Assignment- 5 Submission

MA 202 Numerical Techniques (2021-22)

Name - Rajan Kumar

Roll No - 202051152

Q. 1: (a) Write a MATLAB script to calculate numerical derivative of tan1(x) at x = 1.

(b) Find the error using forward difference, backward difference, and central difference methods. Comment on order of accuracy.

(c) Plot the error using log-log scale for the step size h = 1e04.

d) Use step sizes ranging from 10 to 10.Plot the error using log-log scale for each of your step sizes. (Note: Use array operations not a for loop).

(e) Comment on trade-off between truncation error and roundooff error, i.e., look at the minima of di↵erent methods in the graphs.

```
% Matlab script to calculate numerical differentiation of arctan(x)
clc;
clear;
close all;
a = 1;
truVal = 1/(1 + a.^2);
h = 10.^[-1:-1:-8];
% Forward difference formula
fwdDiff = (f(a + h) - f(a))./h;
errFwd = abs(truVal - fwdDiff);
disp(['Error in forward difference: ', num2str(errFwd)]);
```

```
Error in forward difference: 0.024169    0.0024917   0.00024992   2.4999e-05      2.5e-06   2.5006e-07   2.4133e-08   3.038
```

```
% Backward difference formula
bckDiff = (f(a) - f(a - h))./h;
errBck = abs(truVal - bckDiff);
disp(['Error in backward difference: ', num2str(errBck)]);
```

```
Error in backward difference: 0.025831    0.0025083   0.00025008   2.5001e-05      2.5e-06   2.4998e-07   2.4717e-08   3.03
```

```
% Central difference formula
cntrDiff = (f(a + h) - f(a - h))./(2.*h);
errCntr = abs(truVal - cntrDiff);
disp(['Error in central difference: ', num2str(errCntr)]);
```

```
Error in central difference: 0.00083082   8.3331e-06   8.3333e-08   8.3317e-10   8.8267e-12   4.1133e-11   2.9193e-10   3.0
```

```
% Improved Forward difference formula
```

```
iFwdDiff = (-f(a + 2*h) + 4*f(a + h) - 3*f(a))./(2.*h);
errIFwd = abs(truVal - iFwdDiff);
disp(['Error in improved forward difference: ', num2str(errIFwd)]);
```

```
Error in improved forward difference: 0.0016374   1.6663e-05   1.6667e-07   1.6665e-09   3.2756e-12   1.5216e-10   1.4022e
```

