Evaluation of Numerical IVP Methods

Lucius Anderson, David Thau, Harry Nguyen
Georgia State University Department of Mathematics and Statistics, and Department of Computer Science, February 2017

Sections:

- Introduction
  - Goals
- Mechanisms
  - Euler's method
  - Modified Euler's
  - Midpoint method
  - Runge Kutta Fourth order
  - Adams Bashforth four-step explicit
  - Predicator Corrector Method, using Adams-Bashforth 4-step explicit and Adams Moulton 3-step implicit scheme
- Implementation
  - Euler's Method
  - Modified Euler's
  - Midpoint Method
  - Runge Kutta Fourth Order
  - Adams Bashforth Four-Step Explicit
  - Predicator Corrector Method
  - Error
- Accuracy/Stability Analysis
  - Euler's Method
  - Modified Euler's
  - Midpoint Method
  - Runge Kutta Fourth Order
  - Adams Bashforth Four-step
  - Predictor Corrector Method
- Conclusion

Introduction

In this paper, we will be evaluating numerical methods for approximating ODE. From class we have discussed the various methods; Euler's Method, Modified Euler's Method, Midpoint Method, Runge-Kutta Fourth Order method, Adams-Bashforth 4-Step Explicit Method, and Predictor Correct Method using Adams-Bashforth 4-Step Explicit and Adams Moulton 3-Step Implicit Scheme. In this paper, using python, we will discuss and show how each method evaluates various Initial Value Problems, and then we will discuss the complexity, accuracy, and stability of each method.

Mechanisms

Here we will show the various numerical methods and their layout before evaluating them on functions. Each numerical method was translated into python for evaluation.

- Euler's Method:

$$w_0 = \alpha$$
$$w_{i+1} = w_i + hf(t_i, w_i), \quad i = 0, 1, \ldots, N-1$$

- Modified Euler's:

$$w_0 = \alpha$$
$$w_{i+1} = w_i + \frac{h}{2}\left(f(t_i, w_i) + f\left(t_{i+1}, w_i + h\big(f(t_i, w_i)\big)\right)\right)$$
$$i = 0, 1, \ldots, N-1$$

- Midpoint Method:

$$w_0 = \alpha$$
$$w_{i+1} = w_i + hf\left(t_i + \frac{h}{2}, w_i + \frac{h}{2}f(t_i, w_i)\right)$$
$$i = 0, 1, \ldots, N-1$$

- Runge Kutta (4th Order):

$$k_1 = f(t_i, w_i)$$
$$k_2 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_1)$$
$$k_3 = f(t_i + \frac{h}{2}, w_i + \frac{h}{2}k_2)$$
$$k_4 = f(t_{i+1}, w_i + hk_3)$$
$$w_{i+1} = w_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

- Adams-Bashforth 4-Step Explicit:

$$w_0 = \alpha\ , w_1 = \alpha_1 , w_2 = \ \alpha_2\ , w_3 = \ \alpha_3$$

$$w_{i+1} = \ w_i + \frac{h}{24}\left[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f\left(t_{i-2,} w_{i-2}\right) - 9f(t_{i-3}, w_{i-3})\right]$$

$$for\ i = 0{,}1\ , \dots,\ N - 1$$

- Adams 4th Order Predictor Method:

  With initial value $w_0\ =\ \alpha\ , w_1, w_2, w_3$ using the RK4 method.

  Then for $i = 0, 1, 2, \dots, N - 1$: Adams-Bashforth 4th Order explicit:

$$w_{i+1,\ p} = \ w_i + \frac{h}{24}\left[55f(t_i, w_i) - 59f(t_{i-1}, w_{i-1}) + 37f\left(t_{i-2,} w_{i-2}\right) - 9f(t_{i-3}, w_{i-3})\right]$$

  Using Adams-Mouton 3-step implicit method to get a corrector w_{i+1} :

$$w_{i+1} = \ w_i + \frac{h}{24}\left[9f\left(t_i, w_{i+1,\ p}\right) - 59f(t_{i-1}, w_{i-1}) + 37f\left(t_{i-2,} w_{i-2}\right) - 9f(t_{i-3}, w_{i-3})\right]$$

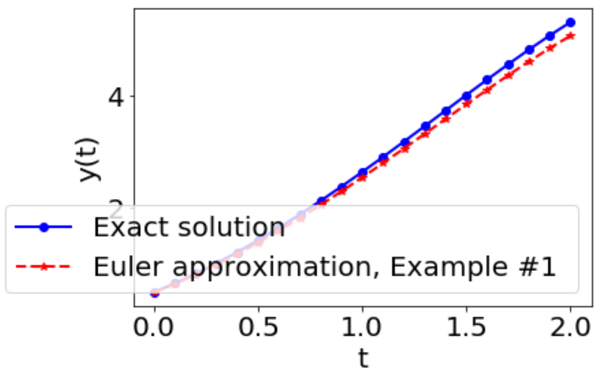
Implementation

For these various approximation methods were tested on four different functions. All obtained from the textbook, the various error obtained for each particularly function with different approximation method are shown here.

