MA 322: Scientific Computing
LAB 8 Report
Abheek Ghosh - 140123047

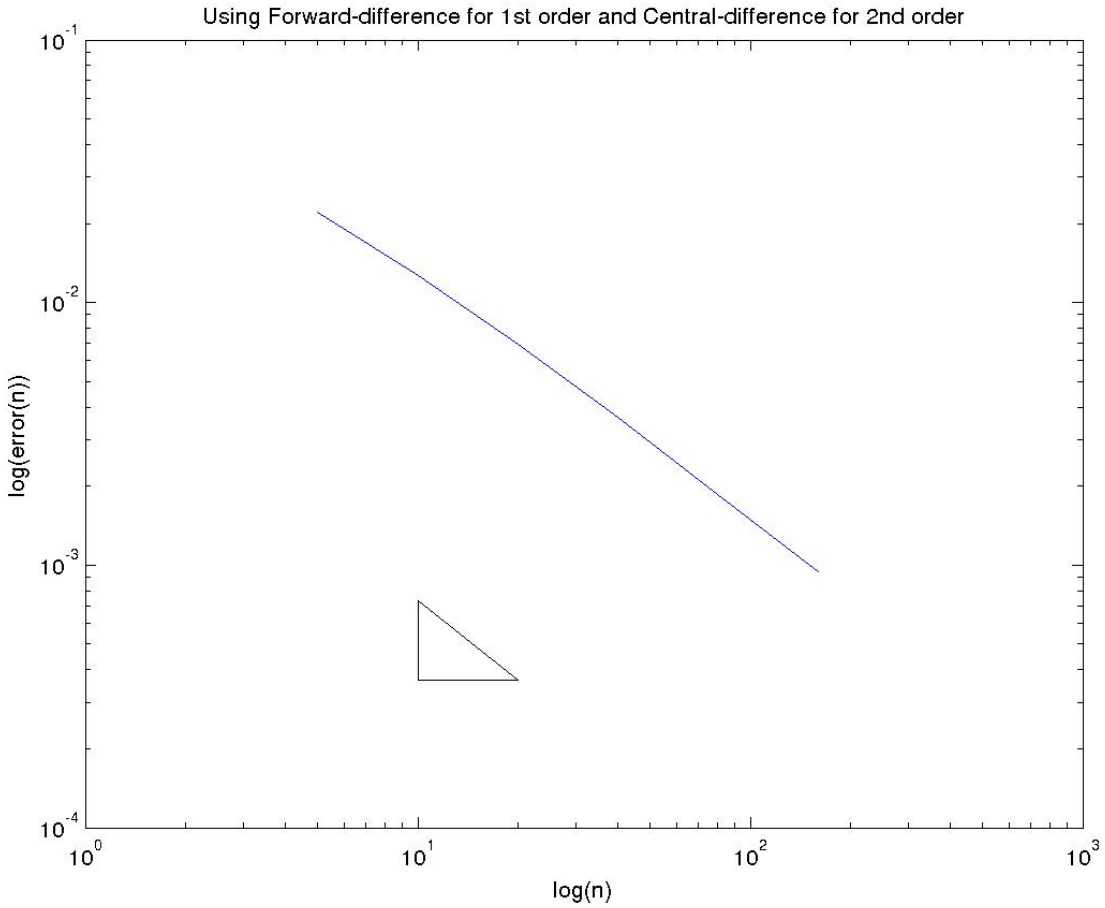
NOTE: The right angled triangle is for comparison of the slope. It has been made with the hypotaneuse slope as expected for respective methods.

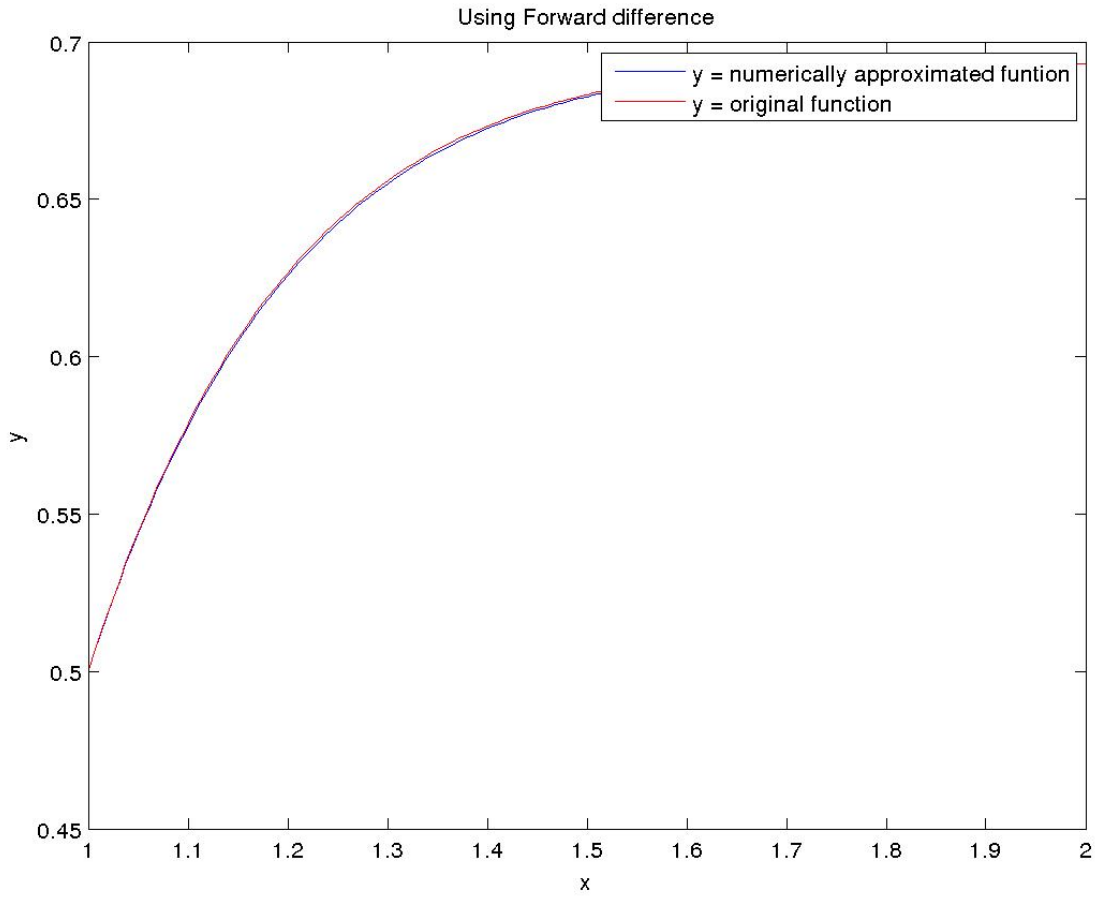
Question 1

(A) Dirichlet Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

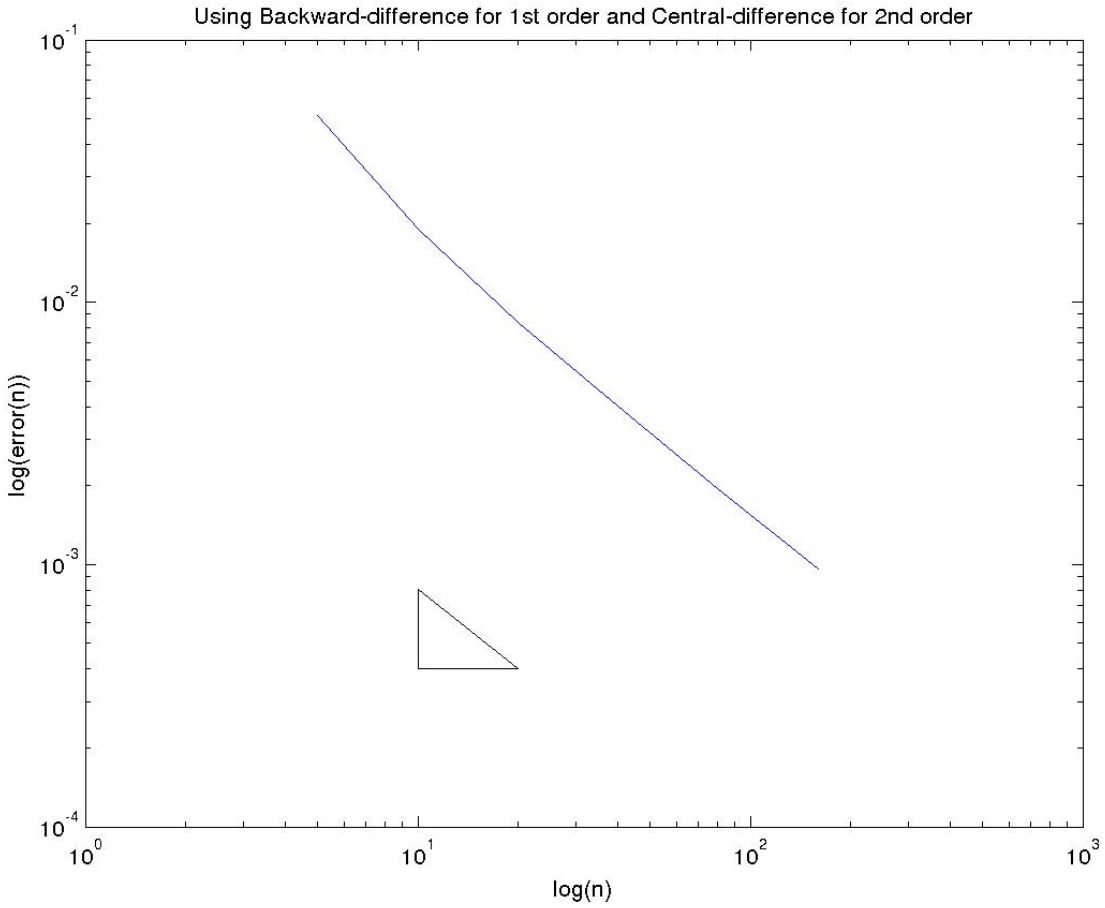
N	Error	Order
5	2.197919e-02	0.792223
10	1.269196e-02	0.864787
20	6.969505e-03	0.935965
40	3.642909e-03	0.967211
80	1.863326e-03	0.982910

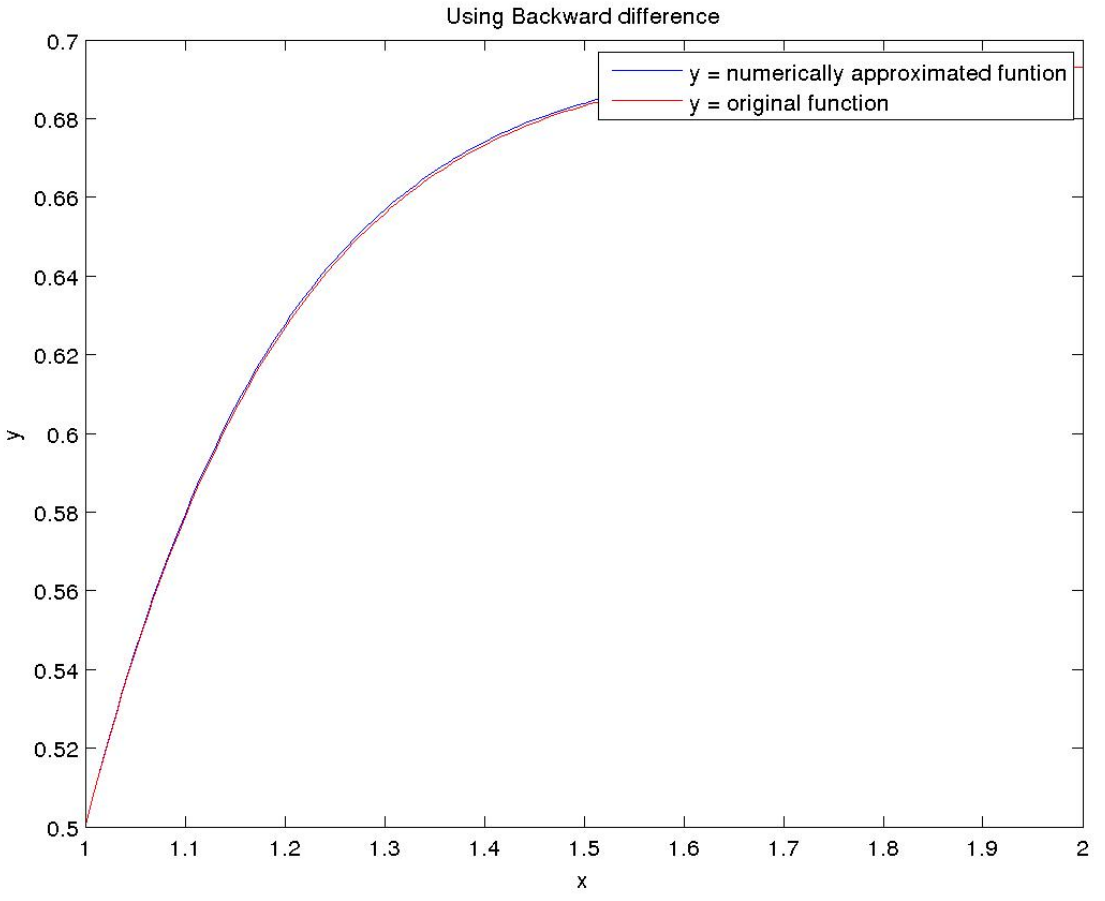




Using Backward-difference for 1st order and Central-difference for 2nd order

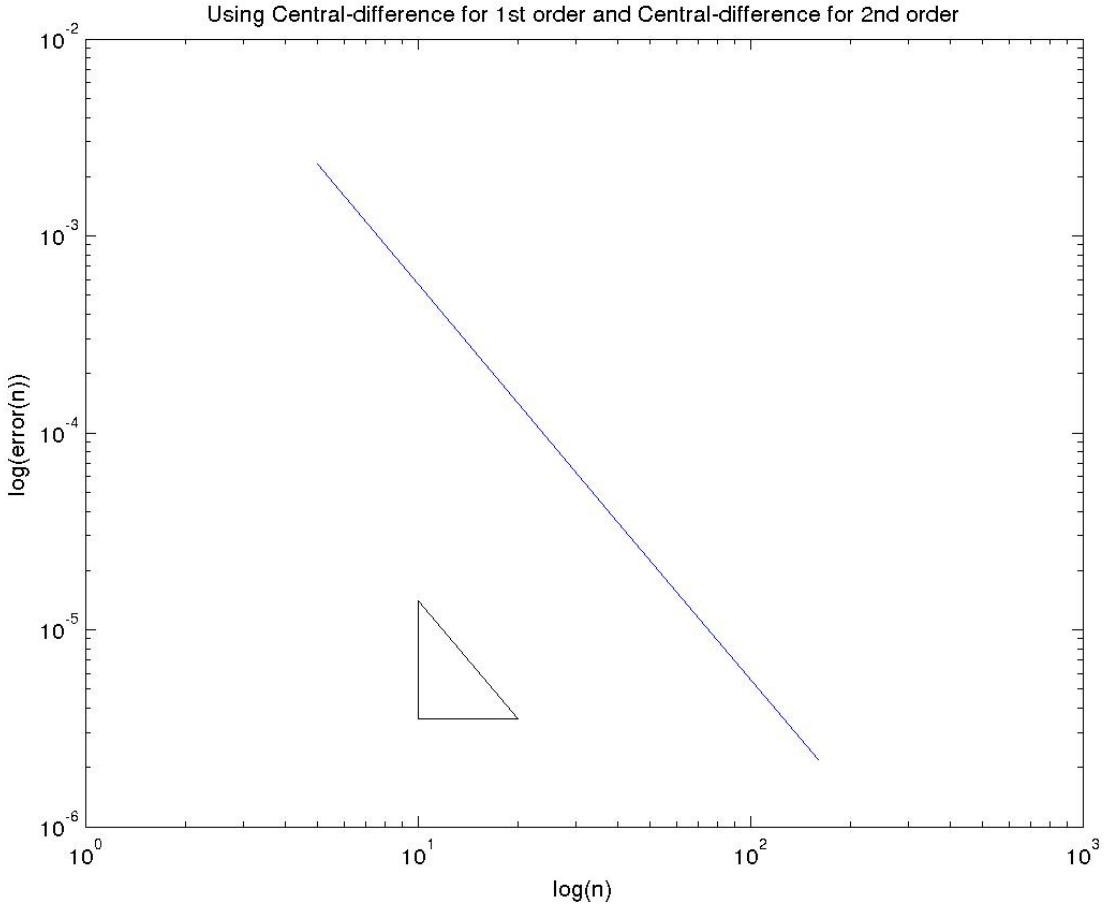
N	Error	Order
5	5.166753e-02	1.448127
10	1.893599e-02	1.172924
20	8.398521e-03	1.067958
40	4.006042e-03	1.035812
80	1.953913e-03	1.017441

