,y(x)\right)\right)^2\left(D_{2,2}\right)(f)\left(x,y(x)\right).$$

We see that the local error for the forward and backward schemes is $O\left(h^2\right)$.

Furthermore, the leading order terms for those stencils are the negative of the other one. Since the trapezoidal scheme is the average of the forward and backward algorithms, this explains why the leading order error for the trapezoidal scheme is $O\left(h^3\right)$. In other words, the trapezoidal scheme is more accurate than the other two. Note that if all we are after is the magnitude of the leading order terms in the errors, we can just expand the above in a low order series:

$$\text{Error Forward Euler} = O\left(h^2\right) \tag{13}$$

$$\text{Error Backward Euler} = O\left(h^2\right) \tag{14}$$

$$\text{Error Trapezoidal} = O\left(h^3\right). \tag{15}$$

We now turn our attention to actual numerical algorithms employing these stencils. Since the forward Euler scheme is explicit, it is particularly easy to develop some code for it that works for arbitrary functions *f*. It is useful to rewrite the stencil with y_{i+1} isolated on the LHS:

$$y_{i+1} = y_i + f\left(x_i, y_i\right)h$$

Now we can transcribe this as a mapping that takes the step size and the old values of y_i and x_i and returns the new value y_{i+1}
$y_{\text{new}} := \left(h, x_{\text{old}}, y_{\text{old}}\right) \to y_{\text{old}} + f\left(x_{\text{old}}, y_{\text{old}}\right)h$ We could have equivalently accomplished this by using the unapply command, which converts expression into mappings without having to re-write them manually:
$y_{\text{new}} := \left(h, y^2, y^3\right) \to y^3 + f\left(y^2, y^3\right)h$ Here is a procedure that calculates the numerical solution of ode in the interval $x \in [0,1]$ once the form of $f(x,y)$ has been fixed. Its arguments are initial data in the form $y(0) = y_0$ and the number of steps *N*. The output is a list of points, which we compare in the plot to the output generated by numeric for the same problem.

Example: Solve the equation $f := (x,y) \to -y^2 + 2x$

$$D(y)(x) = -\left(y(x)\right)^2 + 2x,\ y(0) := 2,\ N := 15$$

Solution:

$$\begin{aligned}
&\text{data} := [0,2],[0.06666666667,1.733333333],\\
&[0.1333333333,1.541925926],[0.2000000000,1.401201333],\\
&[0.2666666667,1.296976988],[0.3333333334,1.220389256],\\
&[0.4000000001,1.165543705],[0.4666666668,1.128310896],\\
&[0.5333333335,1.105660753],[0.6000000002,1.095272817],\\
&[0.6666666669,1.095297981],[0.7333333336,1.104208359],\\
&[0.8000000003,1.120701063],[0.8666666670,1.143636338],\\
&[0.9333333337,1.171998289],[1.000000000,1.204870734]
\end{aligned}$$

Of course Figure 1 it is really comparing the results of two numeric approximations to the true solution of the problem. It is also useful to compare numeric answers to analytic results (if available). For the above choice of $f(x,y)$ there is an analytic solution in terms of Airy functions:

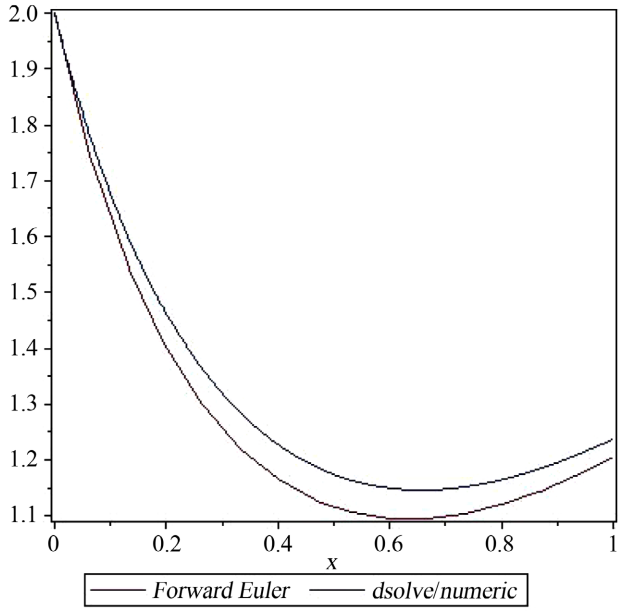

Figure 1. Comparison between forward euler and dsolve numeric.