Euler's Method:

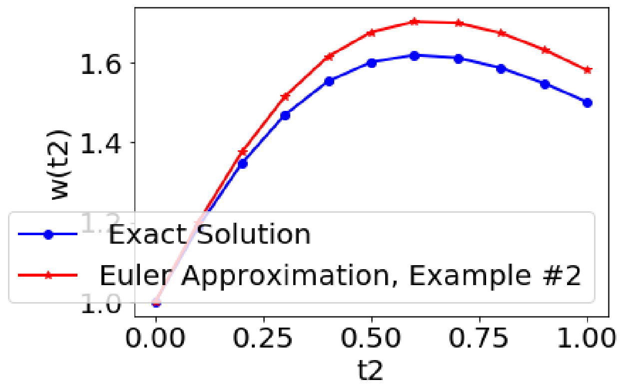
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$



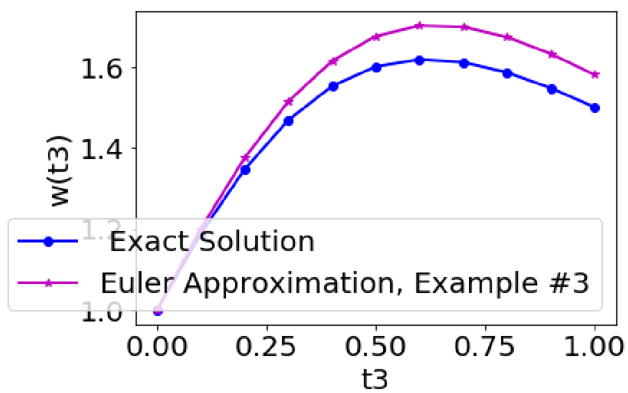
$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



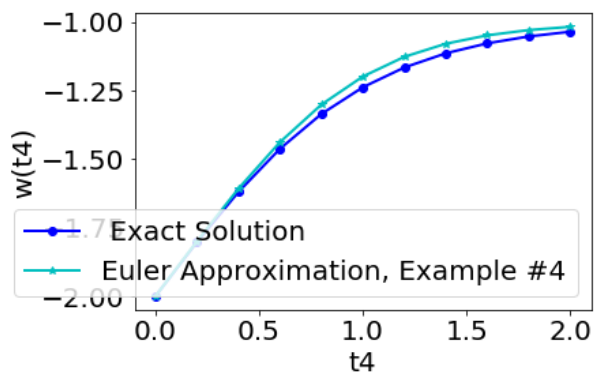
$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

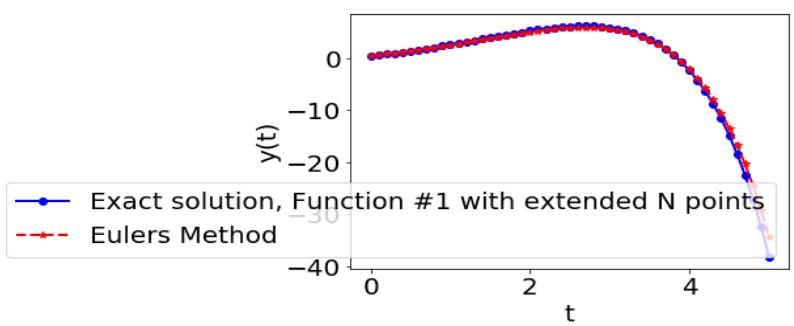
$Exact\ sol: y(t) = t(\tan(\ln(t)))$



$IVP: y' = -(y+1)(y+3)$

$Exact\ sol: y(t) = -3 + 2(1+e^{-2t})^{-1}$

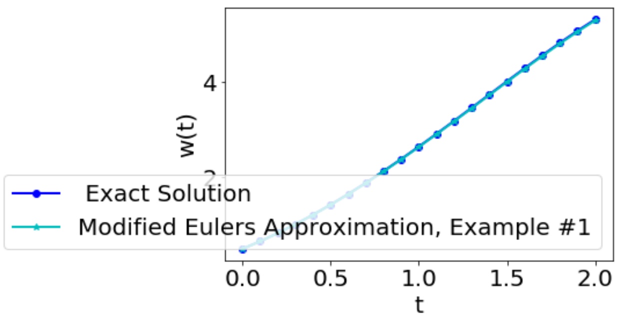




Modified Euler’s Method:

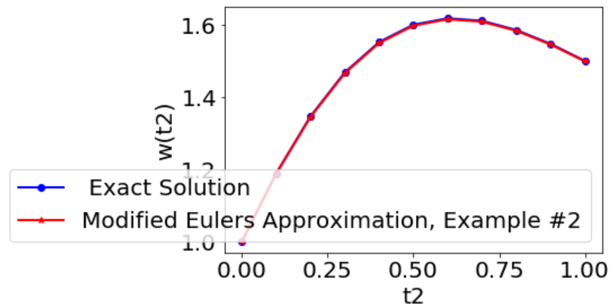
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$



$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



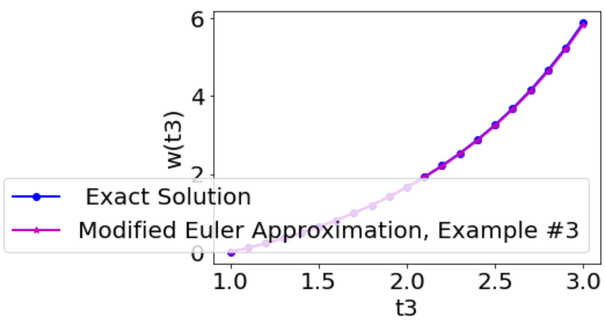
$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

