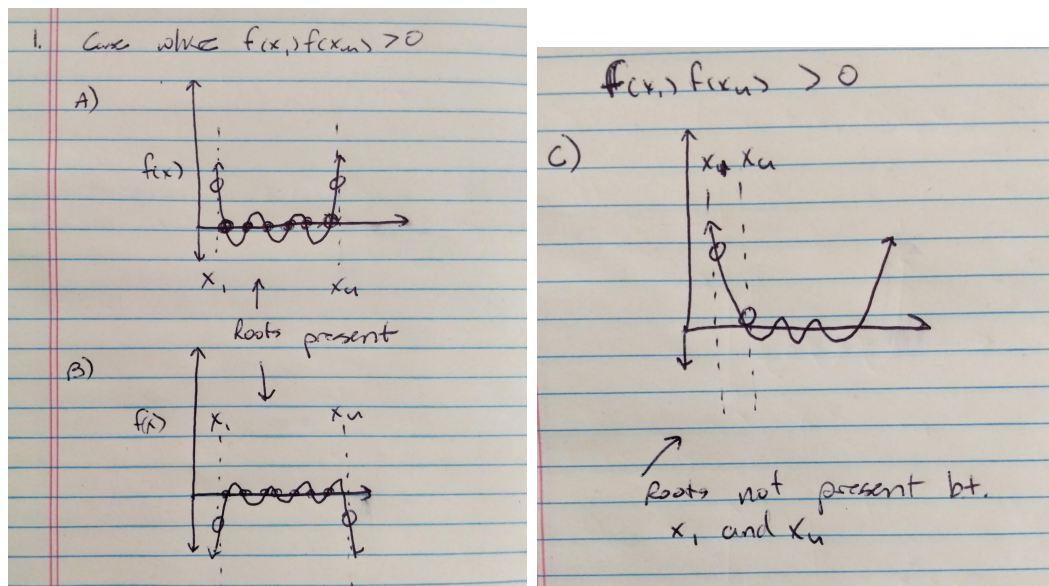


1 Solving a Nonlinear Single Equation

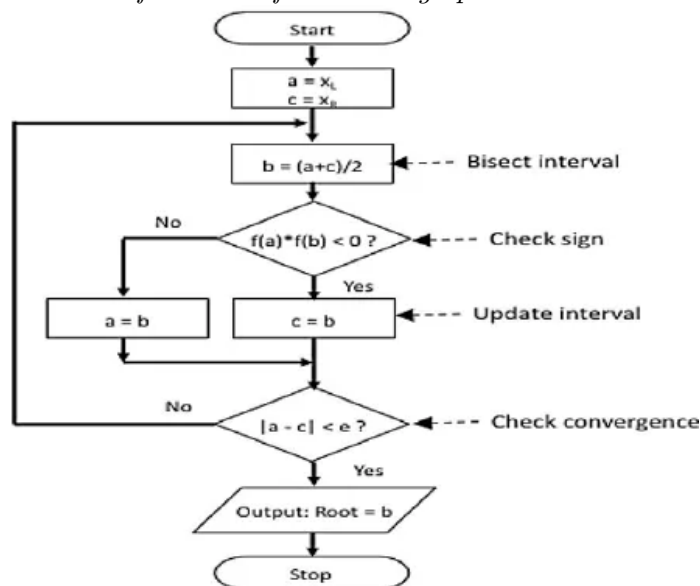
1.2 Bisection Method Thought Exercise | Task List 1-A

1. A, B, and C demonstrate cases where there may or may not be a root between $f(x_1)$ and $f(x_m)$ when their product is greater than zero. ****Note:** when I drew these, I accidentally put x_u instead of x_m .
2. A and B also demonstrate that there may be multiple roots between x_1 and x_u when their product is greater than zero.