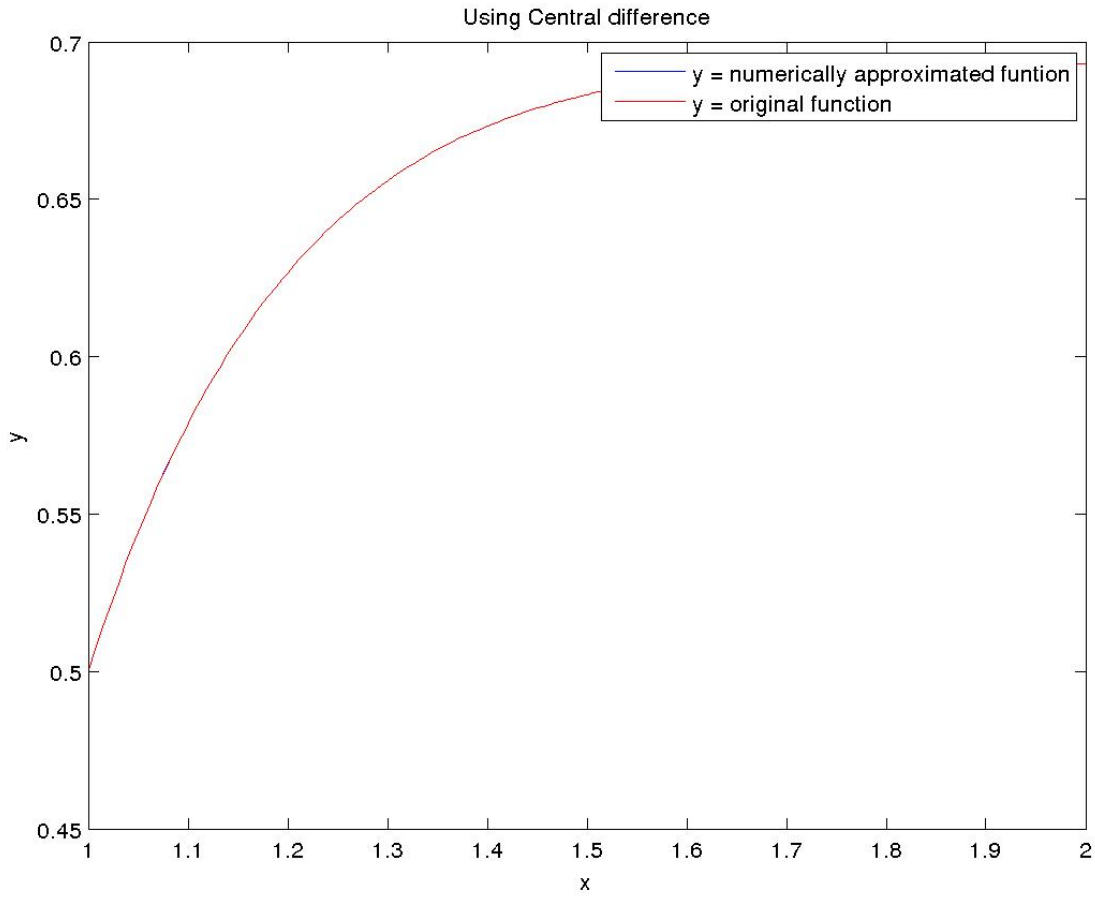

Using Backward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^-1
10^-2
10^-3
10^-4
10^0
10^1
10^2
10^3


Using Backward difference
y = numerically approximated funtion
y = original function
y
x
0.7
0.68
0.66
0.64
0.62
0.6
0.58
0.56
0.54
0.52
0.5
1
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2

Using Central-difference for 1st order and Central-difference for 2nd order

N	Error	Order
5	2.311583e-03	2.031265
10	5.655066e-04	2.007719
20	1.406222e-04	2.001924
40	3.510871e-05	1.998787
80	8.784558e-06	2.000152

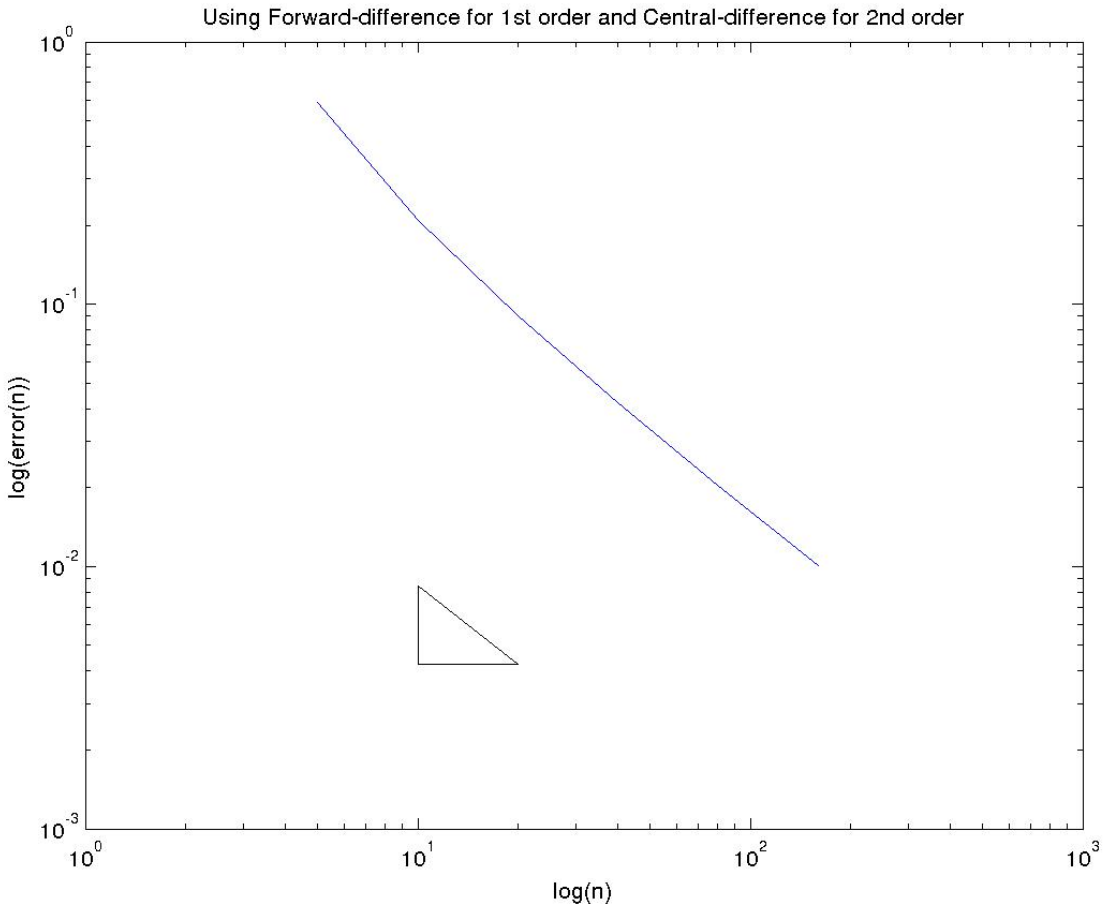


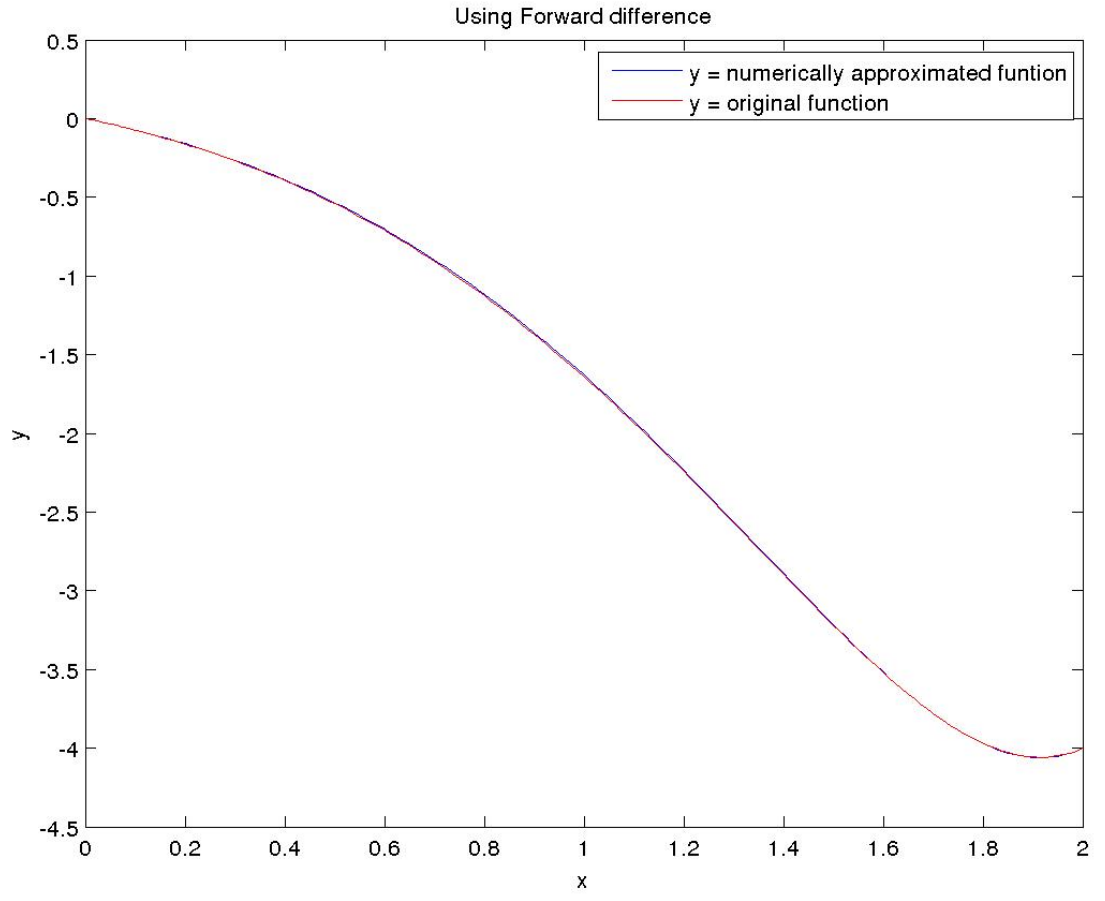


(B) Dirichlet Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

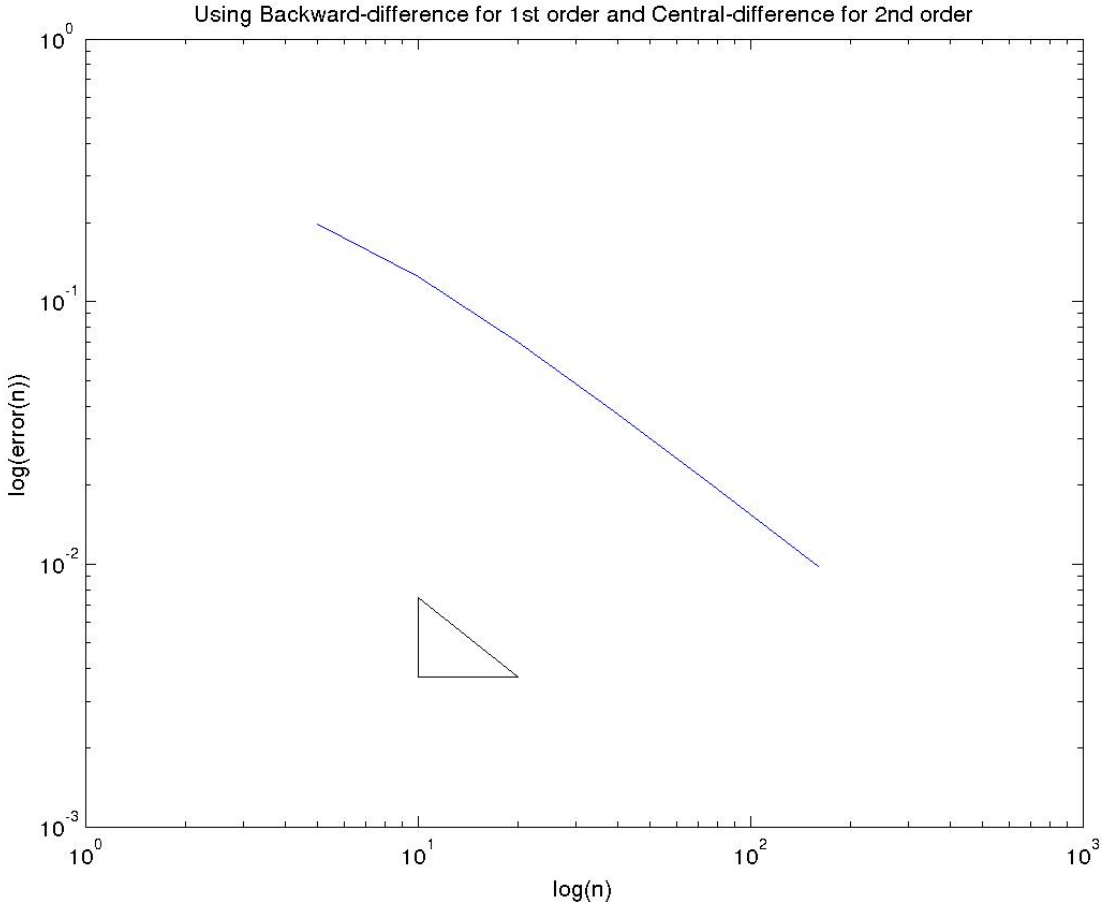
N	Error	Order
5	5.851788e-01	1.481215
10	2.096035e-01	1.210517
20	9.057256e-02	1.098021
40	4.231160e-02	1.047158
80	2.047544e-02	1.022947