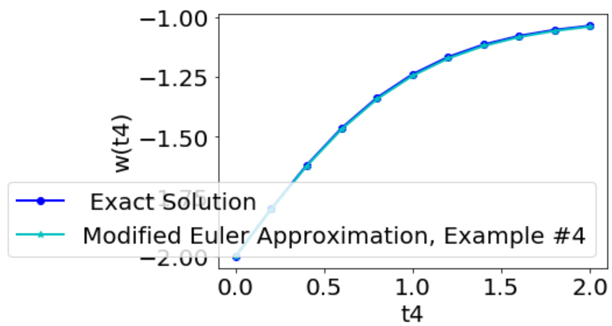
$Exact\ sol: y(t) = t(\tan(\ln(t)))$

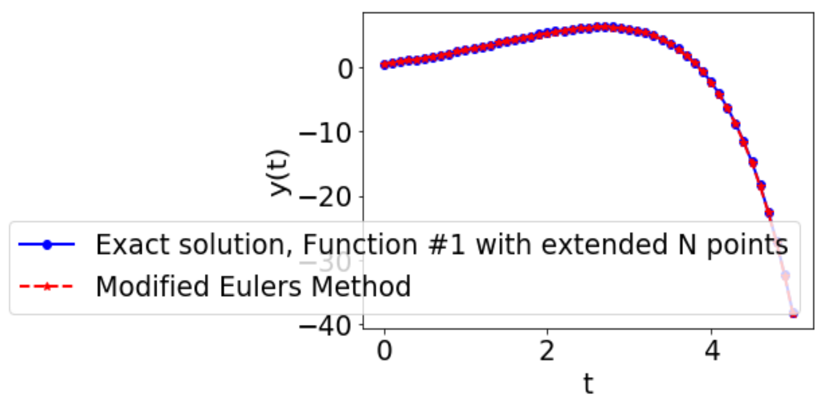
$IVP: y' = -(y+1)(y+3)$

$Exact\ sol: y(t) = -3 +$

$2(1 + e^{-2t})^{-1}$







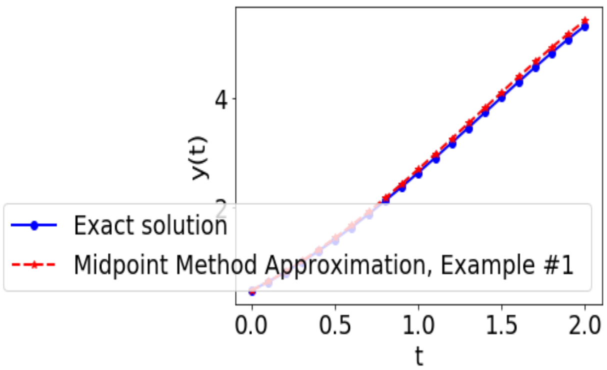
Midpoint Method

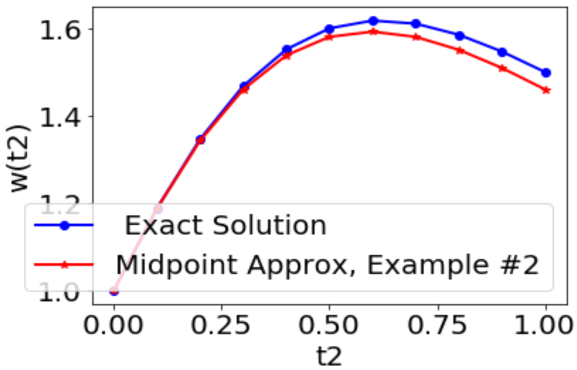
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$

$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



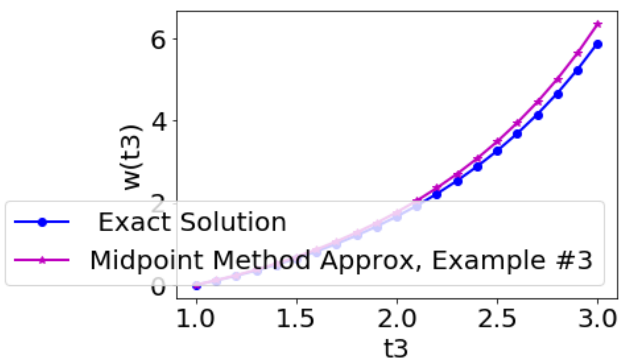


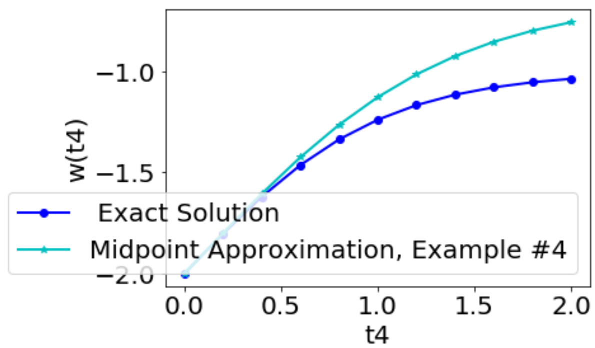
$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