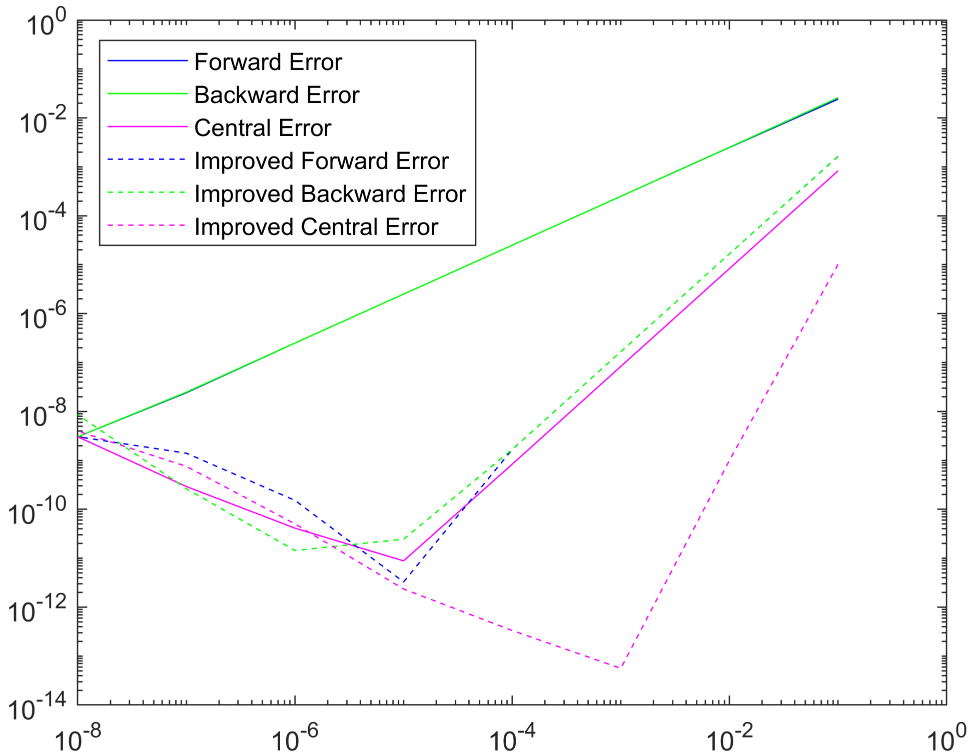
```
% Improved Backward difference formula
iBckDiff = (3*f(a) - 4*f(a - h) + f(a - 2*h))./(2.*h);
errIBck = abs(truVal - iBckDiff);
disp(['Error in improved backward difference: ', num2str(errIBck)]);
```

```
Error in improved backward difference: 0.0016249   1.6663e-05   1.6667e-07   1.6682e-09    2.448e-11   1.4378e-11   2.6318
```

```
% Improved Central difference formula
iCntrDiff = (-f(a + 2*h) + 8*f(a + h) - 8*f(a - h) + f(a - 2*h))./(12.*h);
errICntr = abs(truVal - iCntrDiff);
disp(['Error in improved central difference: ', num2str(errICntr)]);
```

```
Error in improved central difference: 1.0176e-05   1.0002e-09   5.5955e-14   3.3268e-13   2.3505e-12   5.0385e-11   7.5453
```

```
% Plots
loglog(h, errFwd, '-b', h, errBck, '-g', h, errCntr, '-m');
hold on;
loglog(h, errIFwd, '--b', h, errIBck, '--g', h, errICntr, '--m');
legend('Forward Error', 'Backward Error', 'Central Error', 'Improved Forward Error', 'Improved
```




```
% Function


%Truncation Error is caused by a truncated Taylor series expansion replacing the spatial deriva
```

Q. 2: Write a MATLAB script to calculate first order as well as second order numerical derivative of 2x + ln(x) at x = 1.

Repeat the steps (b), (c), (d), and (e) of Q.1

```
% Matlab script to calculate first order numerical differentiation of 2 # x + ln(x) at x = 1
clc;
clear;
close all;
a = 1;
truVal = 0;
h = 10.^[-1:-1:-8];
% Forward difference formula
fwdDiff = (f(a + h) - f(a))./h;
errFwd = abs(truVal - fwdDiff);
disp(['Error in forward difference: ', num2str(errFwd)]);
```

```
Error in forward difference: 0.47583      0.49751      0.49975      0.49998          0.5          0.5          0.5
```

```
% Backward difference formula
bckDiff = (f(a) - f(a - h))./h;
errBck = abs(truVal - bckDiff);
disp(['Error in backward difference: ', num2str(errBck)]);
```

```
Error in backward difference: 0.52583      0.50251      0.50025      0.50003          0.5          0.5          0.5
```

```
% Central difference formula
cntrDiff = (f(a + h) - f(a - h))./(2.*h);
errCntr = abs(truVal - cntrDiff);
disp(['Error in central difference: ', num2str(errCntr)]);
```

```
Error in central difference: 0.50083      0.50001          0.5          0.5          0.5          0.5          0.5
```

```
% Improved Forward difference formula
iFwdDiff = (-f(a + 2*h) + 4*f(a + h) - 3*f(a))./(2.*h);
errIFwd = abs(truVal - iFwdDiff);
disp(['Error in improved forward difference: ', num2str(errIFwd)]);
```

```
Error in improved forward difference: 0.49836      0.49998          0.5          0.5          0.5          0.5          0
```

```
% Improved Backward difference formula
iBckDiff = (3*f(a) - 4*f(a - h) + f(a - 2*h))./(2.*h);
errIBck = abs(truVal - iBckDiff);
disp(['Error in improved backward difference: ', num2str(errIBck)]);
```

```
Error in improved backward difference: 0.49838      0.49998          0.5          0.5          0.5          0.5
```

```
% Improved Central difference formula
iCntrDiff = (-f(a + 2*h) + 8*f(a + h) - 8*f(a - h) + f(a - 2*h))./(12.*h);
```

```
errICntr = abs(truVal - iCntrDiff);
disp(['Error in improved central difference: ', num2str(errICntr)]);
```

```
Error in improved central difference: 0.50001         0.5         0.5         0.5         0.5         0.5         0
```

```
% Plots
loglog(h, errFwd, '-b', h, errBck, '-g', h, errCntr, '-m');
hold on;
loglog(h, errIFwd, '--b', h, errIBck, '--g', h, errICntr, '--m');
legend('Forward Error', 'Backward Error', 'Central Error', 'Improved Forward Error', 'Improved
hold on;


% Matlab script to calculate second order numerical differentiation of 2 # x + ln(x) at x = 1
clc;
clear;
close all;
a = 1;
truVal = -1;
h = 10.^[-1:-1:-8];
% Forward difference formula for second order derivative
fwdDiff = (f(a + 2*h) - 2*f(a + h) + f(a))./(h.*h);
errFwd = abs(truVal - fwdDiff);
disp(['Error in forward difference for second order derivative: ',num2str(errFwd)]);
```

```
Error in forward difference for second order derivative: 0.54937       0.505       0.5005      0.50005         0.5
```