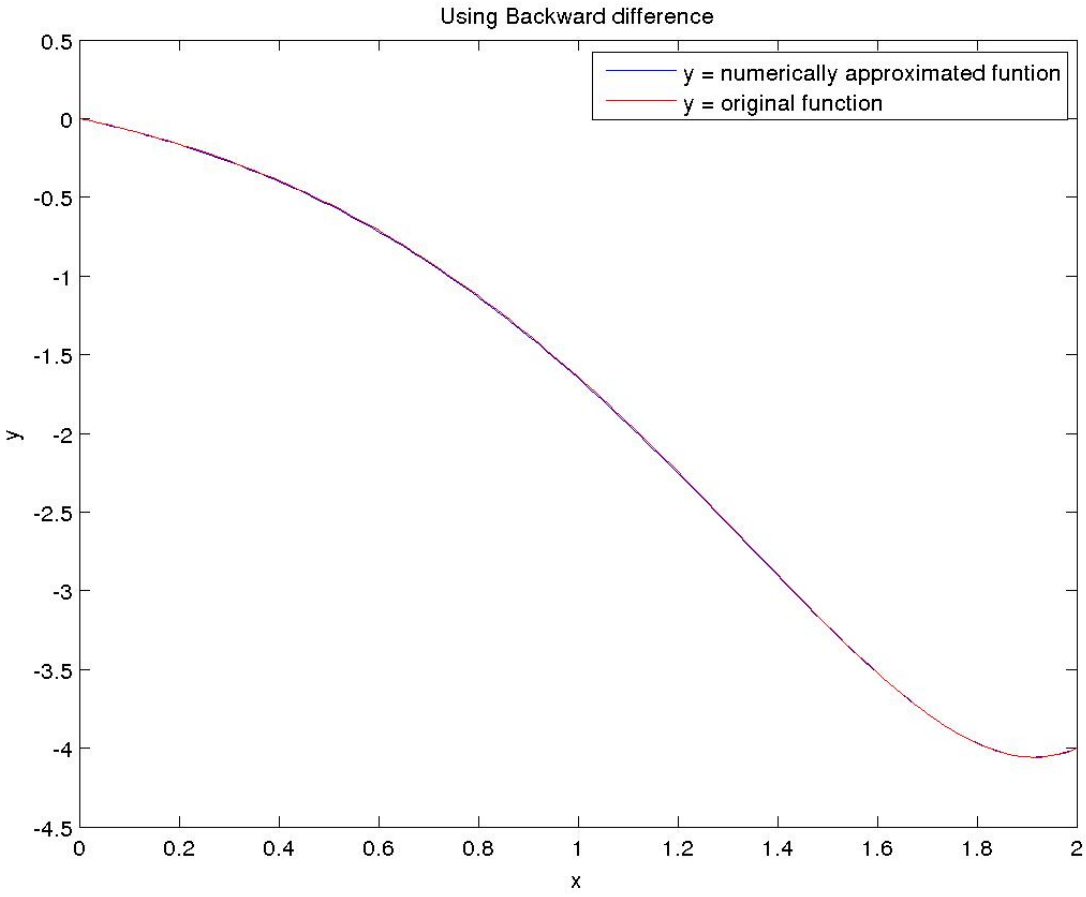

Using Forward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^0
10^-1
10^-2
10^-3
10^0
10^1
10^2
10^3


Using Forward difference
y = numerically approximated funtion
y = original function
y
x
0.5
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

Using Backward-difference for 1st order and Central-difference for 2nd order

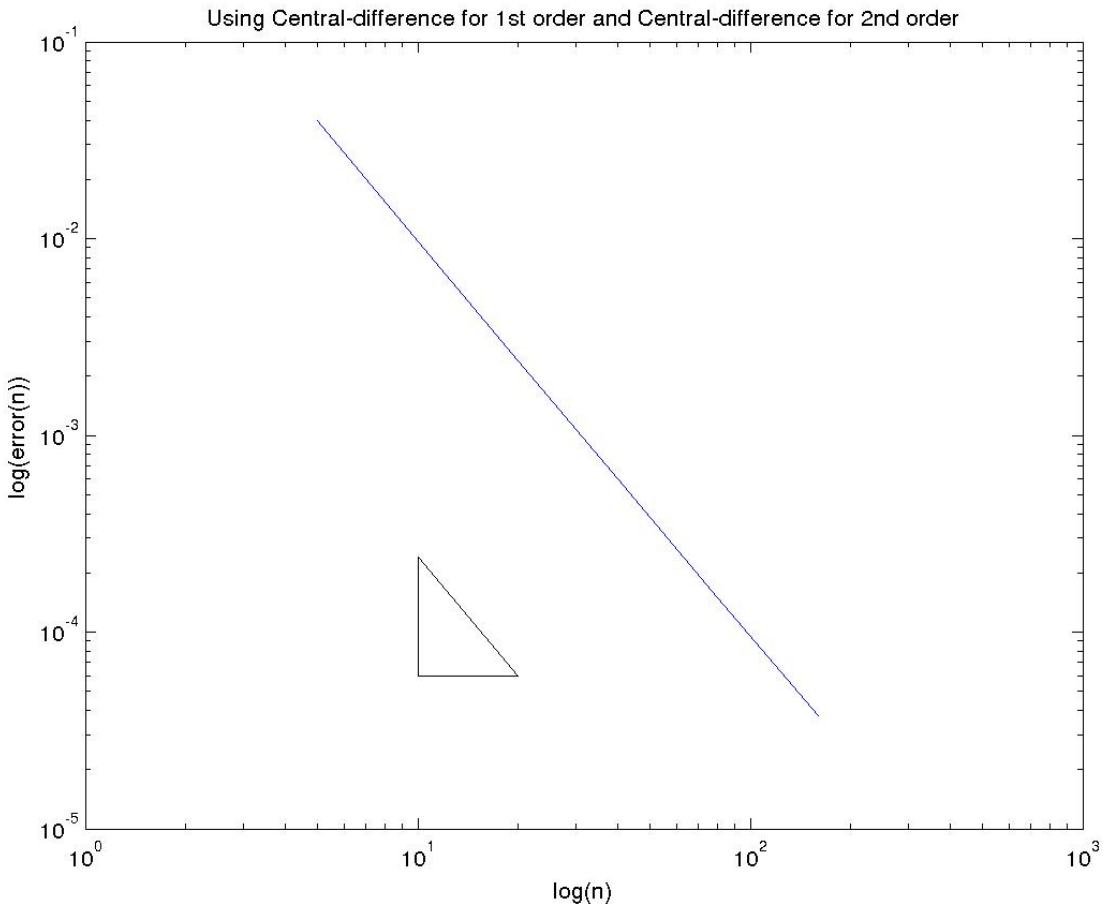
N	Error	Order
5	1.973590e-01	0.661807
10	1.247479e-01	0.832122
20	7.007123e-02	0.911586
40	3.724988e-02	0.954342
80	1.922381e-02	0.977562

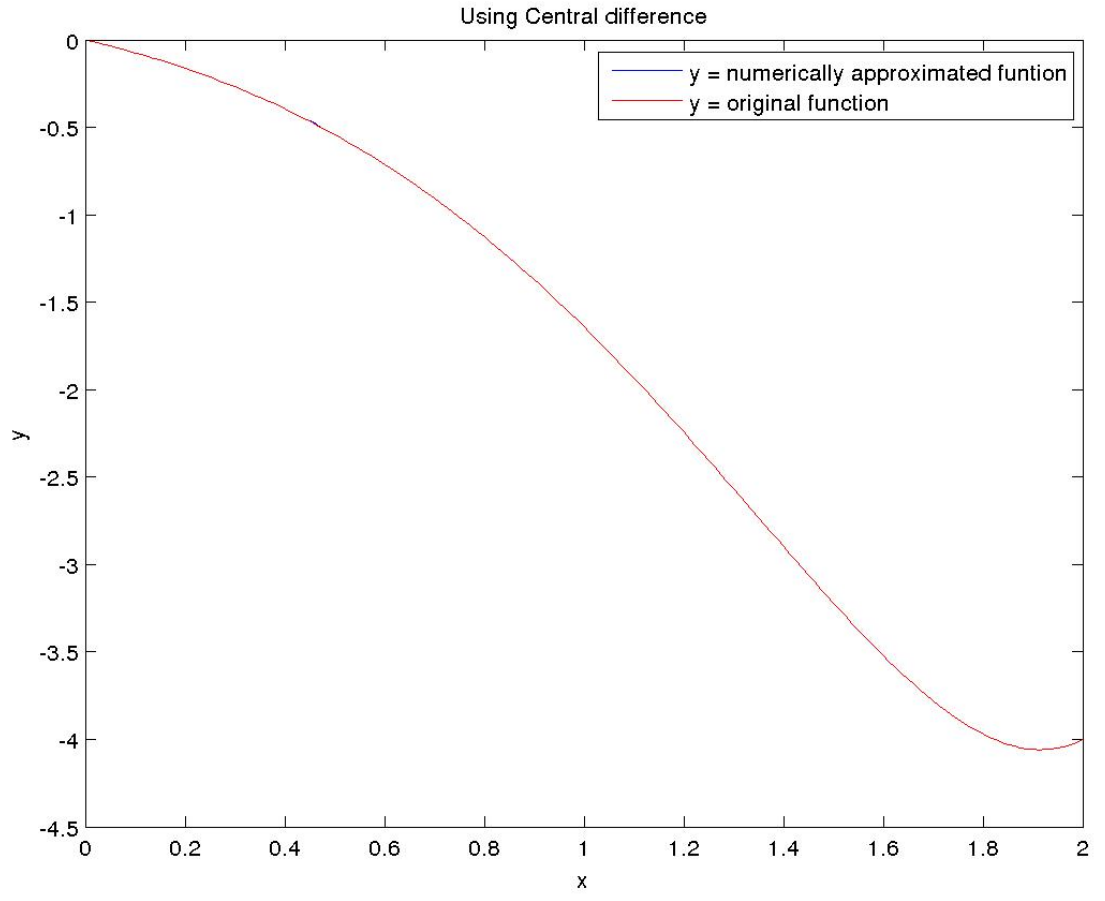




Using Central-difference for 1st order and Central-difference for 2nd order

N	Error	Order
5	3.999023e-02	2.059519
10	9.593494e-03	2.003573
20	2.392441e-03	1.995024
40	6.001770e-04	2.002832
80	1.497500e-04	1.998718

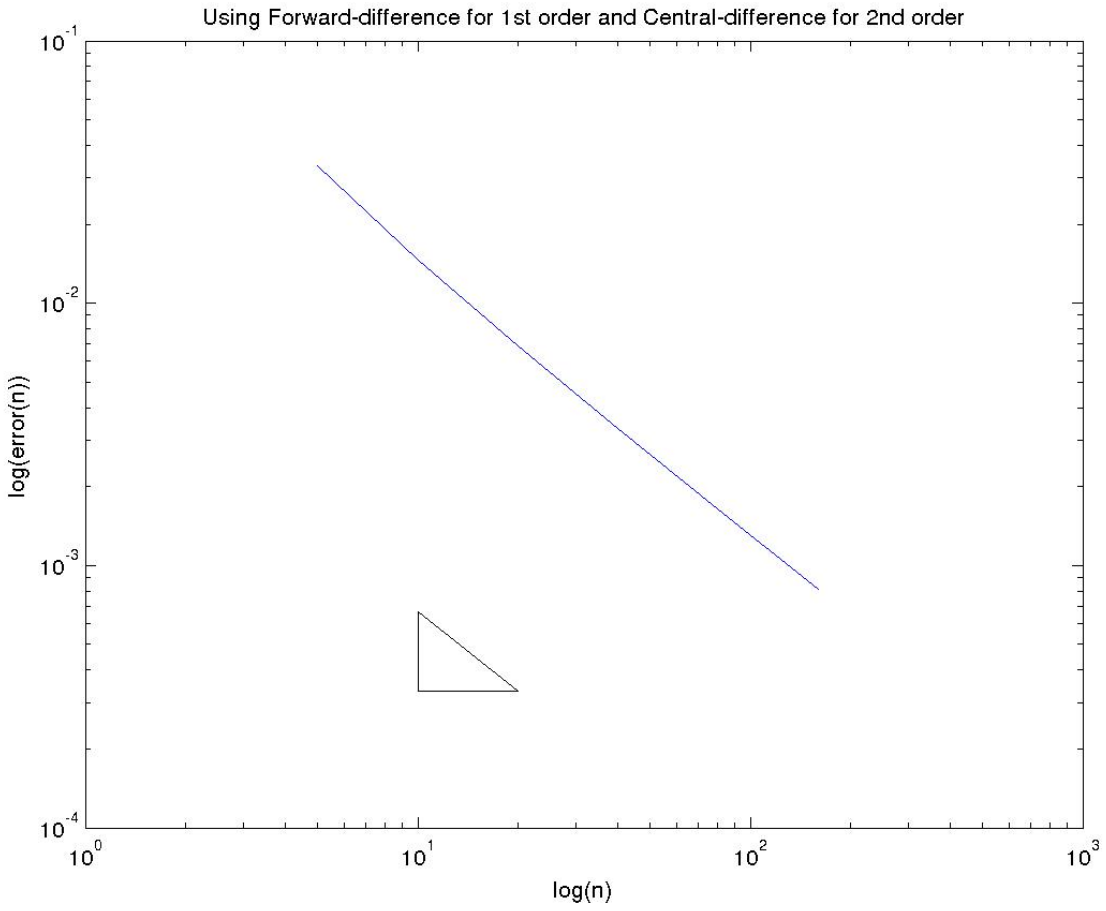

Using Central-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^-1
10^-2
10^-3
10^-4
10^-5
10^0
10^1
10^2
10^3

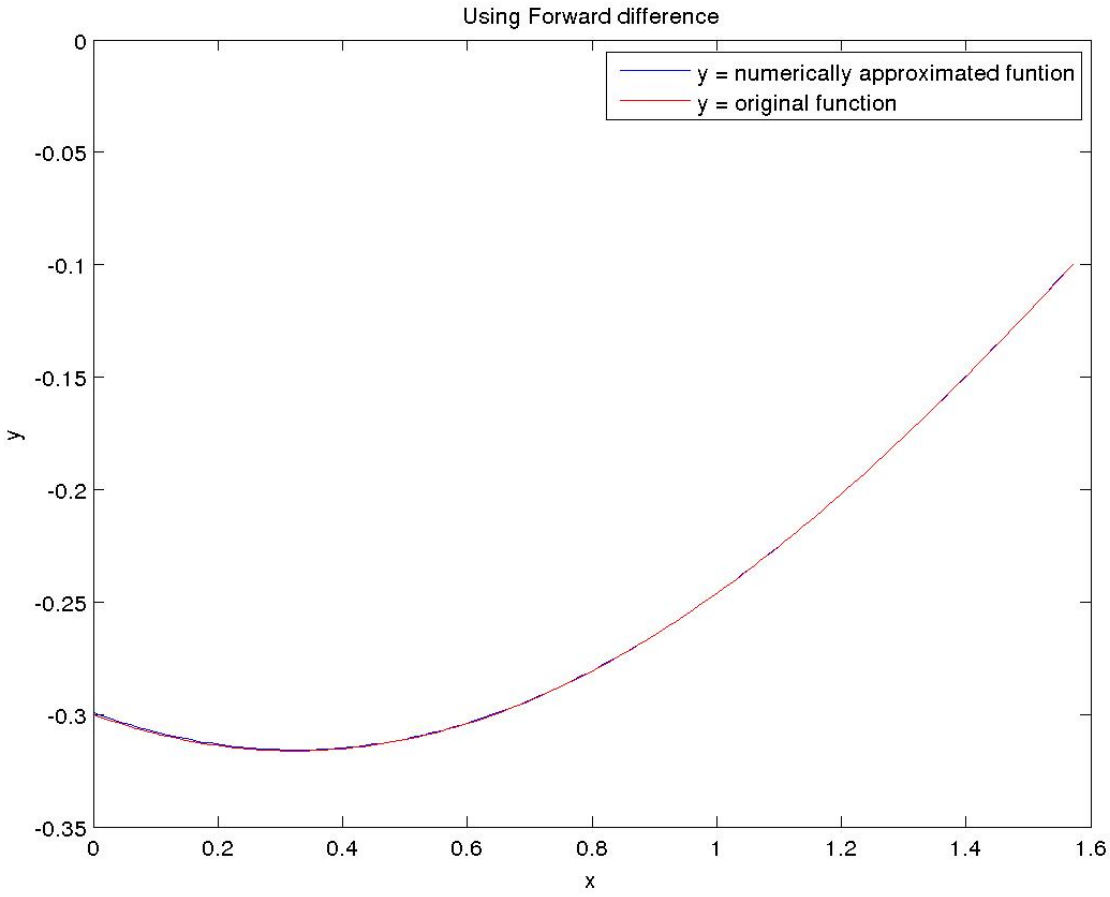

Using Central difference
y = numerically approximated funtion
y = original function
y
x
0
-0.5
-1
-1.5
-2
-2.5
-3
-3.5
-4
-4.5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