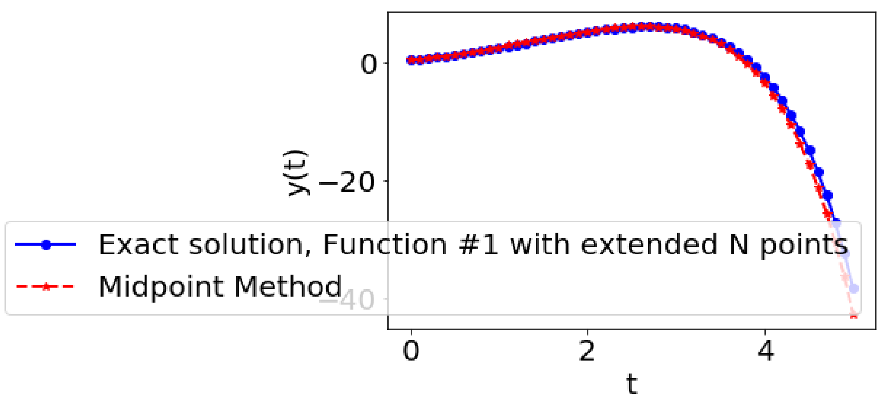
$Exact\ sol: y(t) = t(\tan(\ln(t)))$

$IVP: y' = -(y+1)(y+3)$

$Exact\ sol: y(t) = -3 + 2(1 + e^{-2t})^{-1}$



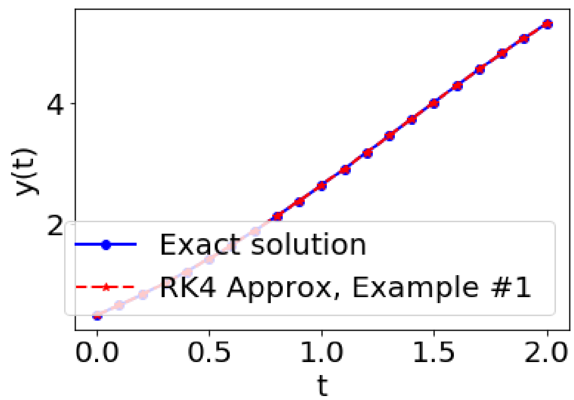




Runge-Kutta Fourth Order Method

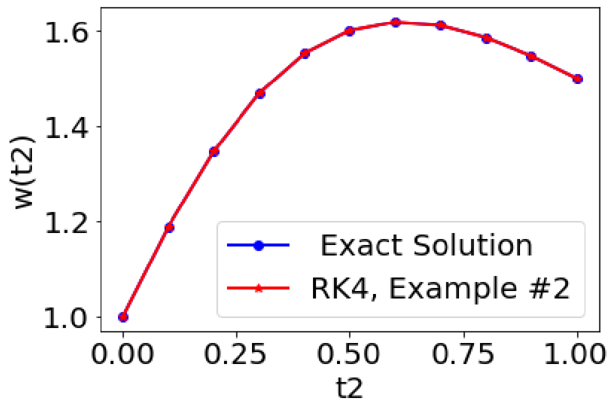
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$



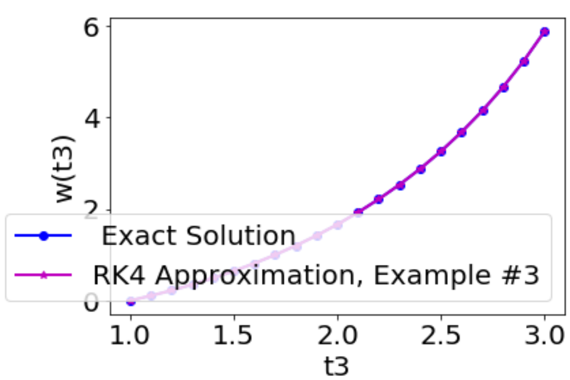
$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



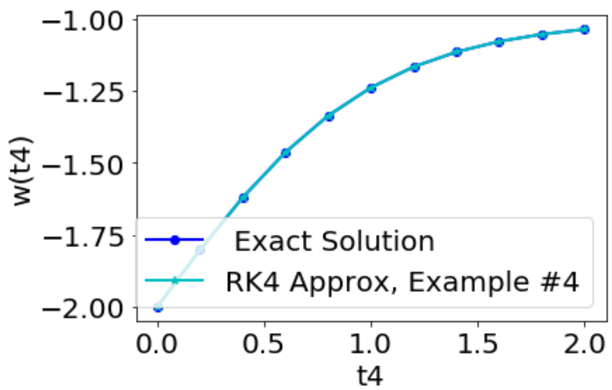
$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

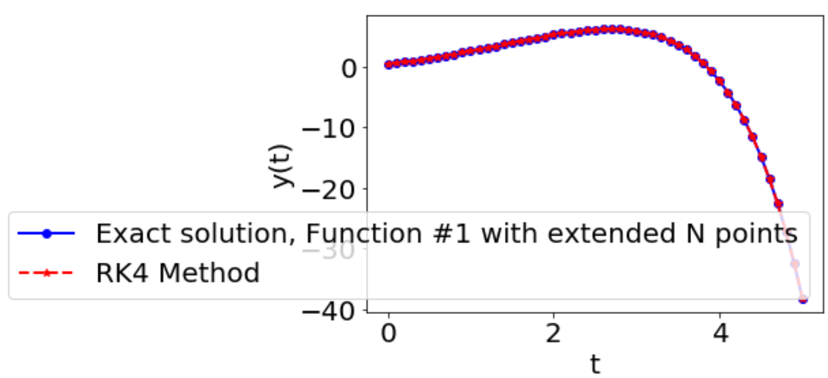
$Exact\ sol: y(t) = t(\tan(\ln(t)))$



$IVP: y' = -(y+1)(y+3)$

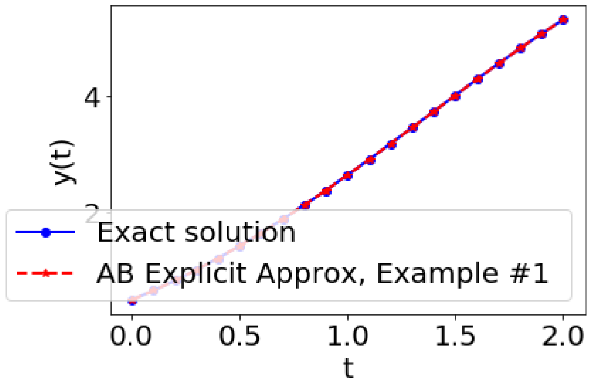
$Exact\ sol: y(t) = -3 + 2(1 + e^{-2t})^{-1}$





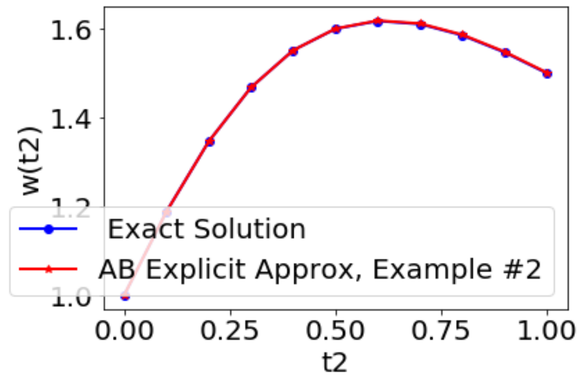
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$