$$\text{Analytical solution} := \frac{2^{\frac{1}{3}}\left(-\frac{\xi_1}{\xi_2} + AiryBi\left(1, \sqrt[3]{2}x\right)\right)}{-\frac{\xi_3}{\xi_4} + AiryBi\left(\sqrt[3]{2}x\right)} \tag{16}$$

where

$$\xi_1 = -AiryAi\left(1, 2^{\frac{1}{3}}x\right)\left((4)3^{5/6}\pi - 3\Gamma(2/3)^2 3^{2/3}2^{1/3}\right)\Big/\left((3)3^{1/6}\Gamma(2/3)^2 2^{1/3} + 4\pi 3^{1/3}\right)$$

$$\xi_2 = 3\sqrt[6]{3}\left(\Gamma(2/3)\right)^2\sqrt[3]{2} + 4\pi\sqrt[3]{3}$$

$$\xi_3 = \left((4)3^{5/6}\pi - 3\left(\Gamma(2/3)\right)^2 3^{2/3}\sqrt[3]{2}\right)AiryAi\left(\sqrt[3]{2}x\right)$$

$$\xi_4 = 3\sqrt[6]{3}\left(\Gamma(2/3)\right)^2\sqrt[3]{2} + 4\pi\sqrt[3]{3}.$$

Here is some code that generates a movie comparing how the numerical solution converges to the analytic solution as the number of cells is increased:

We would also like to examine the performance of the backward and trapezoidal stencils see **Figure 2**, so let's make a simple choice of $f(x,y)$ that admits an analytic solution allows each stencil to be solve for y_{i+1} explicitly:

$$f := (x,y) \to \lambda y,\ y(0) := y_0$$

$$D(y)(x) = \lambda y(x)$$

$$\text{Analytic Solution} := y(0)e^{\lambda x}$$

$$\text{Forward Euler}: y_{i+1} = h\lambda y_i + y_i$$

$$\text{Backward Euler}: y_{i+1} = \frac{y_i}{-h\lambda + 1}$$

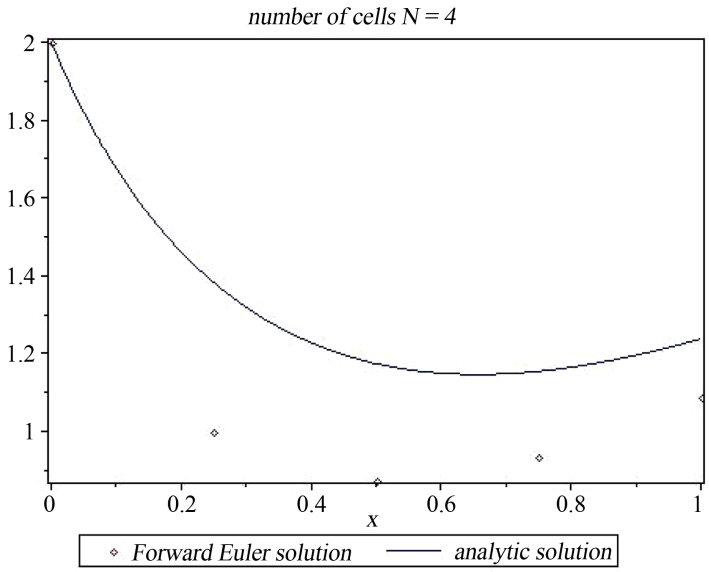

Figure 2. Comparison between backward and trapezoidal and dsolve numeric.

$$\text{Trapezoidal}: y_{i+1} = \frac{y_i + 1/2\, h\lambda y_i}{1 - 1/2\, h\lambda}.$$

As for the above code, it will be useful to realize each stencil as a mapping with arguments h, λ and y_i. We can do this using a loop and the unapply command:

$$\text{stencil}_1 := (h, \lambda, y_3) \to h\lambda y_3 + y_3$$

$$\text{stencil}_2 := (h, \lambda, y_3) \to \frac{y_3}{-h\lambda + 1}$$

$$\text{stencil}_3 := (h, \lambda, y_3) \to \frac{y_3 + 1/2\, h\lambda y_3}{1 - 1/2\, h\lambda}.$$

Conversely, we could have accomplished the same thing in one line by using the map command, which applies the same mapping onto each element of a list (in this case we are applying operations onto each element of Stencils to generate a new list stencil):

$$\text{stencil} := (h, \lambda, y_3) \to (h\lambda y_3) + y_3,$$

$$(h, \lambda, y_3) \to \frac{y_3}{-h\lambda + 1},$$

$$(h, \lambda, y_3) \to \frac{y_3 + 1/2\, h\lambda y_3}{1 - 1/2\, h\lambda}.$$

It is interesting to note that even though each of the stencils give different expressions for y_{i+1}, they actually agree with one another to order (h^2). To see this, let's perform some Taylor series expansions:

$$\text{ForwardEuler}\ y_{i+1} = h\lambda y_i + y_i$$

$$\text{BackwardEuler}\ y_{i+1} = y_i + h\lambda y_i + O\left(h^2\right)$$

$$\text{Trapezoidal } y_{i+1} = y_i + \lambda y_i h + O\left(h^2\right).$$

Now, here is a procedure Euler solution that calculates a numeric solution for $y(x)$ for $x \in [0,1]$ using N cells and assuming $y(0) = y_0$ and a given value of λ. The value of choice dictates which stencil is used: 1 for forward Euler, 2 for backward Euler, and 3 for trapezoidal. Here is a plot of the numeric output for each stencil compared to the actually solution (because we are going to plot 4 curves on our graph). It seems as if the trapezoidal method performs much better than the other two. $N := 15$, $y_0 := 2$, $\lambda := -4$.