(C) Neumann Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

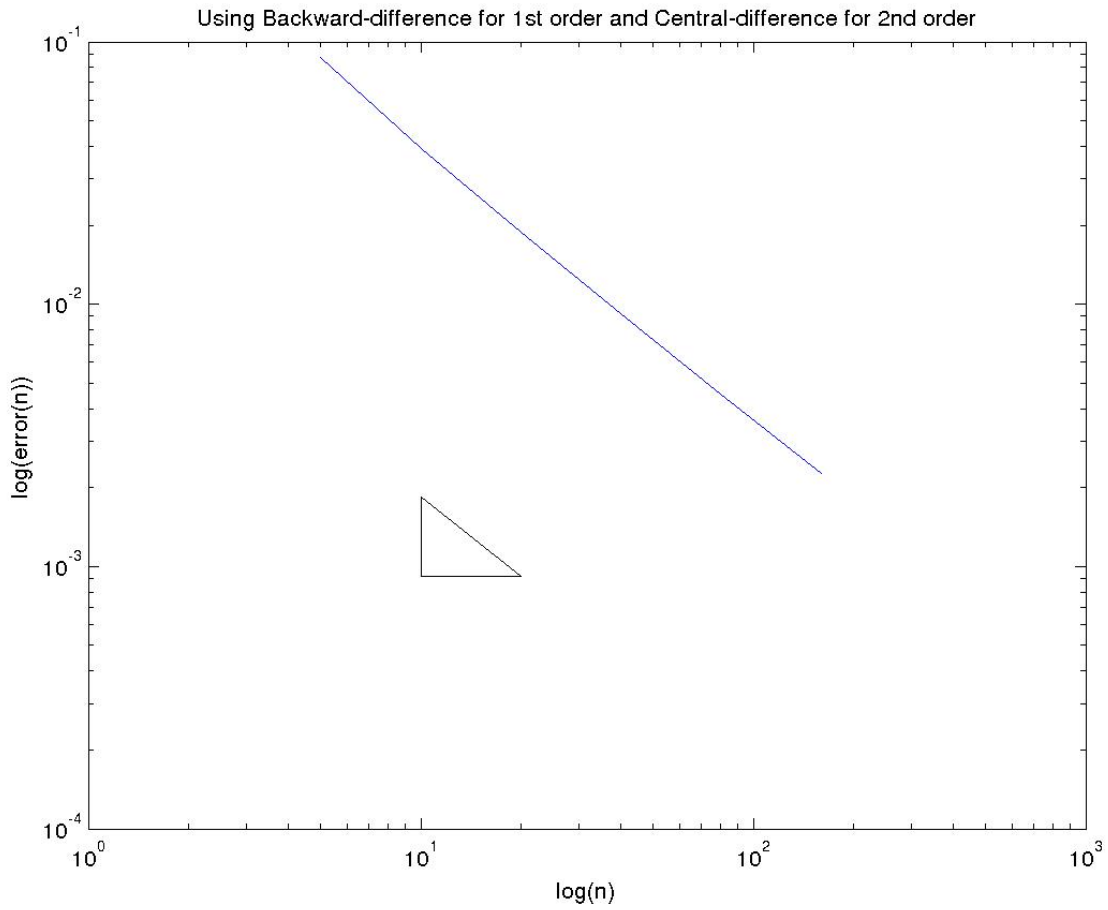
N	Error	Order
5	3.351828e-02	1.193300
10	1.465759e-02	1.092254
20	6.874822e-03	1.044981
40	3.331893e-03	1.022199
80	1.640508e-03	1.011026

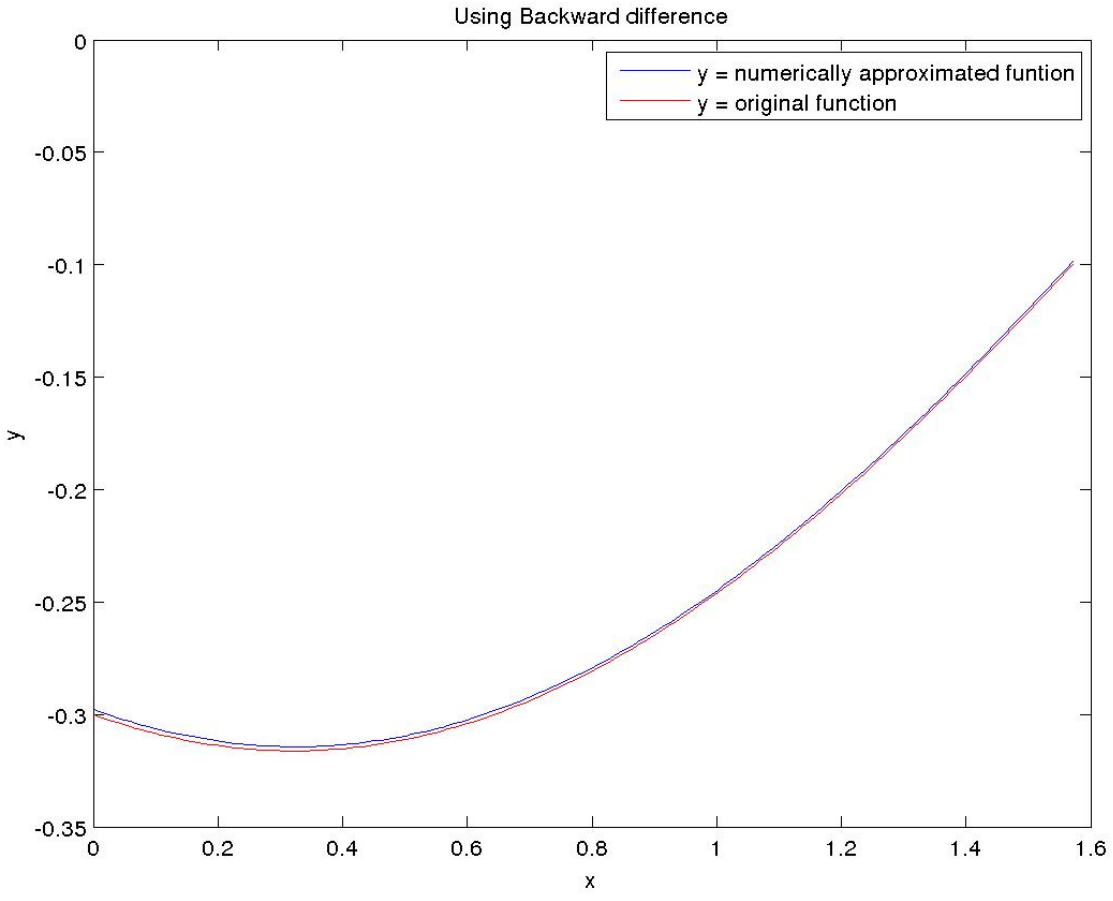




Using Backward-difference for 1st order and Central-difference for 2nd order

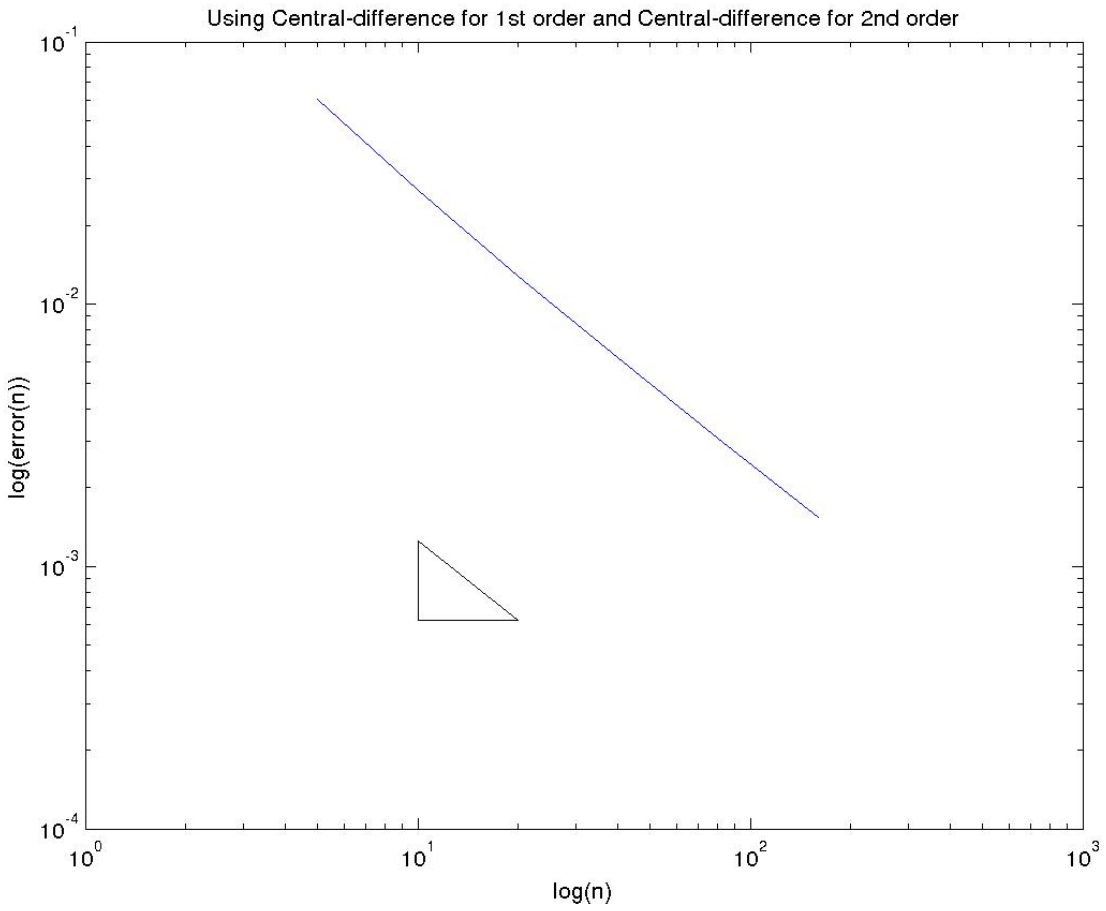
N	Error	Order
5	8.720979e-02	1.148002
10	3.935338e-02	1.068462
20	1.876476e-02	1.033054
40	9.169861e-03	1.016253
80	4.533569e-03	1.008060

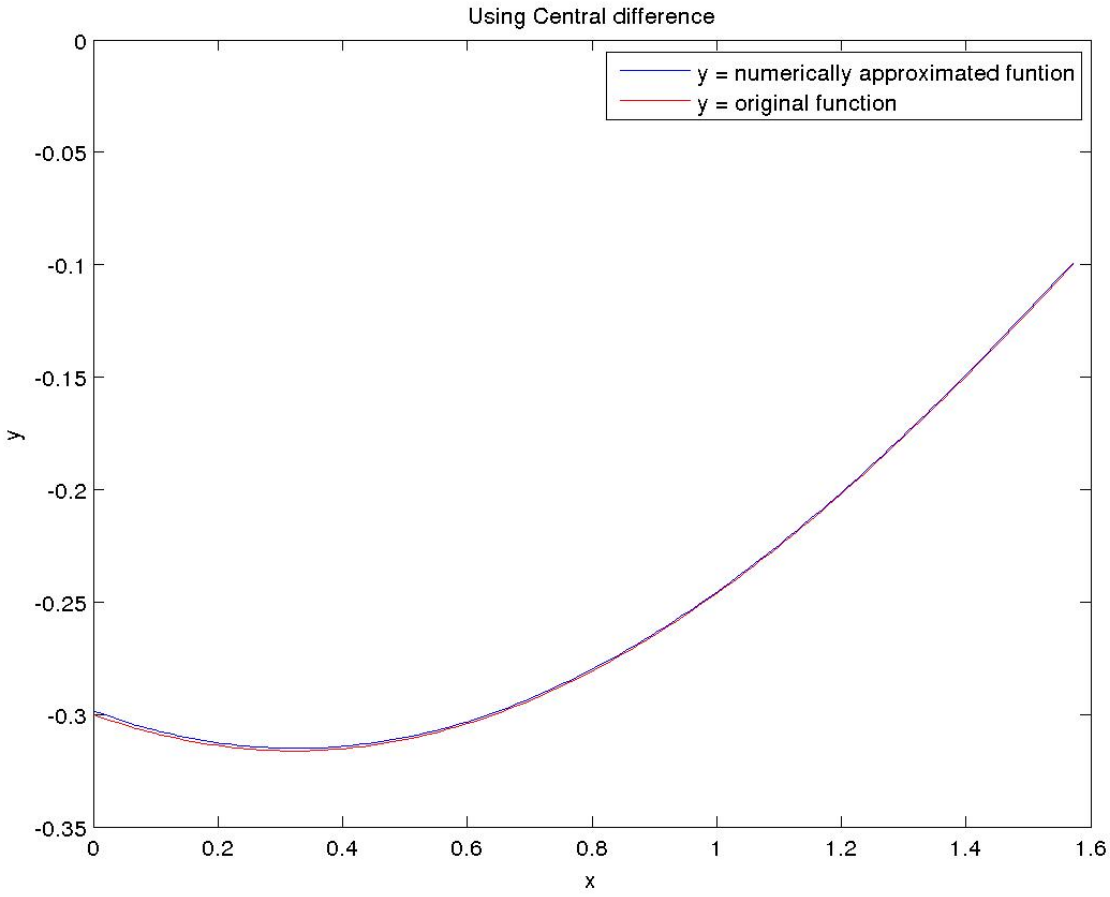

Using Backward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^-1
10^-2
10^-3
10^-4
10^0
10^1
10^2
10^3


Using Backward difference
y = numerically approximated funtion
y = original function
y
x
0
-0.05
-0.1
-0.15
-0.2
-0.25
-0.3
-0.35
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

Using Central-difference for 1st order and Central-difference for 2nd order
The central difference method for Neumann and Robin boundary condition gives order 1 accuracy because of the usage of forward and backward difference at the boundary points.