$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



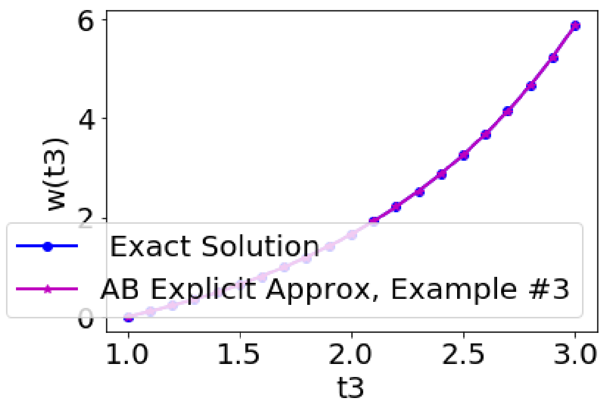
$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

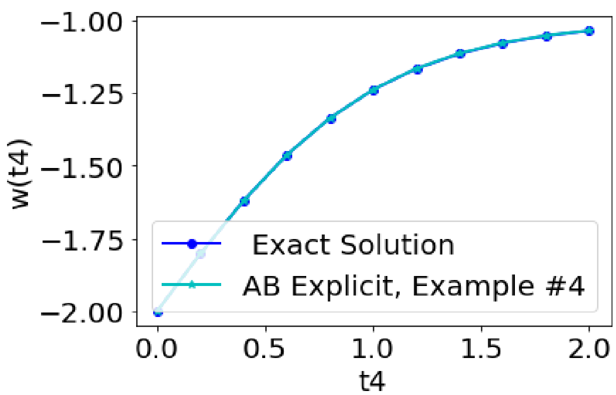
$Exact\ sol: y(t) = t(\tan(\ln(t)))$

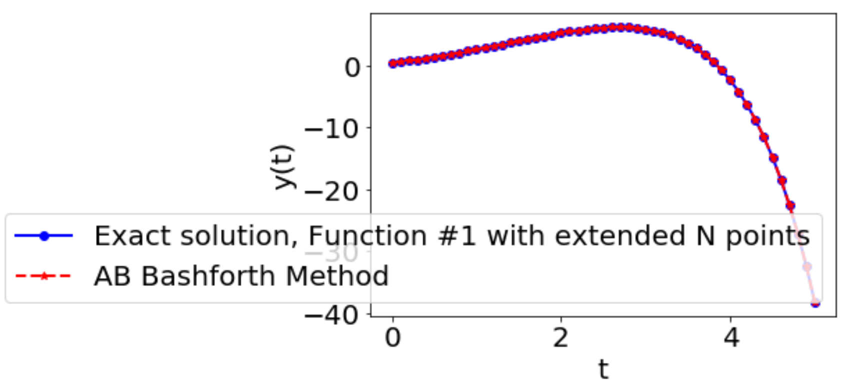
$IVP: y' = -(y+1)(y+3)$

$Exact\ sol: y(t) = -3 +$

$2(1 + e^{-2t})^{-1}$



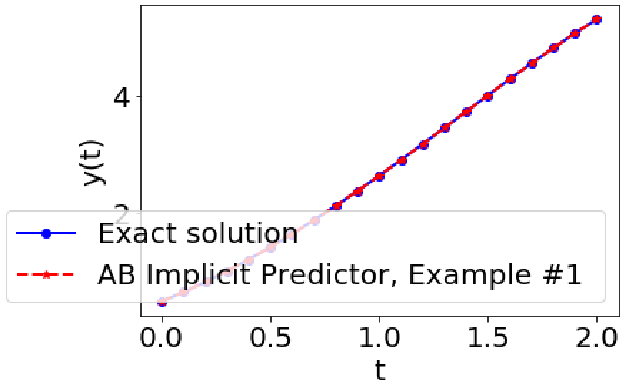




Predictor Corrector Method:

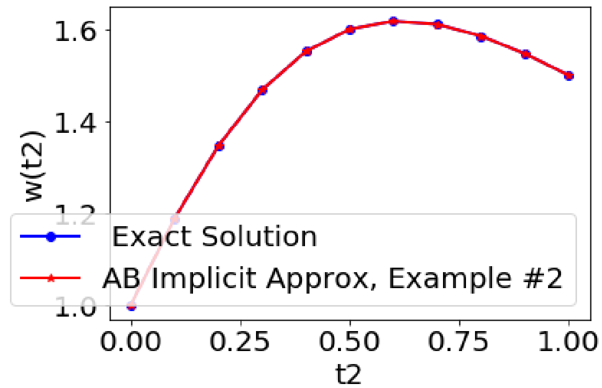
$IVP: y' = y - t^2 + 1$

$Exact\ sol: y(t) = (t+1)^2 - e^{\frac{t}{2}}$



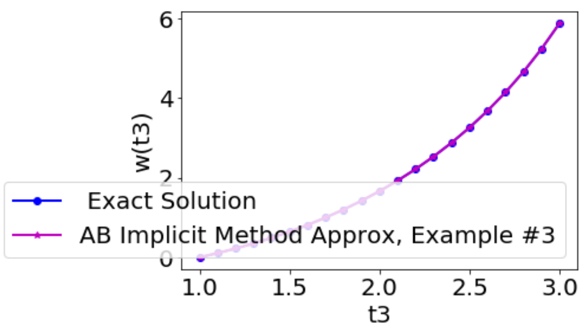
$IVP: y' = \frac{2-2ty}{t^2+1}$

$Exact\ sol: y(t) = \frac{2t+1}{t^2+1}$



$IVP: y' = 1 + \frac{y}{t} + (\frac{y}{t})^2$

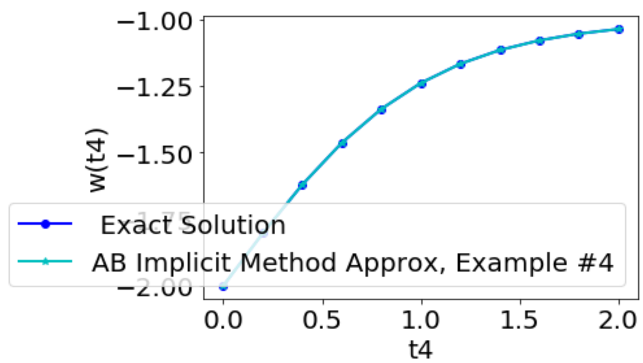
$Exact\ sol: y(t) = t(\tan(\ln(t)))$

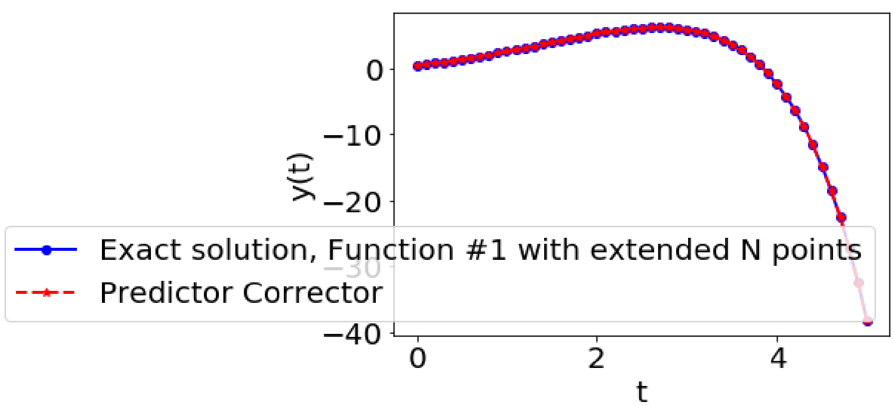


$IVP: y' = -(y+1)(y+3)$

$Exact\ sol: y(t) = -3 +$

$2(1 + e^{-2t})^{-1}$





Error

As with any approximation, there will always be error involved. The tables below are provided to give a more in depth understanding of the accuracy of each method.

Euler's Method:

Function #1:

|y_i-w_i|
0.00E+00
7.41E-03
1.53E-02
2.37E-02
3.25E-02
4.19E-02
5.19E-02
6.23E-02
7.34E-02
8.50E-02
9.71E-02
1.10E-01
1.23E-01
1.37E-01
1.51E-01
1.66E-01
1.80E-01
1.96E-01
2.11E-01
2.27E-01
2.42E-01

Function #2:

|y_i-w_i|
0.00E+00
1.19E-02
2.81E-02
4.58E-02
6.21E-02
7.50E-02
8.33E-02
8.72E-02
8.73E-02
8.45E-02
7.97E-02

Function #3:

|y_i-w_i|
0.00E+00
5.16E-03
1.13E-02
1.87E-02
2.73E-02
3.76E-02
4.97E-02
6.40E-02
8.09E-02
1.01E-01
1.24E-01
1.52E-01
1.85E-01
2.24E-01
2.70E-01
3.25E-01
3.90E-01
4.69E-01
5.63E-01
6.77E-01
8.14E-01

Function #4:

|y_i - w_i |
0.00E+00
2.62E-03
1.20E-02
2.42E-02
3.42E-02
3.92E-02
3.89E-02
3.49E-02
2.92E-02
2.32E-02
1.78E-02

Modified Euler's Method:

Function #1:

0.00E+00
4.15E-04
8.64E-04
1.35E-03
1.88E-03
2.45E-03
3.06E-03
3.73E-03
4.45E-03
5.22E-03
6.06E-03
6.97E-03
7.94E-03
8.99E-03
1.01E-02
1.13E-02
1.26E-02
1.40E-02
1.56E-02
1.72E-02
1.89E-02

Function #2:

0.00E+00
9.90E-04
1.80E-03
2.36E-03
2.66E-03
2.73E-03
2.63E-03
2.42E-03
2.14E-03
1.85E-03
1.57E-03

Function #3:

0.00E+00
2.01E-04
4.71E-04
8.12E-04
1.23E-03
1.73E-03
2.34E-03
3.06E-03
3.92E-03
4.95E-03
6.18E-03
7.64E-03
9.40E-03
1.15E-02
1.40E-02
1.71E-02
2.08E-02
2.54E-02
3.09E-02
3.78E-02
4.62E-02

Function #4:

0.00E+00
1.38E-03
2.87E-03
4.29E-03
5.36E-03
5.88E-03
5.87E-03
5.43E-03
4.75E-03
3.98E-03
3.22E-03

Midpoint Method:

Function #1:

0.00E+00
4.84E-03
1.01E-02
1.59E-02
2.19E-02
2.83E-02
3.49E-02
4.17E-02
4.86E-02
5.54E-02
6.21E-02
6.86E-02
7.48E-02
8.05E-02
8.55E-02
8.98E-02
9.31E-02
9.53E-02
9.61E-02
9.52E-02
9.25E-02

Function #2:

0.00E+00
5.88E-04
3.33E-03
7.86E-03
1.35E-02
1.95E-02
2.53E-02
3.03E-02
3.44E-02
3.75E-02
3.97E-02

Function #3:

0.00E+00
5.27E-03
1.15E-02
1.86E-02
2.67E-02
3.59E-02
4.63E-02
5.80E-02
7.11E-02
8.59E-02
1.02E-01
1.21E-01
1.42E-01
1.66E-01
1.93E-01
2.24E-01
2.59E-01
3.00E-01
3.46E-01
4.00E-01
4.62E-01

Function #4:

0.00E+00
2.62E-03
1.51E-02
3.90E-02
7.25E-02
1.12E-01
1.52E-01
1.91E-01
2.26E-01
2.55E-01
2.80E-01

Runge Kutta 4th Order Method:

Function #1:

0.00E+00
1.66E-07
3.45E-07
5.38E-07
7.45E-07
9.69E-07
1.21E-06
1.47E-06
1.75E-06
2.04E-06
2.36E-06
2.70E-06
3.07E-06
3.46E-06
3.88E-06
4.32E-06
4.79E-06
5.29E-06
5.83E-06
6.39E-06
6.99E-06