3. Bisection Method Flowchart (see code for implementation):

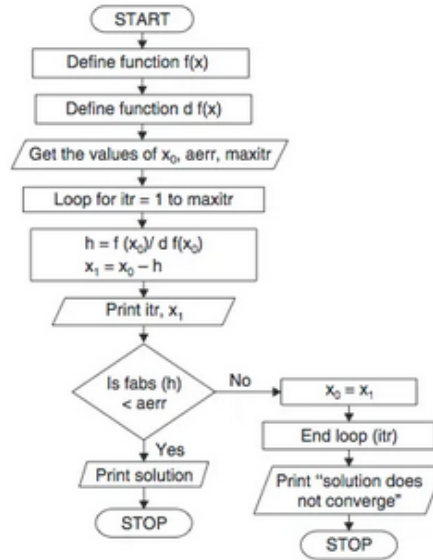
flowchart from codingalpha.com



1.3 Newton's Method | Task List 1-B

- Below is a flowchart demonstrating Newton's Method:

flowchart from codingalpha.com



- Here is a table showing iteration k , x_k , and ϵ for the first root-finding problem. ϵ was set to 10^{-6} .

k	x_k	ϵ
0	1	Inf
1	0.687116367041199	0.455357560912301
2	0.479569647933555	0.432777011643571
3	0.342752246343126	0.399172880849518
4	0.253830851560911	0.350317521433663
5	0.19792929482791	0.282431950164854
6	0.165573528412162	0.195416300697569
7	0.150538921218223	0.099871894074121
8	0.146628975172691	0.026665575756273
9	0.146360742307274	0.001832683144324
10	0.146359504721027	8.46E-06

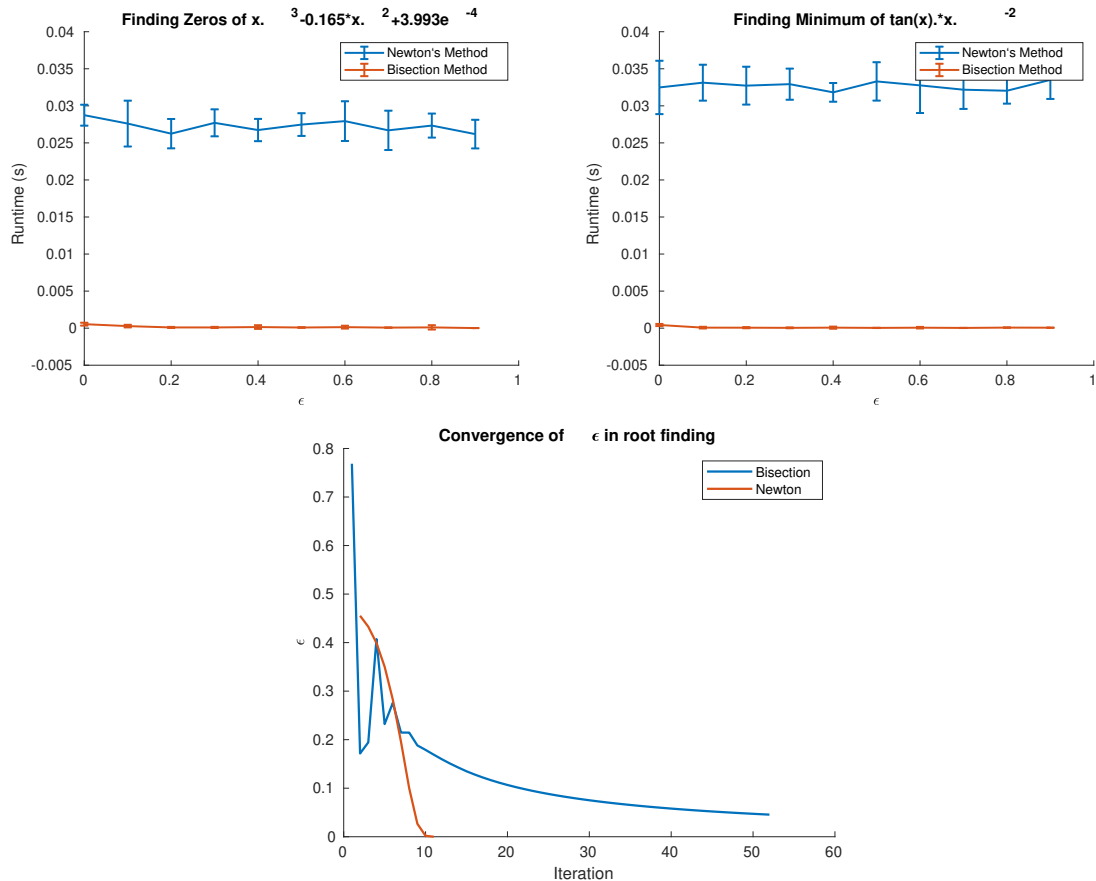
- Here is a table for the minimization problem, also showing iteration k , x_k , and ϵ . ϵ was set to 10^{-6} .