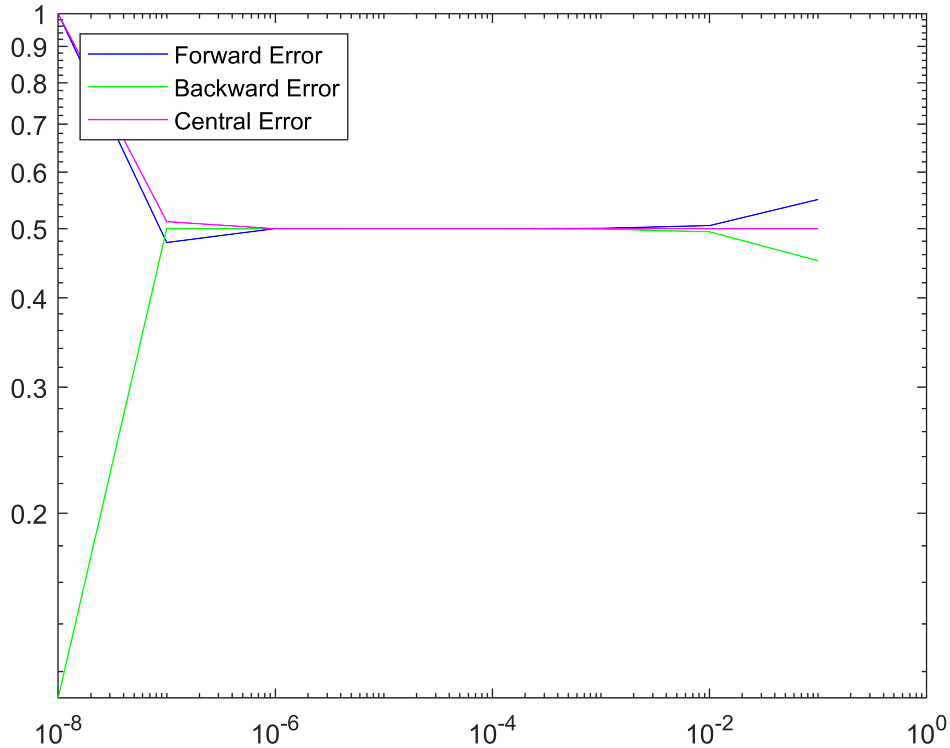
```
% Backward difference formula for second order derivative
bckDiff = (f(a) - 2*f(a - h) + f(a - 2*h))./(h.*h);
errBck = abs(truVal - bckDiff);
disp(['Error in backward difference for second order derivative: ',num2str(errBck)]);
```

```
Error in backward difference for second order derivative: 0.45089       0.495       0.4995      0.49995     0.49999
```

```
% Central difference formula for second order derivative
cntrDiff = (f(a + h) - 2*f(a) + f(a - h))./(h.*h);
errCntr = abs(truVal - cntrDiff);
disp(['Error in central difference for second order derivative: ',num2str(errCntr)]);
```

```
Error in central difference for second order derivative: 0.5         0.5         0.5         0.5         0.5     0.
```

```
% Plots
loglog(h, errFwd, '-b', h, errBck, '-g', h, errCntr, '-m');
legend('Forward Error', 'Backward Error', 'Central Error', 'Location', 'northwest');
```



Q. 3: (a) Write a MATLAB script to calculate partial derivative of f(x) = sin(x1)exp(-x2) at x1 = 0.5 and x2 = 1.

(b) Find the error using central difference method.