Function #2:

0.00E+00
4.72E-08
2.38E-07
5.68E-07
9.65E-07
1.34E-06
1.61E-06
1.77E-06
1.82E-06
1.79E-06
1.70E-06

Function #3:

0.00E+00
1.43E-07
2.16E-07
2.33E-07
2.05E-07
1.32E-07
1.50E-08
1.51E-07
3.73E-07
6.61E-07
1.03E-06
1.49E-06
2.07E-06
2.81E-06
3.72E-06
4.88E-06
6.33E-06
8.18E-06
1.05E-05
1.35E-05
1.75E-05

Function #4:

0.00E+00
2.71E-06
6.60E-06
1.24E-05
1.92E-05
2.49E-05
2.81E-05
2.86E-05
2.68E-05
2.36E-05
1.98E-05

Adams-Bashforth 4th Explicit Method:

Function #1:

0.00E+00
1.66E-07
3.45E-07
5.38E-07
1.52E-06
4.30E-06
7.43E-06
1.12E-05
1.57E-05
2.10E-05
2.73E-05
3.47E-05
4.33E-05
5.33E-05
6.49E-05
7.85E-05
9.41E-05
1.12E-04
1.33E-04
1.57E-04
1.85E-04

Function #2:

0.00E+00
4.72E-08
2.38E-07
5.68E-07
1.87E-05
3.97E-04
8.49E-04
1.22E-03
1.42E-03
1.45E-03
1.37E-03

Function #3:

0.00E+00
1.43E-07
2.16E-07
2.33E-07
2.17E-05
4.21E-05
5.87E-05
7.70E-05
9.88E-05
1.25E-04
1.57E-04
1.97E-04
2.46E-04
3.09E-04
3.88E-04
4.89E-04
6.18E-04
7.86E-04
1.00E-03
1.29E-03
1.68E-03

Function #4:

0.00E+00
2.71E-06
6.60E-06
1.24E-05
2.23E-04
4.68E-04
7.88E-04
1.25E-03
9.87E-04
1.02E-03
4.49E-04

Predicator Corrector Method:

Function #1:

0.00E+00
1.66E-07
3.45E-07
5.38E-07
6.90E-07
8.67E-07
1.07E-06
1.32E-06
1.60E-06
1.92E-06
2.29E-06
2.73E-06
3.22E-06
3.80E-06
4.45E-06
5.21E-06
6.07E-06
7.06E-06
8.18E-06
9.47E-06
1.09E-05

Function #2:

0.00E+00
4.72E-08
2.38E-07
5.68E-07
2.57E-05
7.44E-05
1.23E-04
1.56E-04
1.68E-04
1.64E-04
1.50E-04

Function #3:

0.00E+00
1.43E-07
2.16E-07
2.33E-07
4.27E-07
6.91E-07
1.03E-06
1.46E-06
2.01E-06
2.70E-06
3.59E-06
4.71E-06
6.14E-06
7.98E-06
1.03E-05
1.34E-05
1.73E-05
2.23E-05
2.90E-05
3.77E-05
4.94E-05

Function #4:

0.00E+00
2.71E-06
6.60E-06
1.24E-05
2.28E-05
1.42E-04
2.53E-04
2.88E-04
2.56E-04
1.94E-04
1.30E-04

Euler’s Method Analysis

Euler’s method is the most elementary approximation technique for solving IVP. This method only involves one step in finding each mesh point. Due to its simplicity of Euler’s method has a higher error than other methods. We can depict this error as $O(h)$.

- As we can see above, Euler’s method starting accuracy starts at 10^{-3} and it begins to immediately decease as t increases and we can see a clear depiction of this error by looking at each graph. The approximation of $y(t)$ using becomes Euler’s method becomes become less and less accurate as the interval $[a, b]$ increases. If we look at the extended graph, the error becomes almost 400% when the interval become [0,5] while keeping the step size the same. Euler’s method has a weak stability. When approximating the IVP with Euler’s method, the error bound converges to zero as the step size decreases but as we can see, when the step size is kept the same, the error becomes greater and greater. Therefore in practice, Euler’s method is not used.

Modified Euler’s Method Analysis

Modified Euler’s Method is a one-step ODE IVP approximation technique. It is a clear and precise one-step method that is fairly straightforward to understand, but usually fails to give satisfactory results in science or engineering. It is derived from the Midpoint Method approximation technique and because of this, the Modified Euler’s Method has an error of $O(h^2)$.

- Accuracy: Modified Euler’s method has a less than desirable accuracy when approximating. On the examples tested, the accuracy of the modified Euler’s method usually operated in an error range of $[10^{-2}\ to\ 10^{-4}]$, but on average the error bound value was 10^{-3}. While this type of accuracy is not necessarily bad, this accuracy in application, especially industry is not desirable. The accuracy of this method becomes even more apparent when testing Modified Euler’s Method over an extended range t. From our data collected, over time, Modified Euler’s method becomes extremely unstable, even to the point where they error reaches a 100 percent bound. Highly inaccurate to say the least.

- Stability: Modified Euler’s method has weak stability. When approximating with Modified Euler’s method, error bounds for the method are convergent towards zero. For the most part all error values of modified Euler’s method, from our testing, these error bounds have a limit of zero. Because of this Modified Euler’s method is consistent with weak accuracy and convergent, but also this method has weak stability over time. When this method is tested with a larger t range, the method shows less desirable stability. With a large range t, the error value gets worse and worse quickly, and this attribute in practice, is not desired.

Midpoint Method Analysis

The midpoint method tries for an improved prediction. Midpoint method is a fairly simple one-step method. It does this by taking an initial half step in time, sampling the derivative there, and then using that forward information as the slope. In other words, it replaces the tangent line by a line that is starting to bend correctly. For the midpoint method, our estimated solution values accuracy does not change drastically as we decrease the step size, as the Midpoint Method has truncation error of $O(h^2)$.

