Homework 4

ME 5010

Mathematical Methods for Engineers

Submitted by:
Pushkar Mishra
ME22RESCH11007

github_repository: https://github.com/mepushkar/Math_assngmt

Question 1:

a) The exact roots of the given polynomial are $\pm\sqrt{3}$ and -1. Thus with the Initial guess $P_0=2$ and tolerance $\epsilon=10^{-10}$ we converge to $+\sqrt{3}$ at 4^{th} iteration.
The error_ratio_2 is nearly a constant so we may say that Newton raphson has second order convergenve.

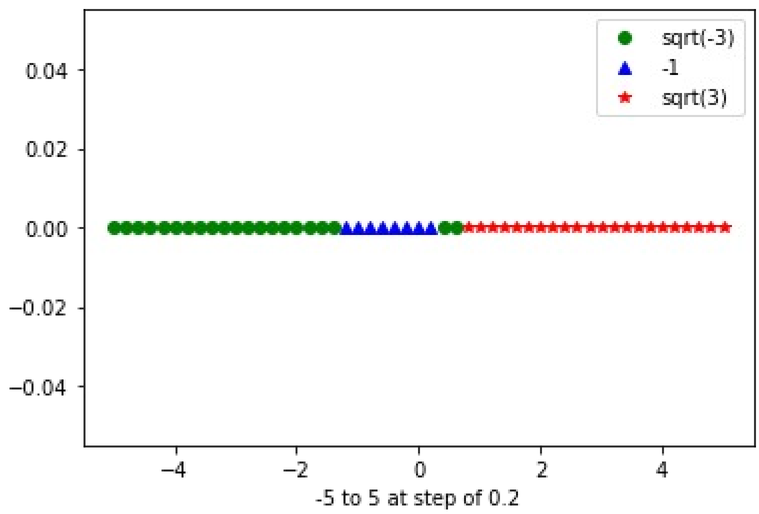
iteration	absolute_error	error_ratio_1	error_ratio_2	error_ratio_3
1	0.037179962	--	--	--
2	0.000873003	0.023480466	0.63153551	16.985910748
3	0.000000499	0.000571064	0.654137117	749.295529697
4	0	0.000000326	0.654853289	1313541.86698098

Table for Q 1a)

b) As there are three roots and the problem we are solving in this question is Newton Raphson root finding problem with P_0 as initial guess, we would have four (4) possible outcomes, namely
(1) The guess P_0 leads us to $-\sqrt{3}$
(2) or to -1
(3) or to $\sqrt{3}$
(4) if none of the above three destinations are reached the initial guess would diverge instead of converging to a particular root.
The figure shown below shows the root which is reached by starting the initial guess from -5 to 5 with an step-size of 0.2.
In the graph shown below, the destination to which a particular initial guess P_0 would lead is indicated by means of different coloured markers.

In the given range [-5,5] all the initial guesses lead to either of the roots, therefore there is no such initial guess P_0 for which Newton Raphson does not converge to a particular root.

As in the table above we can see that the value of error ratio 2 nearly remains constant, hence we may say that Newton Raphson has second order of convergence.