N	Error	Order
5	6.056224e-02	1.162066
10	2.706358e-02	1.076308
20	1.283466e-02	1.037056
40	6.254597e-03	1.018260
80	3.087967e-03	1.009063

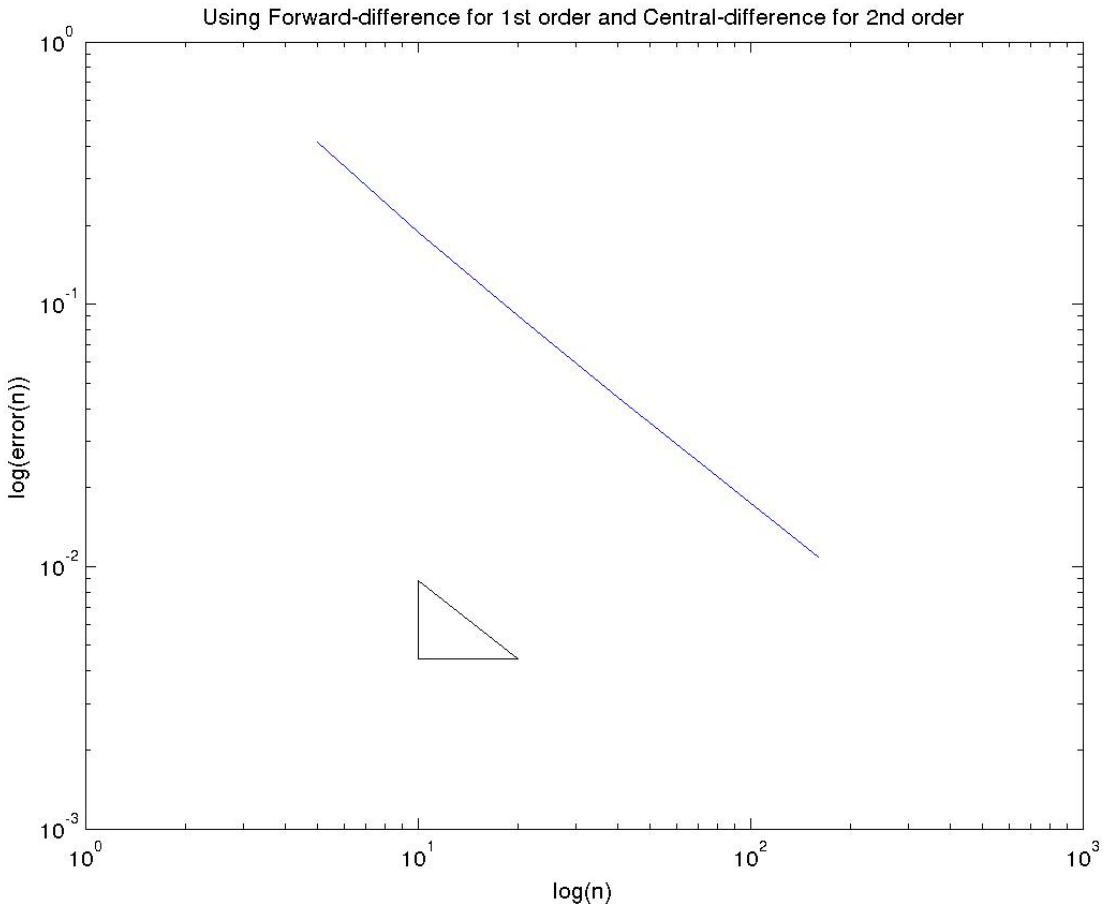


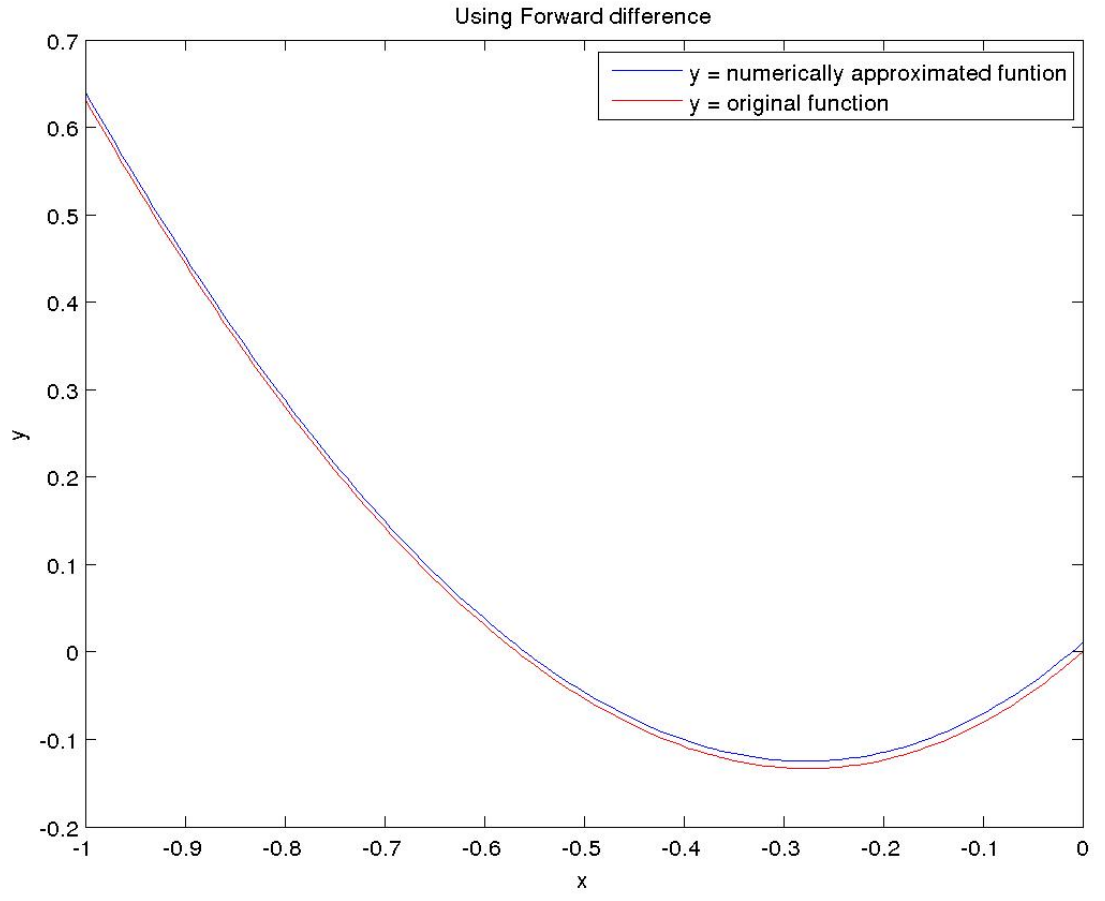


(D) Neumann Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

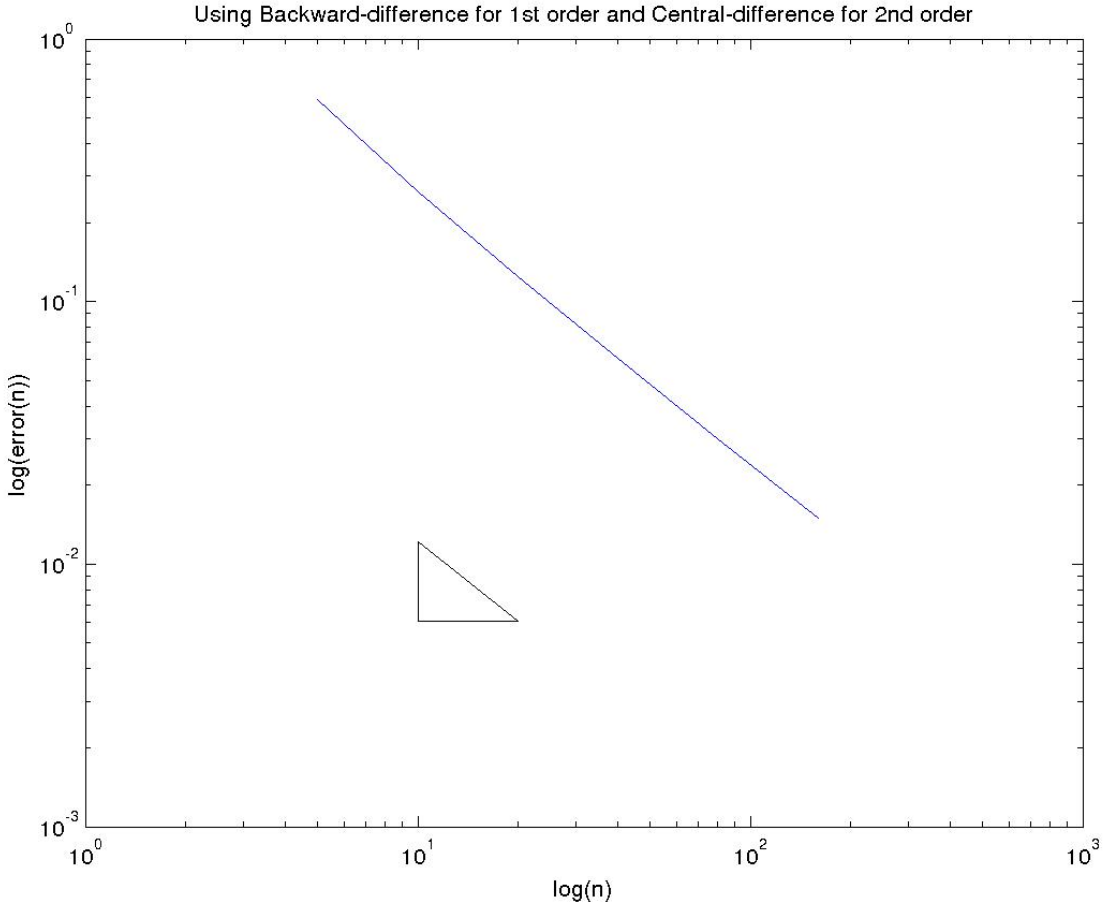
N	Error	Order
5	4.153664e-01	1.138559
10	1.886649e-01	1.062141
20	9.035553e-02	1.029541
40	4.426209e-02	1.014416
80	2.191100e-02	1.007123

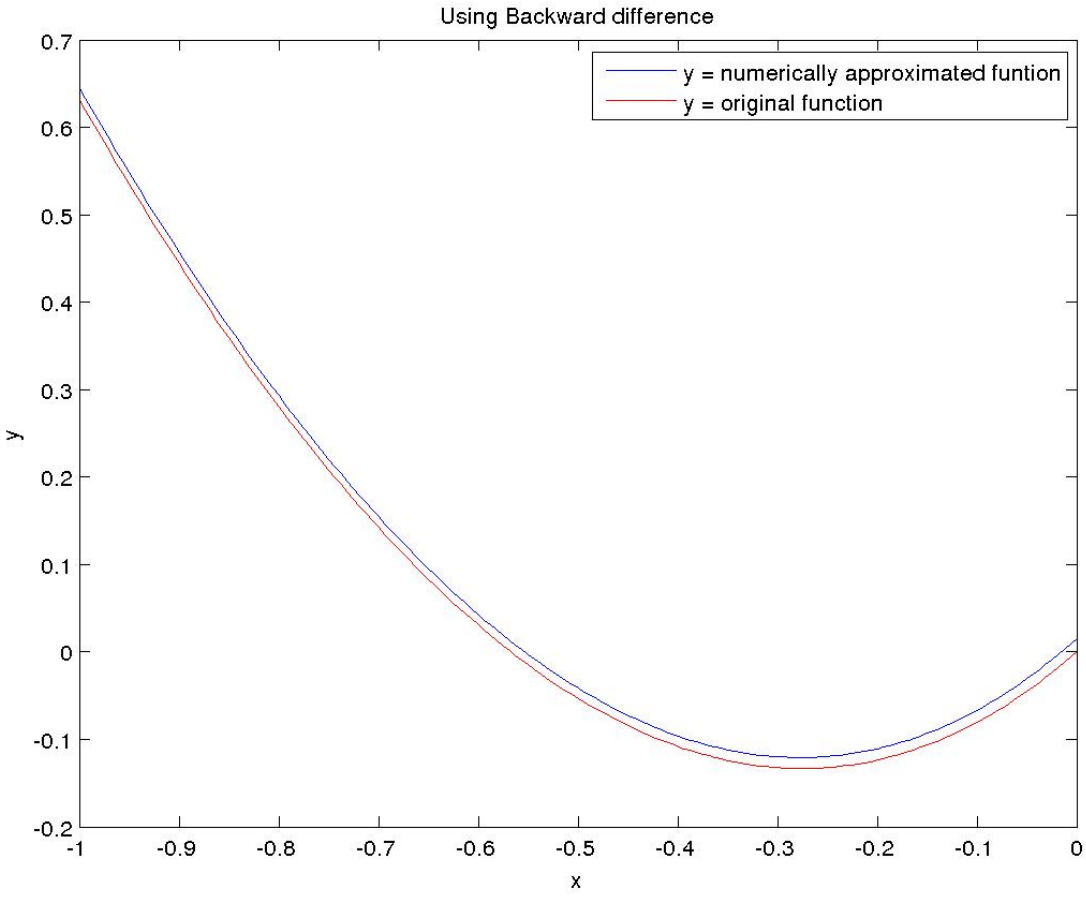

Using Forward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^0
10^-1
10^-2
10^-3
10^0
10^1
10^2
10^3


Using Forward difference
y = numerically approximated funtion
y = original function
y
x
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-1
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0

Using Backward-difference for 1st order and Central-difference for 2nd order

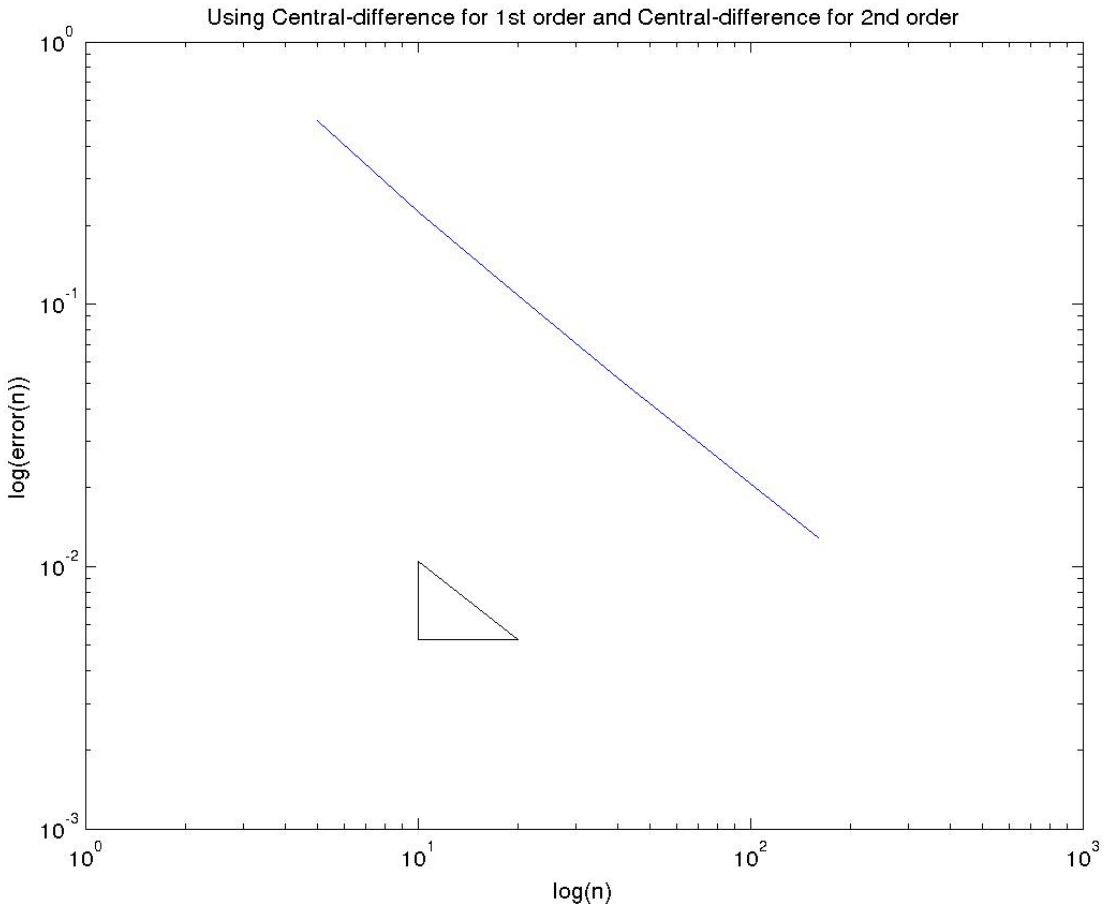
N	Error	Order
5	5.888132e-01	1.164144
10	2.627456e-01	1.075494
20	1.246750e-01	1.036317
40	6.078788e-02	1.017823
80	3.002075e-02	1.008831