k	x_k	ϵ
0	1	Inf
1	0.950777558639313	0.051770722724179
2	0.947754440621915	0.003189769298696
3	0.947747133558387	7.71E-06

1.4 Comparison of Algorithms | Task List 1-C

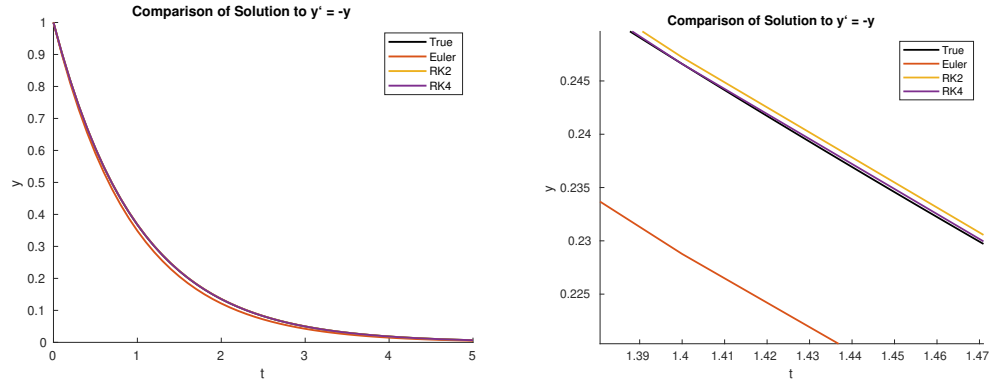
Wall-clock runtime analysis demonstrates, in the cases provided, that the bisection method performs faster than the Newton Method in the top two figures. HOWEVER, this absolutely does not mean Bisection Method is better than Newton's Method. In fact, the bisection requires more iterations to converge to a solution within error ϵ as seen in the bottom figure. The bisection method converges linearly while Newton's Method converges quadratically.

The limiting factor in this case is likely the fact that Newton's Method requires the evaluation of a derivative prior to iterating, which computationally takes a good bit of time.

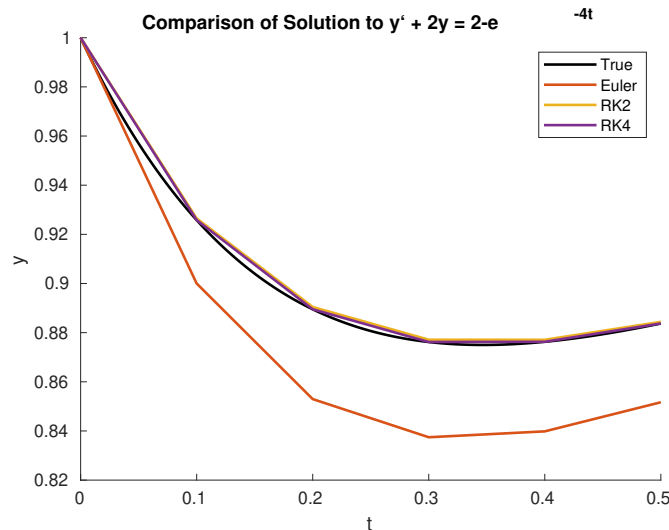


2 Runge-Kutta Methods

1. Euler's Method (RK1), RK2, and RK4 estimations of the solution to the ODE $y' = -y$ were compared to the analytical solution (black). Notably, the RK4 estimation was closest followed by RK2 and Euler's Method. See code for implementations.



2. Similar findings were observed in the approximations for the ODE $y' + 2y = 2 - e^{-4t}$.



3 Nelder-Mead Optimization Algorithm

- The Nelder-Mead algorithm can primarily be used when trying to determine the minimums presented in an n -dimensional function. This might be particularly useful in mapping the lowest point of a 3D map or in *non-linear optimization problems* such as Rosenbrock's Function. Nelder-Mead can also be used when the derivative of the function is not known, for example in a black-box type problem where a system must be perturbed in order to quantify a response.
- Using the `fminsearch` function, I was able to ascertain that the minimum of the Rosenbrock function is $[1, 1]$.