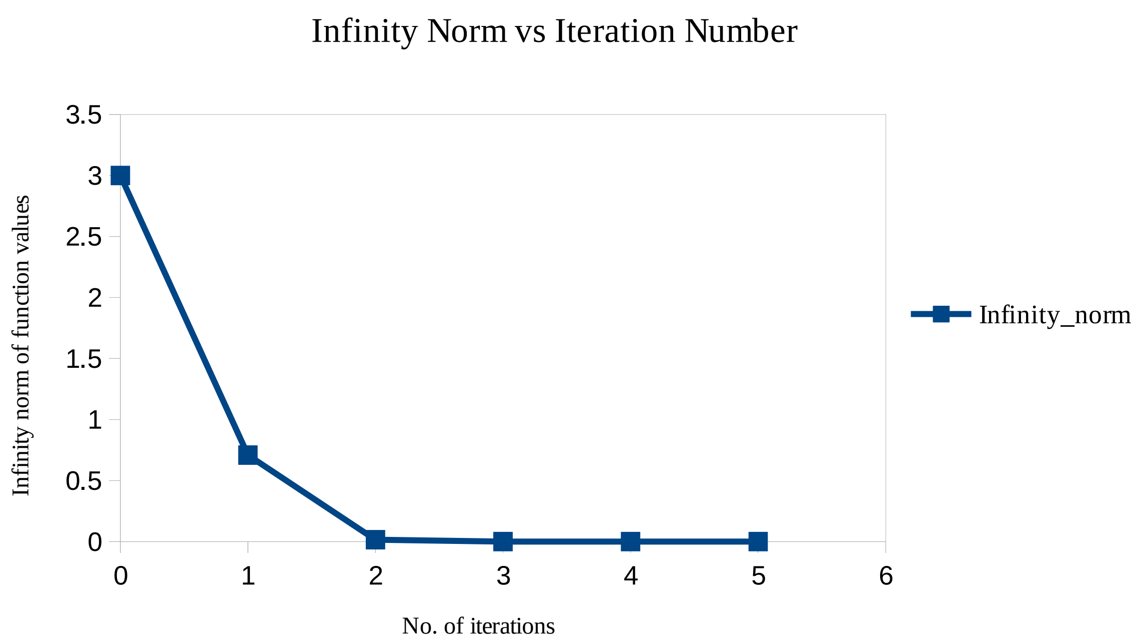
Question 2:

The Table shown below shows the data which was asked in the assignement the table includes the values rounded off to 5 decimal places due to ‘aesthetic’ reasons.
If anyhow more accuracy is desired code shared can be referred.
Non linear Newton-Raphson method is employed for solving the given set of equations as there are 3 unknowns and 3 equations and in this case we are getting 3 distinct roots. Hence the given set of equations has unique solution.

Iteration_No.	Infinity_norm_f	[f1 f2 f3]	vector_x [x1,x2,x3]
0	3	[-3. 3. 0.]	[1.0, 1.0, 1.0]
1	0.71	[0.629738 -0.28863 -0.708455]	[1.42857, 0.14286, 1.42857]
2	0.02	[0.000572 -0.000592 -0.015152]	[1.44011, 0.49305, 1.41331]
3	1.987236353251E-05	[2.0e-05 1.6e-05 1.8e-05]	[1.44226, 0.50001, 1.41421]
4	0	[0. 0. 0.]	[1.44225, 0.5, 1.41421]
5	1.7763568394002E-15	[-0. 0. -0.]	[1.44225, 0.5, 1.41421]

Table showing Infinity norm of vector f, values of functions f and x after each iteration

The graph below shows us a variation between number of iterations and infinity norm of function values (**[f1,f2,f3]**)

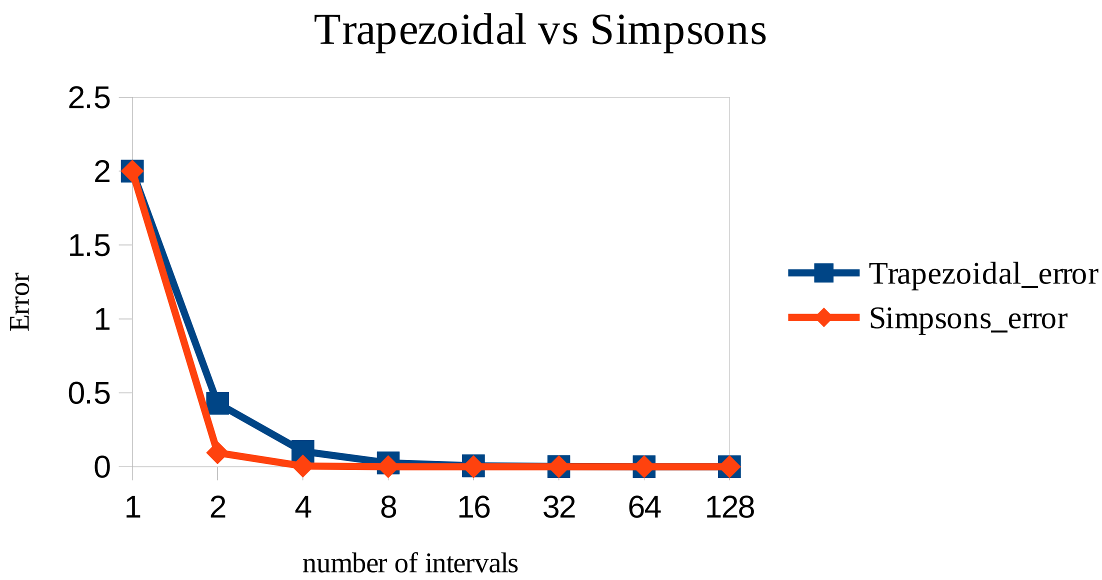


Question 3: all answers are in the writeup below
As it can be interpreted from the table below that:

- Error in simpsons in less than the trapezoidal rule.
- Error ratio in trapezoidal becomes nearly 4 which is 2^2 when step size is doubled so error reduces with order 2.
- Whereas in simpsons rule the error reduces at 4^{th} order as ration attains a constant value of 16 i.e. 2^4.
- I_{exact}=2.
- The graph below shows the reduction in error with increase in number of intervals **n.**
- **** for aesthetic purpose The value upto 6^{th} place of decimal are shown for more details refer code.**
- Incase the exact solution is not known then we may look for the error ratio untill which the ratio becomes nearly constant.

n_value	Trapezoidal	Trapezoidal_error	error_ratio_t	Simpsons	Simpsons_error	error_ratio_s
1	0	2	--	0	2	--
2	1.570796	0.429204	4.659792	2.094395	0.094395	21.18754
4	1.896119	0.103881	4.131682	2.00456	0.00456	20.701793
8	1.974232	0.025768	4.031337	2.000269	0.000269	16.94006
16	1.99357	0.00643	4.007741	2.000017	0.000017	16.223806
32	1.998393	0.001607	4.00193	2.000001	0.000001	16.055292
64	1.999598	0.000402	4.000482	2	0	16.013783
128	1.9999	0.0001	4.00012	2	0	16.00344

Table for Q3



Question 4:

S No.	Step_h	y_num6	Error	e_2h/e_h
1	1	16.1741666666667	0.576390148701666	3.09283855637344
2	0.5	16.5641939845268	0.186362830841556	3.49840026861121
3	0.25	16.6972859439607	0.0532708714076158	3.74249386546785
4	0.125	16.7363227582603	0.0142340571080553	3.87100722780494
5	0.0625	16.746879721457	0.00367709391132465	3.93567474810188
6	0.03125	16.7496225171484	0.000934298219917196	3.96791114674847
7	0.015625	16.750321351875	0.000235463493350352	3.98397795641787
8	0.0078125	16.7504977127588	5.91026094838298E-05	3.99199506575477
9	0.00390625	16.7505420100871	1.48052811965726E-05	3.99599908389922
10	0.001953125	16.7505531103422	3.70502617386137E-06	3.99800008018449
11	0.0009765625	16.7505558886484	9.26719884830618E-07	--

Table showing the relation between error ratio for $y_{num}(6)$ and step-size h

a) The handwritten solution of the first part is attached at the end of this pdf
b) The final value $y_{num}(6)$ for h=0.5 can be seen from the table above.
c) The code submitted in the zip file gives the table above and from the error ratio we can observe that it nearly approaches a constant ratio 2^2 upon reducing the step size to half, the error reduces 4 times hence we can say that with the help of error ratio that error reduces at second order.

Handwritten part

$$y_{01} = y_0 + \left(\frac{1}{2}k_1 + \frac{k_2}{2}\right)h.$$

$$k_1 = f(x_0, y_0)$$

$$k_2 = f(x_0 + h,\ y_0 + k_1 h)$$

$$k_1 = 2$$

$$k_2 = f(1.5,\ 2)$$

$$= 1 + \frac{2}{1.5} = 1 + \frac{4}{3} = \frac{7}{3}$$

$$y_1 = 1 + \frac{h}{2}(k_1 + k_2)$$

$$y_1 = 1 + \frac{0.5}{2}\left(\frac{7}{3} + 2\right).$$

$$y_1 = 2.083333333$$

for 2nd

$$k_1 = f(1.5,\ 2.083333333)$$

$$= 2.388888889$$

$$k_2 = f(2,\ 2.3888 \times h)$$

$$= 1.59722222$$

$$y(2) = y_1 + \frac{h}{2}(k_1 + k_2)$$

$$= 3.34027$$