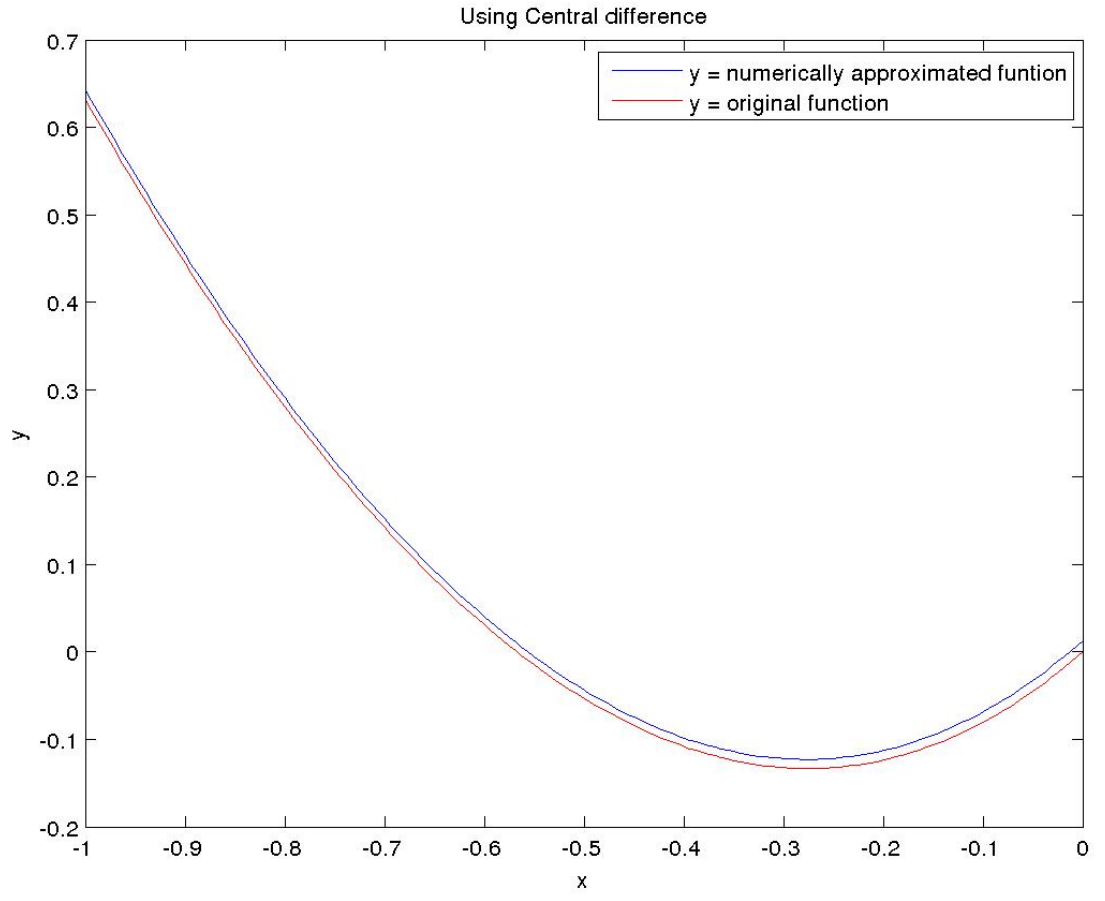




Using Central-difference for 1st order and Central-difference for 2nd order
The central difference method for Neumann and Robin boundary condition gives order 1 accuracy because of the usage of forward and backward difference at the boundary points.

N	Error	Order
5	5.026561e-01	1.154100
10	2.258667e-01	1.070364
20	1.075575e-01	1.033737
40	5.253572e-02	1.016531
80	2.596858e-02	1.008184

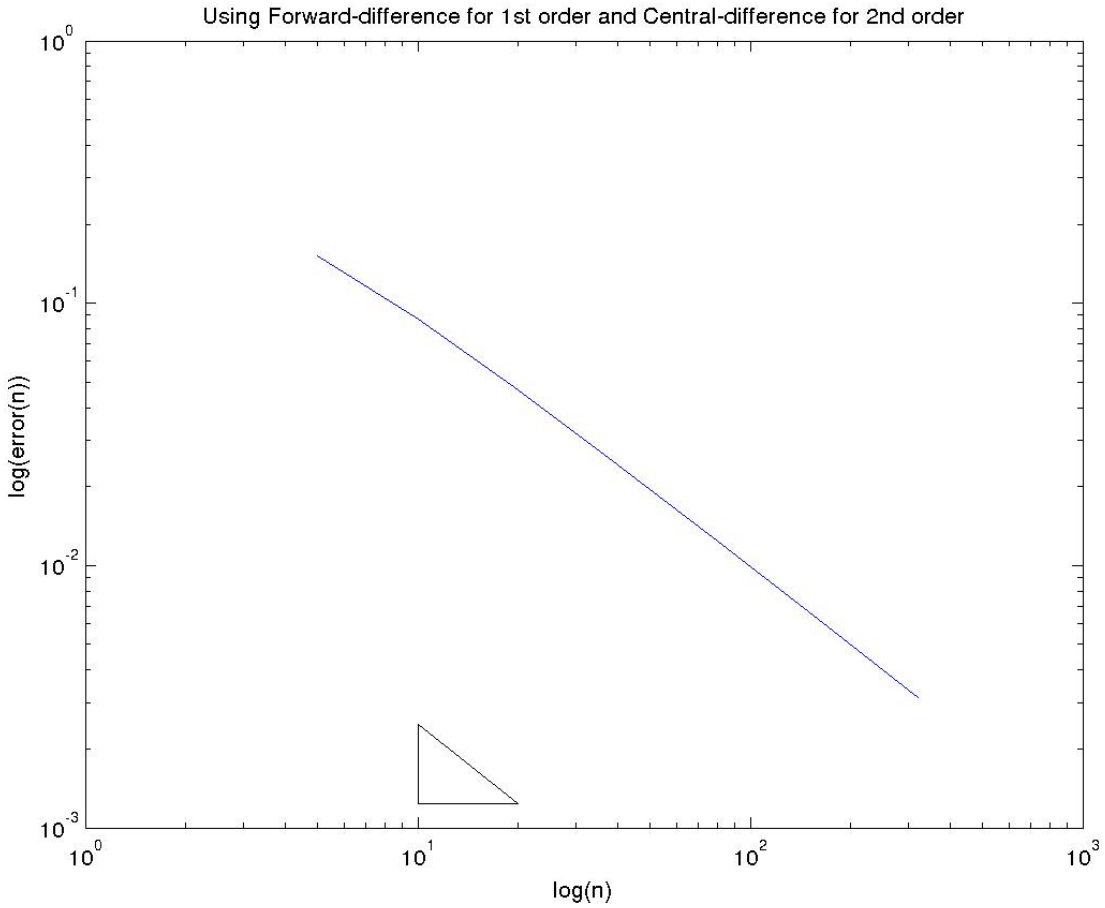

Using Central-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^0
10^-1
10^-2
10^-3
10^0
10^1
10^2
10^3

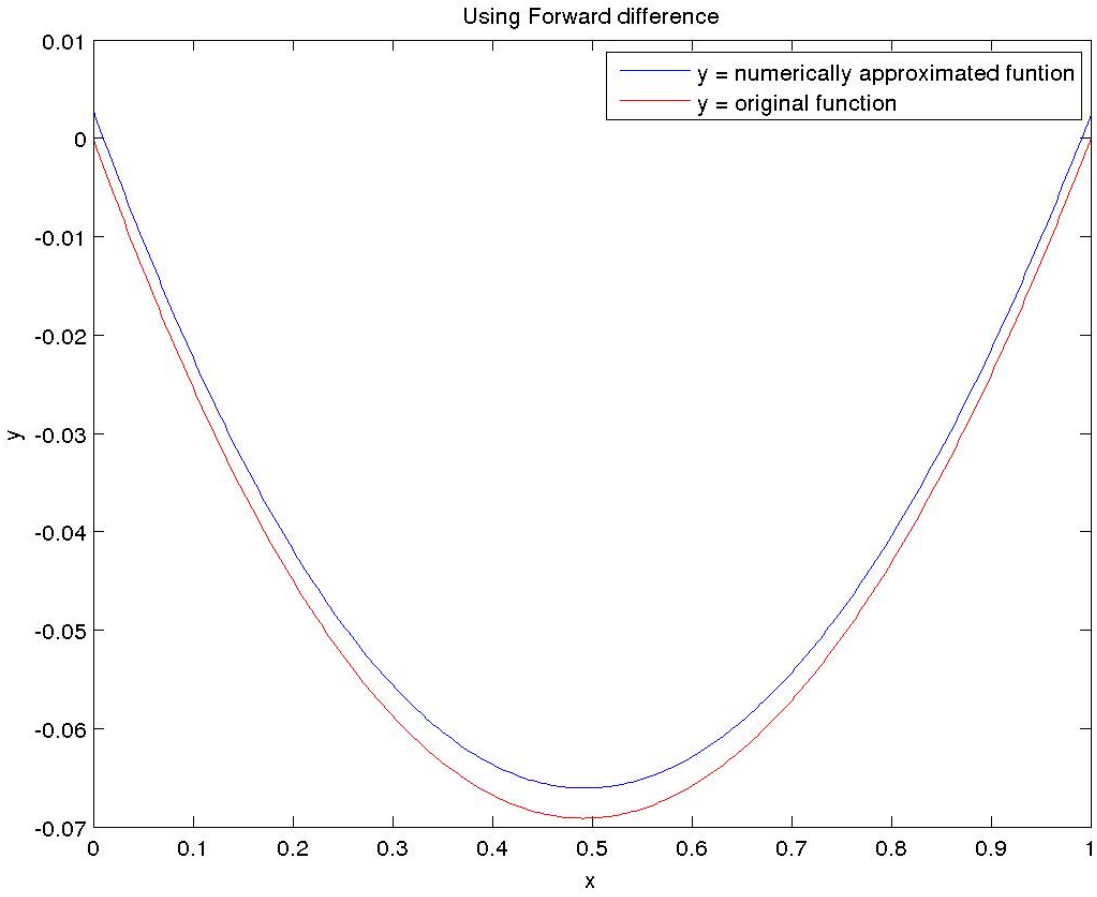

Using Central difference
y = numerically approximated funtion
y = original function
y
x
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.1
-0.2
-1
-0.9
-0.8
-0.7
-0.6
-0.5
-0.4
-0.3
-0.2
-0.1
0

(E) Robin Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

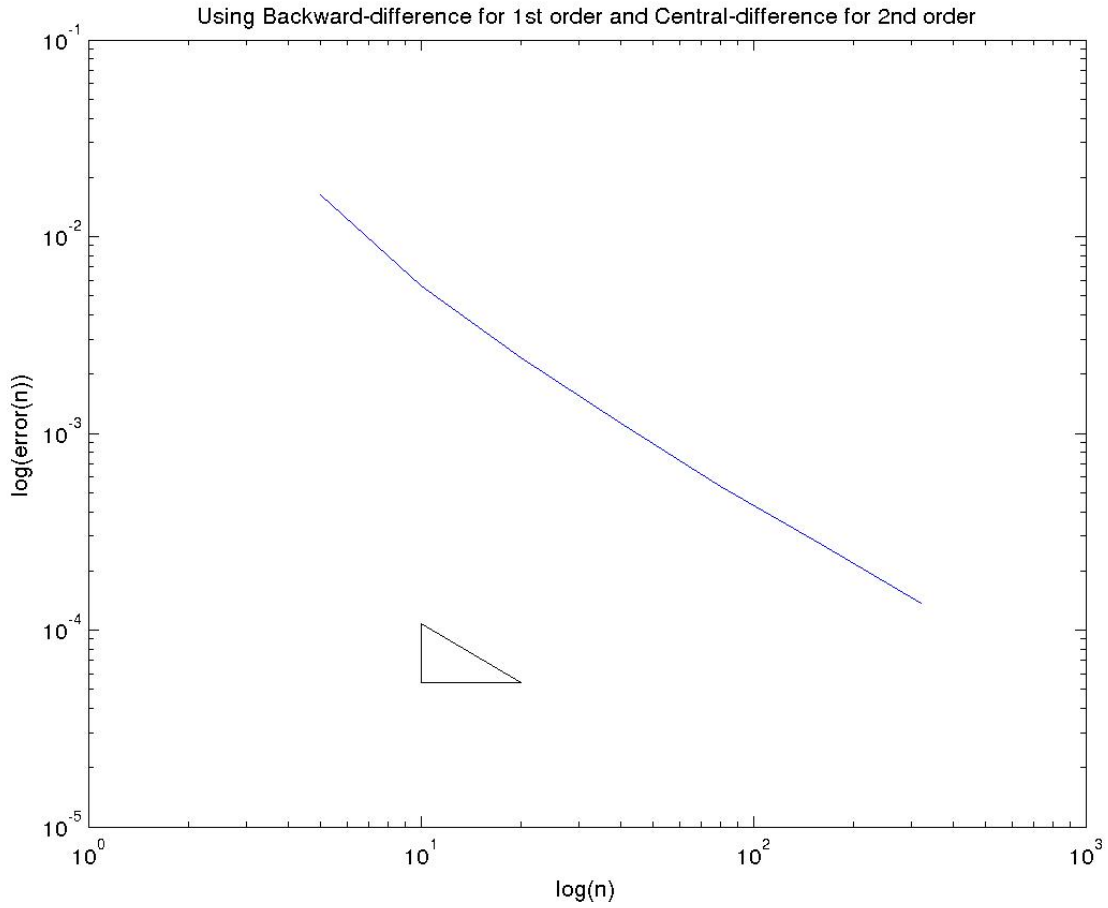
N	Error	Order
5	1.512438e-01	0.799801
10	8.687873e-02	0.892967
20	4.678469e-02	0.945379
40	2.429496e-02	0.972441
80	1.238176e-02	0.986163
160	6.250544e-03	0.993068

