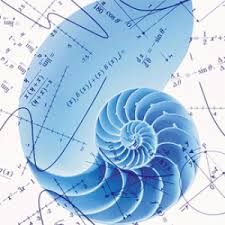
## 

Numerical Analysis - Project 2

Sam Meyerson

UF ID: 2157-5991

04/29/2020

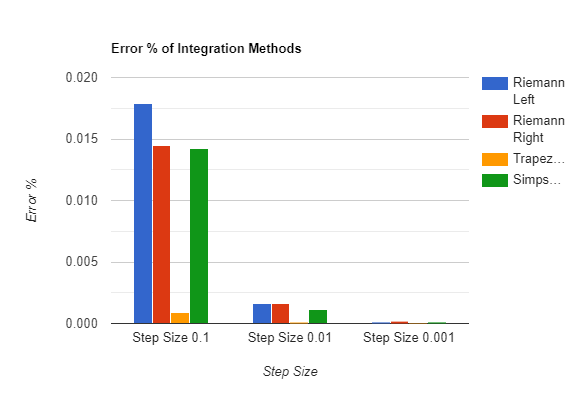


## Intro

In this project, I will find what methods work best for numerical integration, numerical differentiation, and numerical ODE solving. I will identify which method performs the best by comparing their errors at various step sizes. Code and results in excel for this project are on my github page:

## Numerical Integration Methods

For this function, the trapezoid method performed the best(i.e. had the smallest error) at every step size. The error size for all methods decreases as the step size also decreases, as expected.



## Numerical Differentiation Methods

The error for symmetric differentiation is much lower than the errors for the backward and forward methods. The symmetric method is not as prone to having higher error when the h-value is changed as the other two methods. The higher the h-value, the higher the error is in general.

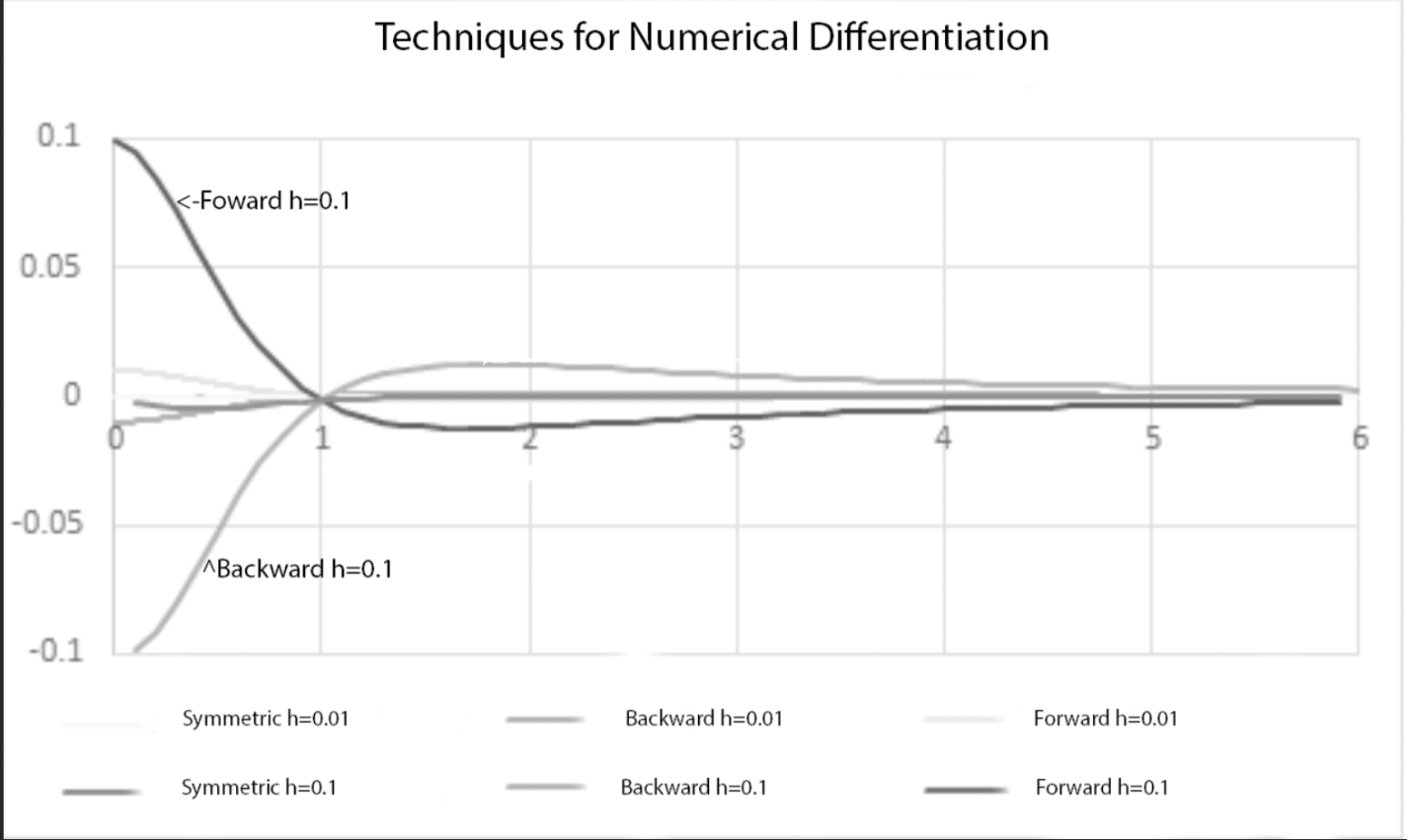
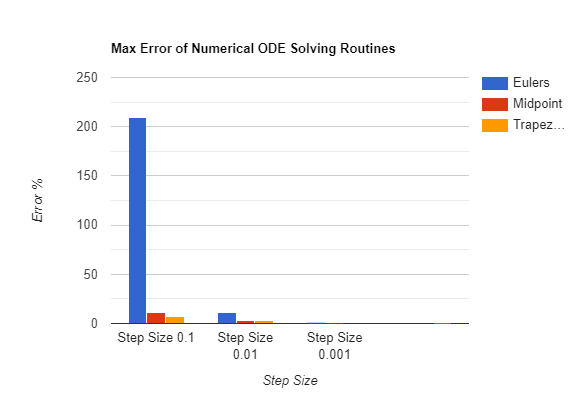


Figure - The graph lost its color for some reason when I put exported it so I added labels for forward and backward errors of h=0.1

## Numerical ODE Solving Routines

### y’=3y, y(0)=1, x=[0,3]

The trapezoid method will work best for equations like this where you are able to get the implicit form of the equation. The midpoint method was a close second and Euler’s method performed much worse at each step size.

### y’=1 / (1 + x^2) - 2y^2, y(0)=0, x=[0,10]

For this function, the midpoint method performed the best at each step size. The iterative function uses an approximation of itself in order to get an overall approximation, which is why the midpoint method performed best here.