(c) Plot the error using log-log scale for the step size h = 1e06.

```
% Matlab script to calculate partial derivative of f(x) = sin(x1) * exp(-x2)
clc;
clear;
close all;
a = [0.5; 1];
h = [10e-6; 10e-6];
truVal = [cos(a(1))*exp(-a(2)); -sin(a(1))*exp(-a(2))];
% Central difference formula
% Partial differentiation w.r.t x
```

```
cntDiff(1) = (f3(a(1) + h(1), a(2)) - f3(a(1) - h(1), a(2)))./(2.*h(1));
% Partial differentiation w.r.t y
cntDiff(2) = (f3(a(1), a(2) + h(2)) - f3(a(1), a(2) - h(2)))./(2.*h(2));
errCnt = abs(truVal - cntDiff.');
disp('Error in cntDiff is:');
```

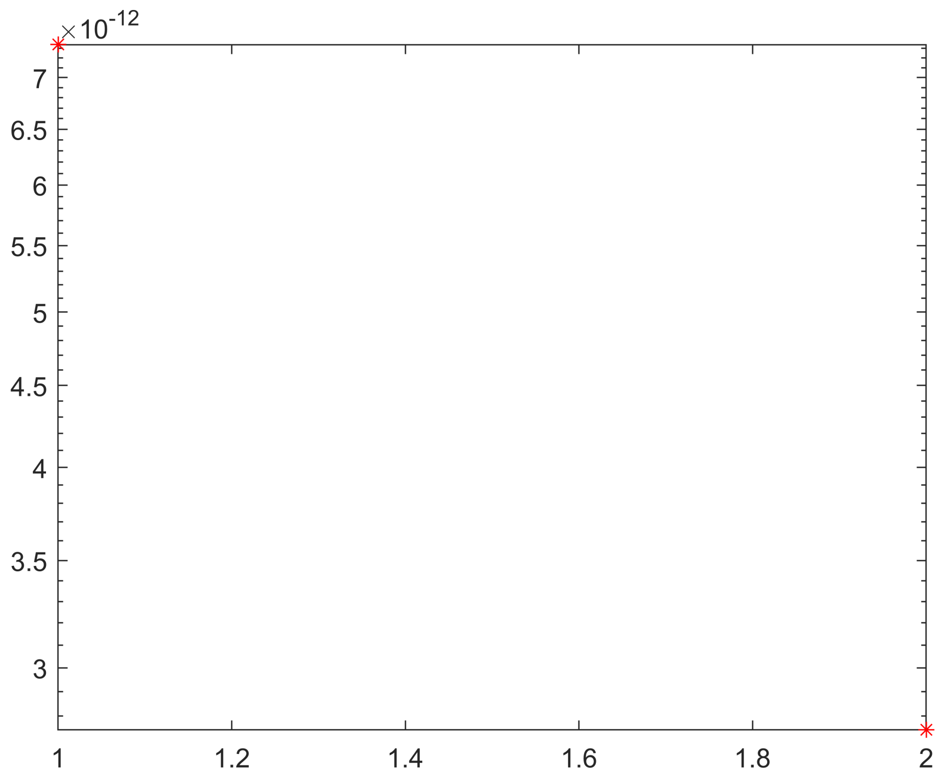
```
Error in cntDiff is:
```

```
disp(errCnt)
```

```
   1.0e-11 *

    0.7339
    0.2744
```

```
% The errCnt has two values: The first one is the differentiation w.r.t x
% and second is differentiation w.r.t y.
% Plot
semilogy(errCnt, 'r*')
```




```
% Function
```

```
function fx = f2(x)
  fx = 2 - x + log(x);
end
```

```
function fx = f(x)
  fx = atan(x);
end

function fx = f3(x1, x2)
  fx = sin(x1)*exp(-x2);
end
```