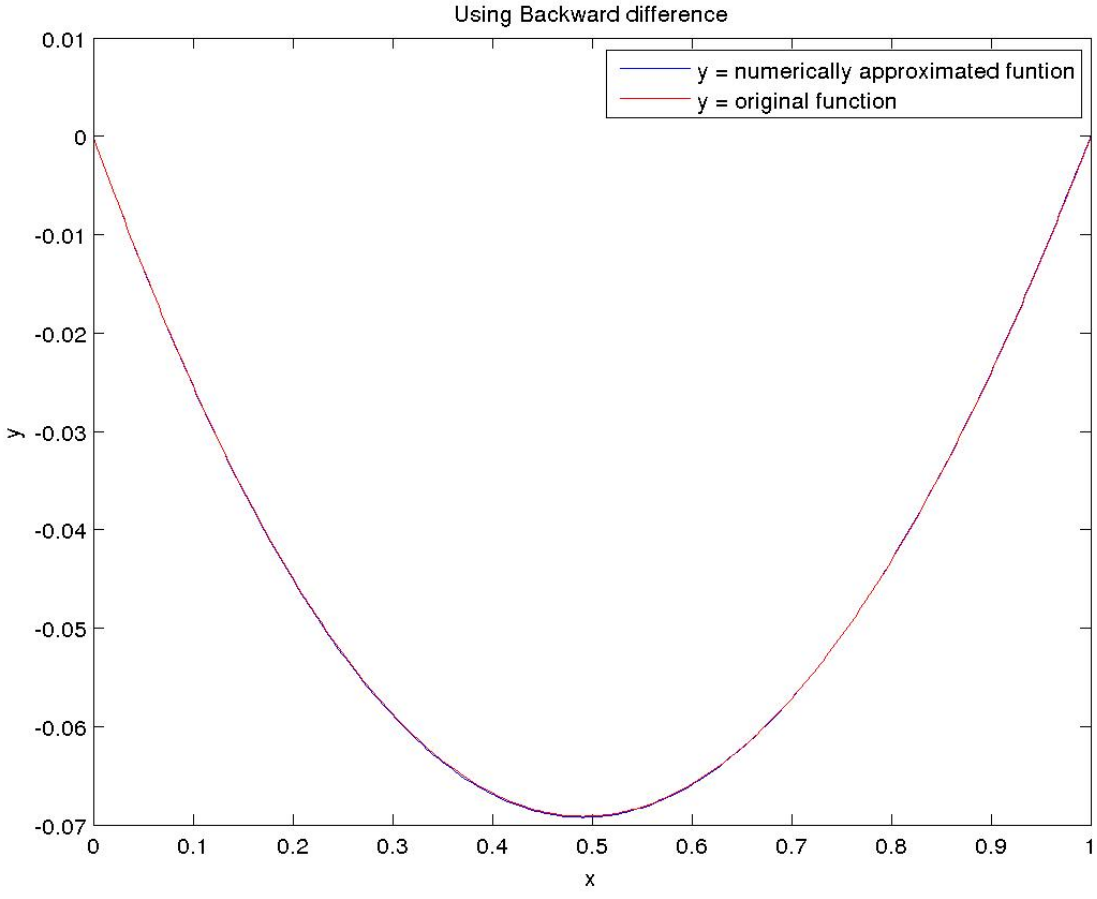




Using Backward-difference for 1st order and Central-difference for 2nd order

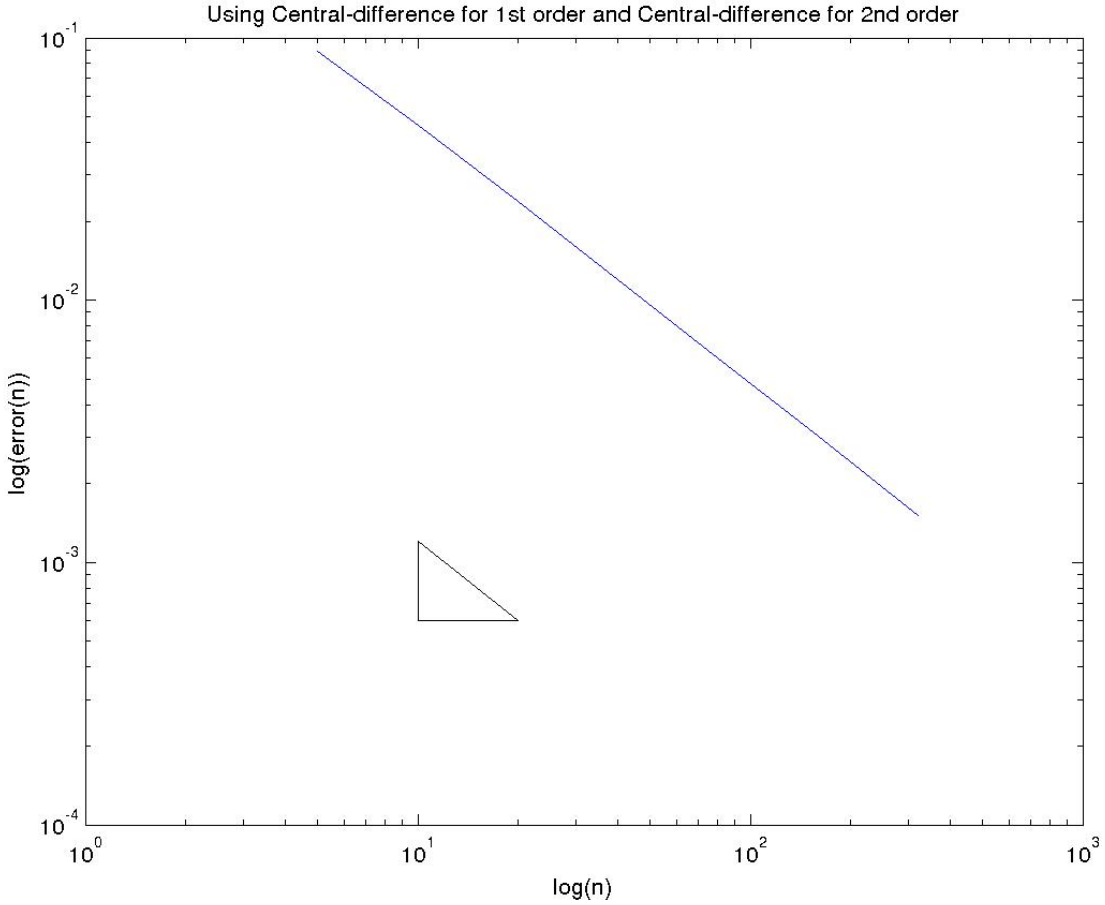
N	Error	Order
5	1.635202e-02	1.536949
10	5.635129e-03	1.222959
20	2.414107e-03	1.107945
40	1.120036e-03	1.052669
80	5.399418e-04	0.986022
160	2.725993e-04	0.992948

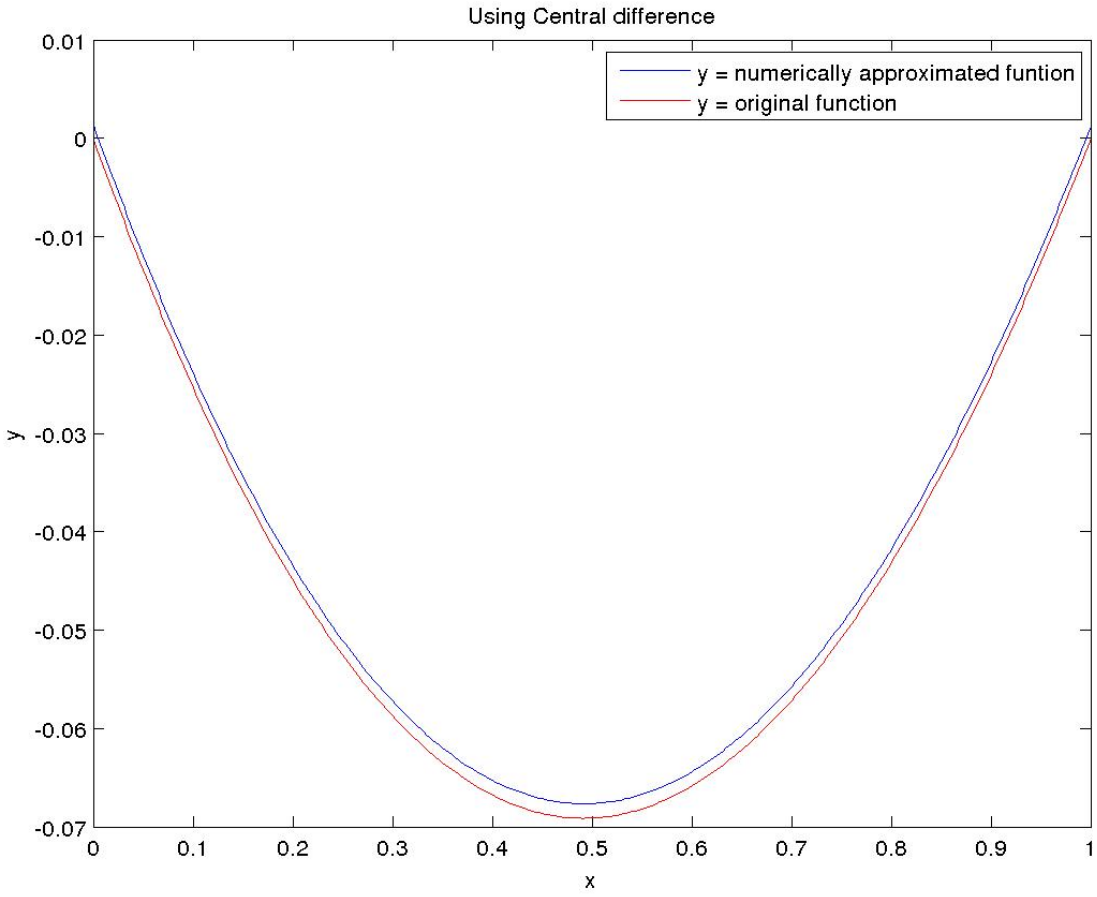

Using Backward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^-1
10^-2
10^-3
10^-4
10^-5
10^0
10^1
10^2
10^3


Using Backward difference
y = numerically approximated funtion
y = original function
y
x
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
-0.07
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Using Central-difference for 1st order and Central-difference for 2nd order
The central difference method for Neumann and Robin boundary condition gives order 1 accuracy because of the usage of forward and backward difference at the boundary points.

N	Error	Order
5	8.909580e-02	0.939644
10	4.645111e-02	0.970397
20	2.370705e-02	0.984977
40	1.197760e-02	0.992274
80	6.020960e-03	0.996155
160	3.018515e-03	0.998077