We can quantify exactly how well the various stencils are doing by comparing the numeric and actual values for y at some fixed value of x, say $x = 1$. We call this the global error in the numeric solution at $x = 1$, which we denote by ε In this procedure, note that the $[N+1][2]$ suffix after Euler solution (y_0, N, λ) has the effect of picking out the last element of the list (which is itself a list of 2 quantities), and then picking out the second element of that sub-list (which is the numerical answer for $y(1)$. We now generate a log-log plot of the global errors for each stencil as a function of N.

For $N \geq 100$, the error curves look linear on this log-log plot. This implies that above some threshold number of steps, the errors obey an approximate power law $\varepsilon \simeq aN^b$ where a and b are constants. We can determine the values of these parameters by fitting a power law to our error data using statistics. Figure 3 and Figure 4 illustrate four solutions Forward Euler, Backward Euler, Trapezoidal and analytic solution and their errors.

(*PowerFit*). We first need to regenerate our data for $N \geq 100$, the range for which we believe a power law ought to be valid. Then, we use the Statistics (*PowerFit*) command to perform the fit. Note that this function requires the input to be a matrix, which is why we use the convert Matrix structure. The raw output

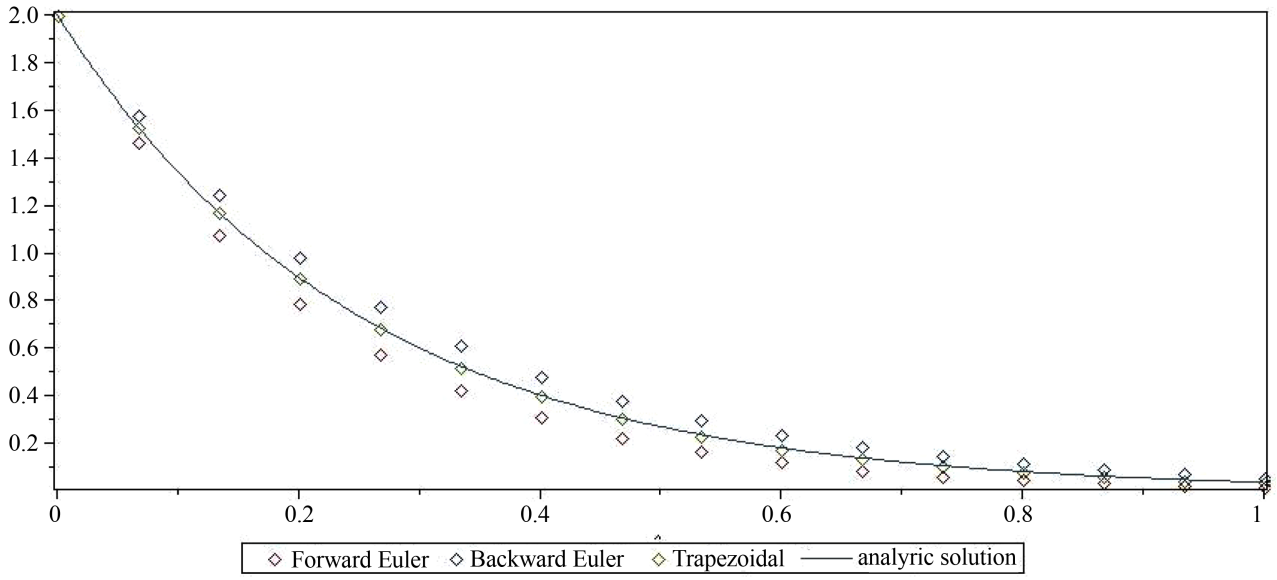

Figure 3. Four solutions forward euler, backward euler, trapezoidal and analytic solution and dsolve numeric.