- Accuracy: Basis of the Midpoint Method is just a half step-sized Euler's method, because of this the accuracy of the Midpoint Method is not desirable. Approximated values for this method could be tested in maybe academia examples but the method would not be suitable for application. There are many reasons for this, the main reasons are; accuracy is not desirable at this stage, and over time the accuracy decreases very easily. This method tested holds at small ranges of t, but when put to higher t range values the error is on par with Euler's method; highly undesirable.

- Stability-Midpoint method has poor stability, because of its simple design approximating with this method will yield decent accuracy results at small t ranges. When tested with extreme conditions, or larger ranges of t, the midpoint method fails to remain stable, the consistency of the method decreases and error spikes of control. In application the midpoint method would not be suitable.

Runge-Kutta 4th Order Method Analysis

Runge-Kutta 4th order is the most commonly used Runge-Kutta methods because of its accuracy and stability. This method requires 4 evaluations per step whereas Euler's and Midpoint method only requires 1 and 2 respectively. The local truncation error is $O(h^4)$.

- As we can see above, Runge-Kutta 4th order is extremely accurate, starting at an error of$10^{-8}$and only decreasing to 10^{-5} as t increases and we can see a clear depiction of the accuracy of this method in the graphs. Also, as we increased the interval of $[a, b]$ while keeping the step size the same, shown in the extended graph, Runge-Kutta holds its accuracy and consistency throughout the approximation of $y(t)$. Runge-Kutta has a strong stability. When approximating the IVP using this method, the error bound converges to zero as the step size decreases, showing that it's convergent.

Adams-Bashforth 4th Step Method

The Adams-Bashforth based off of Adams-Bashforth 4-Step explicit method, and Adams-Moulton 3-step explicit method. This multi-step method is fairly complex, and expensive to compute. As its base it is derived from the Runge Kutta 4th order method and uses the RK4 method to compute the first three values. After the first three values are found, then the AB 4th step explicit method is used to compile into the implicit three step method. This method has an error of $O(h^4)$. This particular

- Accuracy: Adams Bashforth 4th step method is extremely accurate. From our findings it maintains a suitable error rate that is desirable. Whether in theory the accuracy of the method works very well. This also holds over longer approximation ranges and bigger mesh point values. Because of the error of $O(h^4)$ the methods get substantially larger has h decreases depending on the number of iterations wanted. The only downside of this method is its complexities. This method can be harder to program, and it is expensive to compute, and over time with a large approximation range, the computing power becomes very intense.
- Stability:
  - Stability of multistep method Definition of stability of multistep method.
    - 1) Methods that satisfy the root condition and have as the only root of the characteristic equation with magnitude one is called strongly stable.
    - 2) Methods that satisfy the root condition and have more than one distinct roots with magnitude one is called weakly stable.
    - 3) Methods that do not satisfy the root condition are called unstable.

- Because this predictor corrector method is largely based from the AB 4-step explicit method we can derive the characteristic polynomial:
    - $P(\lambda) = \lambda^4 - \lambda^3 = 0 \rightarrow P(\lambda) = \lambda^3(\lambda - 1)$
    - $P(\lambda)\ has\ roots\ \lambda_1 = 1, \lambda_{2,3,4} = 0$
    - $P(\lambda)$ satisfies root condition and therefore the method is strongly stable.
- Based on our project this stability holds true, over various t ranges and regardless of the complexity of the function, the AB 4th method still held desirable accuracy ranges, and it is not as expensive as the predictor method in computing, but the computational cost of the method is higher than the RK4 method.

Predictor Corrector Method Analysis

In our project, we used the Predictor Corrector Method based off of Adams-Bashforth 4-Step explicit method, and Adams-Moulton 3-step explicit method. This multi-step method is fairly complex, and expensive to compute. As its base it is derived from the Runge Kutta 4th order method and uses the RK4 method to compute the first three values. After the first three values are found, then the AB 4th step explicit method is used to compile into the implicit three step method. This method has an error of $O(h^m), but\ particularly\ since\ this\ revolves\ around\ AB\ 4th\ step, the\ truncation\ error\ is\ O(h^4)$.

- Accuracy: The predictor corrector method is extremely accurate. From our findings it maintains a suitable error rate that is desirable. Whether in theory or application the accuracy of the method works very well. This also holds over longer approximation ranges and bigger mesh point values. Because of the error of $O(h^m)$ the methods get substantially larger has h decreases depending on the number of iterations wanted. The only downside of this method is its complexities. This method can be harder to program, and it is expensive to compute, and over time with a large approximation range, the computing power becomes very intense.

- Stability- The Predictor Corrector Method is extremely stable. The predictor corrector method accurately approximates the IVP to suitable accuracies whether the range is large or small. This method holds its consistency in truncation error even when tested on larger mesh points and a higher t range. The predictor corrector method still proves to be a suitable approximation method. The biggest downside to this method is the high computational cost, which in practice may be hard to apply because every resource is be optimized in practice.

Conclusion

In conclusion the various methods we have discussed provide great means as basis of approximating IVP and first ODE. The methods are fairly easy to understand and program. However, the true test of whether or not these methods are applicable is analyzation of their stability, and accuracy. With analyzation of our implementations, we can conclude that multistep methods are extremely more accurate than the one-step methods. The accuracy of the multi-step methods are highly desirable in academia, theory, and industry application. The cost however to using the multi-step methods, is the computational cost. Because of the higher complexity of the multi-step methods, the computational cost is much higher and in application when a user wants to approximate IVP over a large range, the high computational cost could create problems. In conclusion, we have found that the Runge-Kutta 4^{th} order method provides a great balance to computational cost versus accuracy. Hence why the RK4 is the usual go to method for approximation in industry.