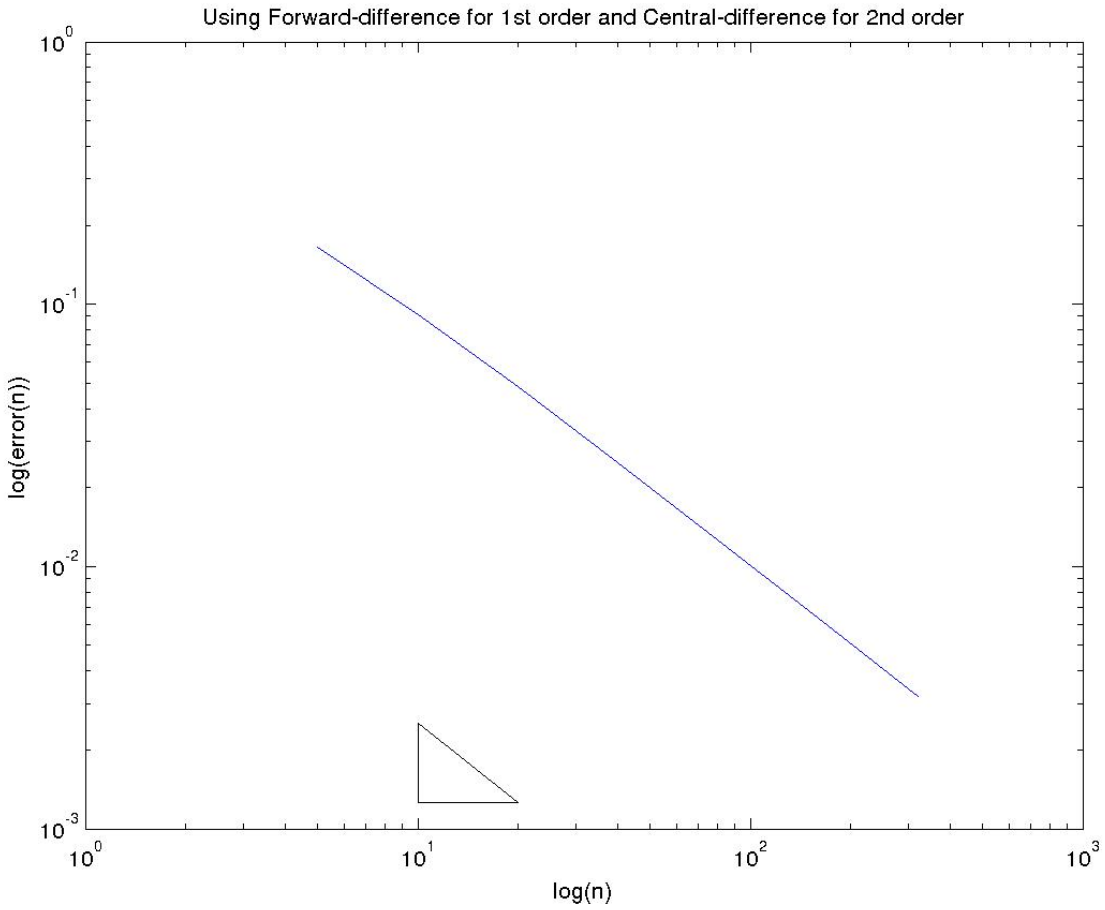


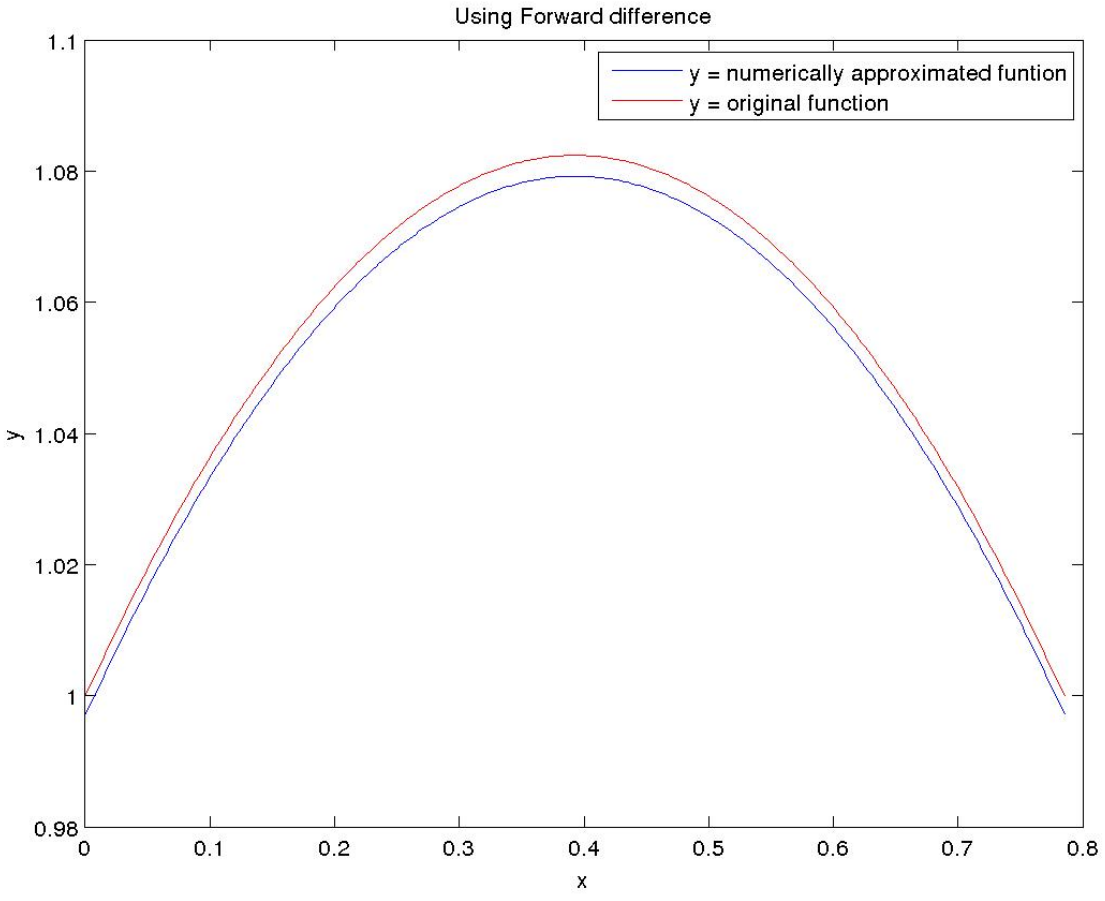


(F) Robin Boundary Condition

Using Forward-difference for 1st order and Central-difference for 2nd order

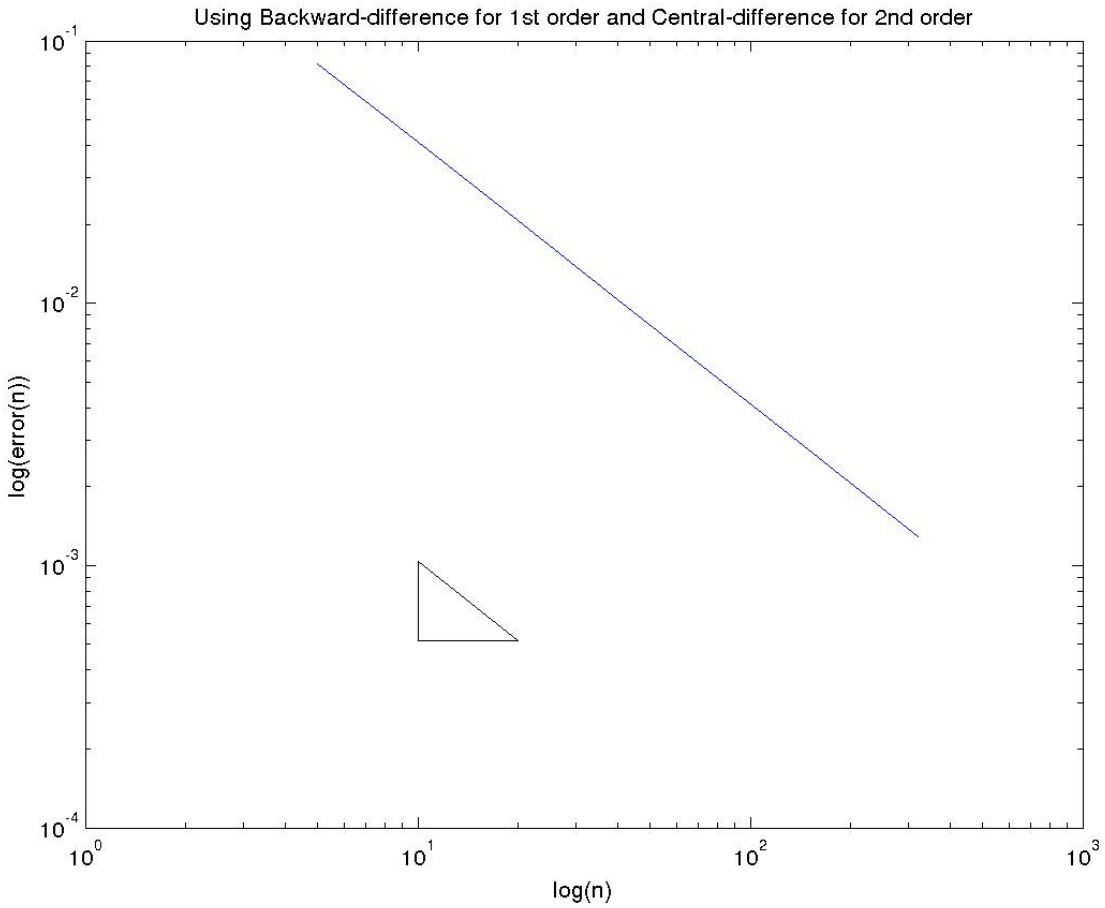
N	Error	Order
5	1.653826e-01	0.853243
10	9.154576e-02	0.920826
20	4.835507e-02	0.958658
40	2.488040e-02	0.979068
80	1.262201e-02	0.989410
160	6.357503e-03	0.994680

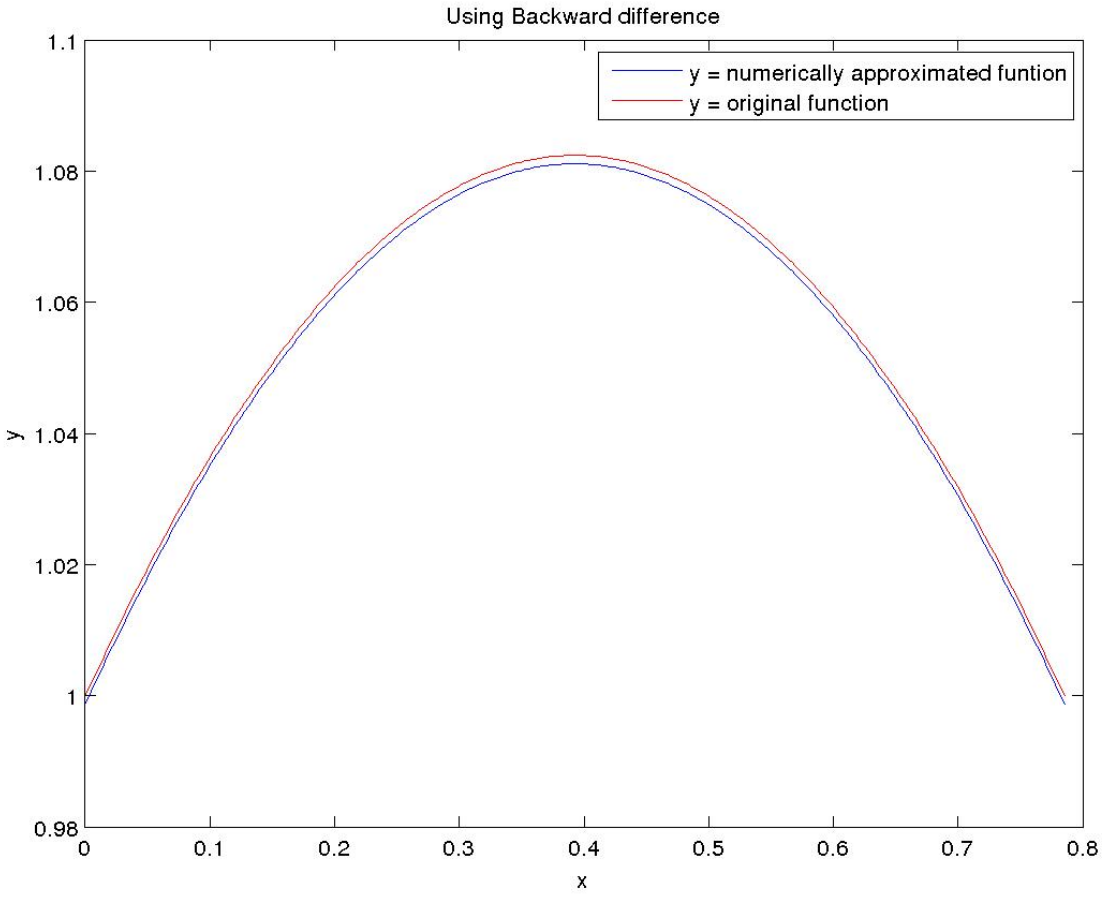

Using Forward-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{0}
10^{1}
10^{2}
10^{3}


Using Forward difference
y = numerically approximated funtion
y = original function
y
x
1.1
1.08
1.06
1.04
1.02
1
0.98
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

Using Backward-difference for 1st order and Central-difference for 2nd order

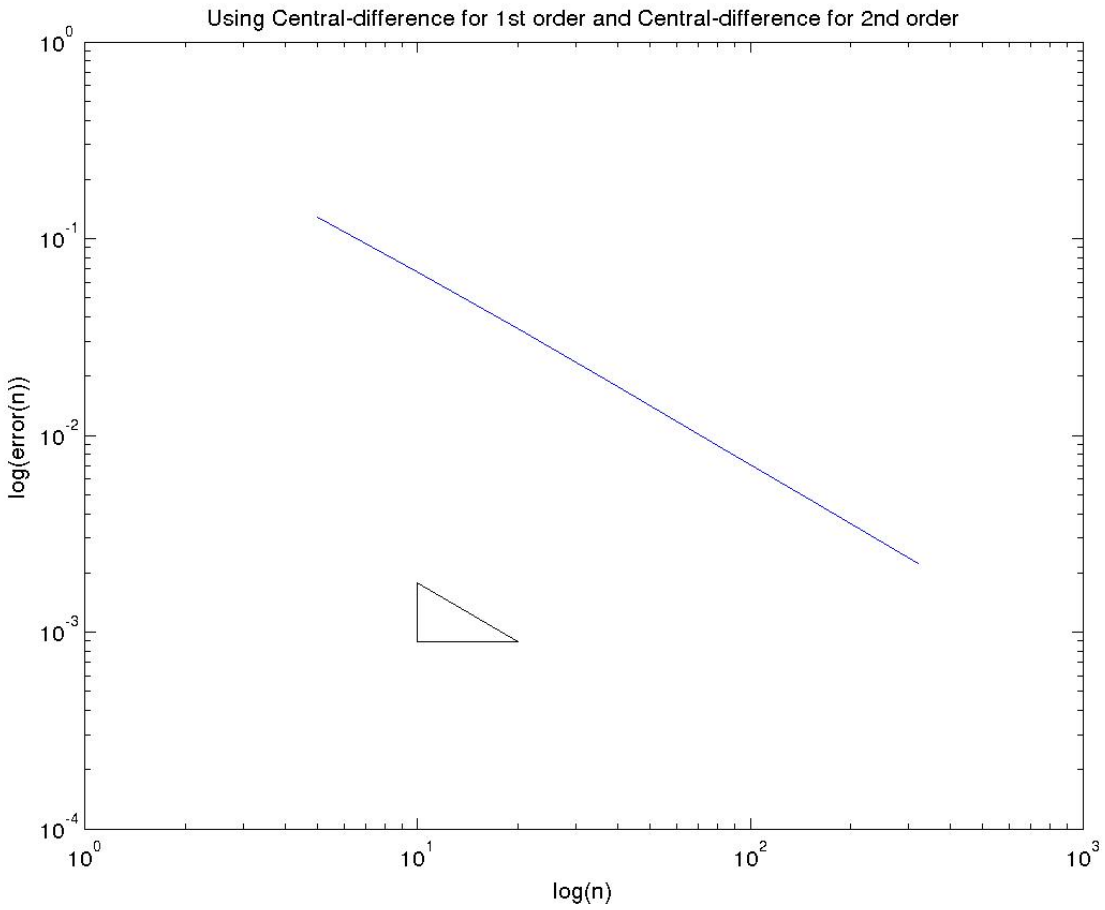
N	Error	Order
5	8.194698e-02	0.994129
10	4.114057e-02	0.996646
20	2.061816e-02	0.998275
40	1.032141e-02	0.999112
80	5.163885e-03	0.999554
160	2.582741e-03	0.999775

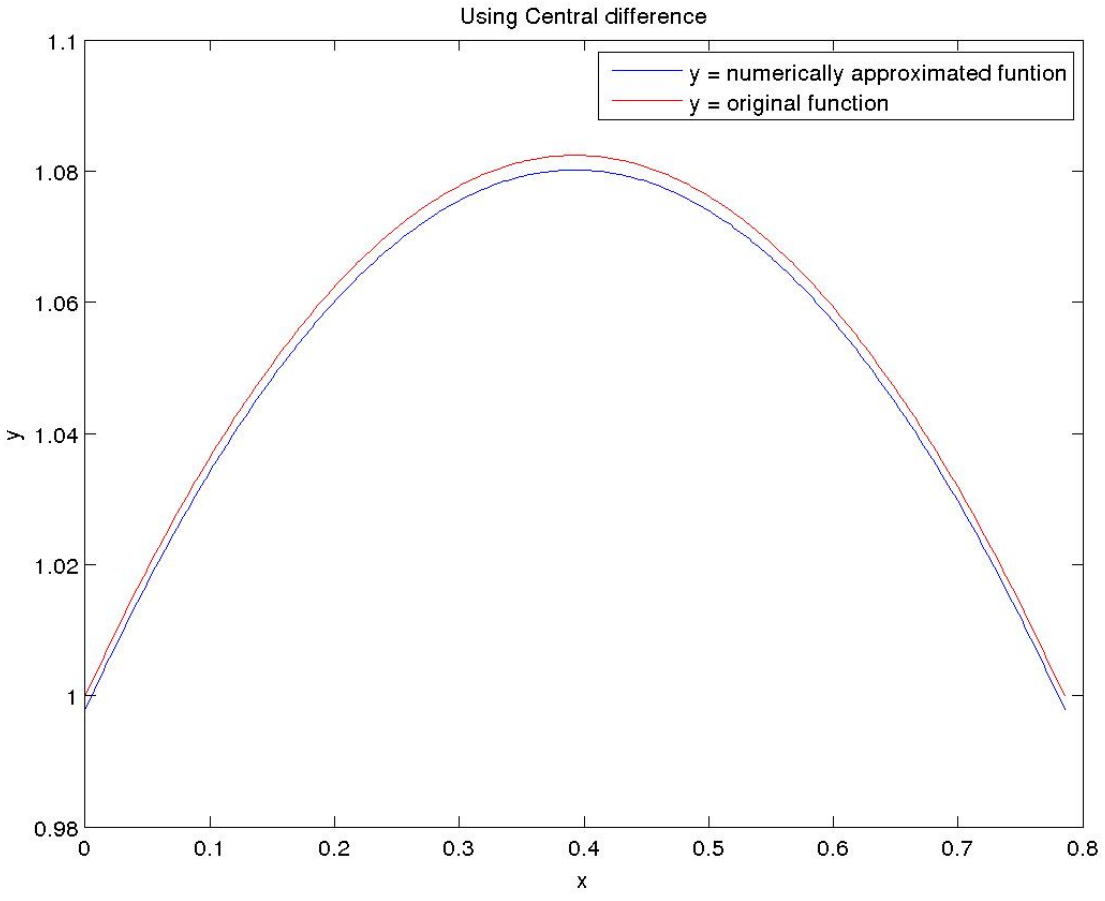




Using Central-difference for 1st order and Central-difference for 2nd order
The central difference method for Neumann and Robin boundary condition gives order 1 accuracy because of the usage of forward and backward difference at the boundary points.

N	Error	Order
5	1.282023e-01	0.920306
10	6.774171e-02	0.958090
20	3.486922e-02	0.978727
40	1.769359e-02	0.989295
80	8.912681e-03	0.994617
160	4.472999e-03	0.997298


Using Central-difference for 1st order and Central-difference for 2nd order
log(error(n))
log(n)
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{0}
10^{1}
10^{2}
10^{3}


Using Central difference
y = numerically approximated funtion
y = original function
y
x
1.1
1.08
1.06
1.04
1.02
1
0.98
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8