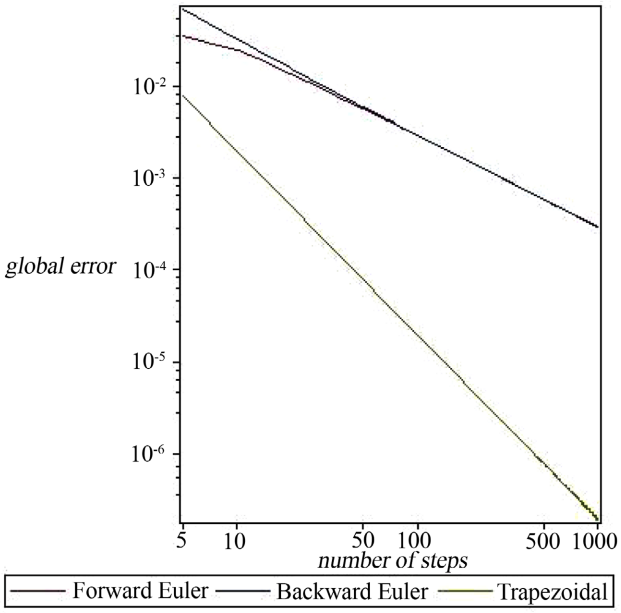

Figure 4. For $N \geq 100$, the error curve and dsolve numeric.

of the fitting algorithms is shown for each stencil, as well as the value of b in the power-law $\varepsilon \simeq aN^b$.

$$\text{Forward Euler} := \left[\frac{0.284665679953286}{N^{0.995851778662443}}, \begin{pmatrix} -1.25643984007394 \\ -0.995851778662443 \end{pmatrix}\right],$$

$$b = -0.995851778662443.$$

$$\text{Backward Euler} := \left[\frac{0.301341961047440}{N^{1.00397729890028}}, \begin{pmatrix} -1.19950957587335 \\ -1.00397729890028 \end{pmatrix}\right],$$

$$b = -1.00397729890028.$$

$$\text{Trapezoidal} := \left[\frac{0.196184023674760}{N^{2.00065878415391}}, \begin{pmatrix} -1.62870216393926 \\ -2.00065878415391 \end{pmatrix}\right],$$

$$b = -2.00065878415391.$$

We see that for asymptotically large numbers of steps $N \gg 1$, the global errors are $O\left(N^{-1}\right) = O(h)$ for the forward and backward Euler stencils, and $O\left(N^{-2}\right) = O\left(h^2\right)$ for the trapezoidal method. This confirms the general expectation that for a stencil with one step error $O\left(h^p\right)$, we expect the global error to obey

$$\text{global error} \sim (\text{number of steps}) \times (\text{one-step error for each step})$$
$$= N \times O\left(h^p\right) = O\left(h^{-1}\right) \times O\left(h^p\right) = O\left(h^{p-1}\right).$$

Recall that we showed in (13, 14, 15) above that $p = 2$ for forward and backward Euler, and $p = 3$ for the trapezoidal algorithm.

4. Conclusion

In this paper, we have investigated the solution of ODEs using three approaches.

The first one, is the Forward Euler method, the second is Backward Euler method and the third one is a numerical method based on Trapezoidal method. In general, we find the numerical method of Trapezoidal method, is a good technique that helped us to find an approximation of the exact solution with small error. In the last, we note the Trapezoidal method.

Acknowledgements

The author would like to thank the reviewers for their valuable comments which have improved the paper.

Conflicts of Interest

The authors declare that they have no competing interests.

References

[1] Altoum, S.H. (2018) Solution of Second Order Ordinary Differential Equation Associated with Toeplitz and Stiffness Matrices. *American Journal of Applied Sciences*, **15**, 416-422. https://doi.org/10.3844/ajassp.2018.416.422

[2] Altoum, S.H. (2018) Analytical and Three Numerical Approach to Solve Second Order ODEs. *International Journal of Advanced Scientific and Technical Research*, **4**, 51-67. https://doi.org/10.26808/rs.st.i8v4.06

[3] Williams, D. (1991) Probability with Martingales. Cambridge University Press, Cambridge, UK. https://doi.org/10.1017/CBO9780511813658

[4] Baker, I.N. and Rippon, P.J. (1983) Convergence of Infinite Exponentials. *Annales Academiac Scentiarium Fennicae*, *Series AI Mathematika*, **8**, 179-186. https://doi.org/10.5186/aasfm.1983.0805

[5] Feynman, R.P. (1986) Surely You're Joking, Mr. Feynman. Bantam Books, New York, USA, 71-72.

[6] Kuo, H.-H. (1996) White Noise Distribution Theory. CRC Press, Boca Raton.

[7] Rudin, W. (1976) Principles of Mathematical Analysis. McGraw Hill, USA.

[8] Euler, L. (1778) De formulis exponentialibus replicatus. *Acta Academiac Petropolitenae*, **1**, 38-60.

[9] Folland, G.B. (1999) Real Analysis: Modern Techniques and Their Applications. 2nd Edition, Wiley, Hoboken, USA.