Maddox Gonzalez

Table of Contents

Due 1/26/2025

Project 1: Finite Difference and why you cannot

```
%& take the limit x approaches 0 on the computer

% Clear command window, close all graphs, clear workspace
clc, close all, clear

fprintf("==============================================================\n")
fprintf("Project 1 - Round-off vs Truncation Error\n")
fprintf("Insert Your Name Here\n")
display(date())
fprintf("==============================================================\n\n")

format short E
% Define the function f(x)
f = @(x) 0.35 + 2.25*cos(2.5*x) ; % Insert Function f(x)

df = @(x) -0.35*2.25*2.5*sin(2.5*x) ; % Insert Exact First Derivative df(x)
% Select x value to evaluate derivative
x = 0.75 ;
```

```
==============================================================
Project 1 - Round-off vs Truncation Error
Insert Your Name Here
19-Jan-2025
==============================================================
```

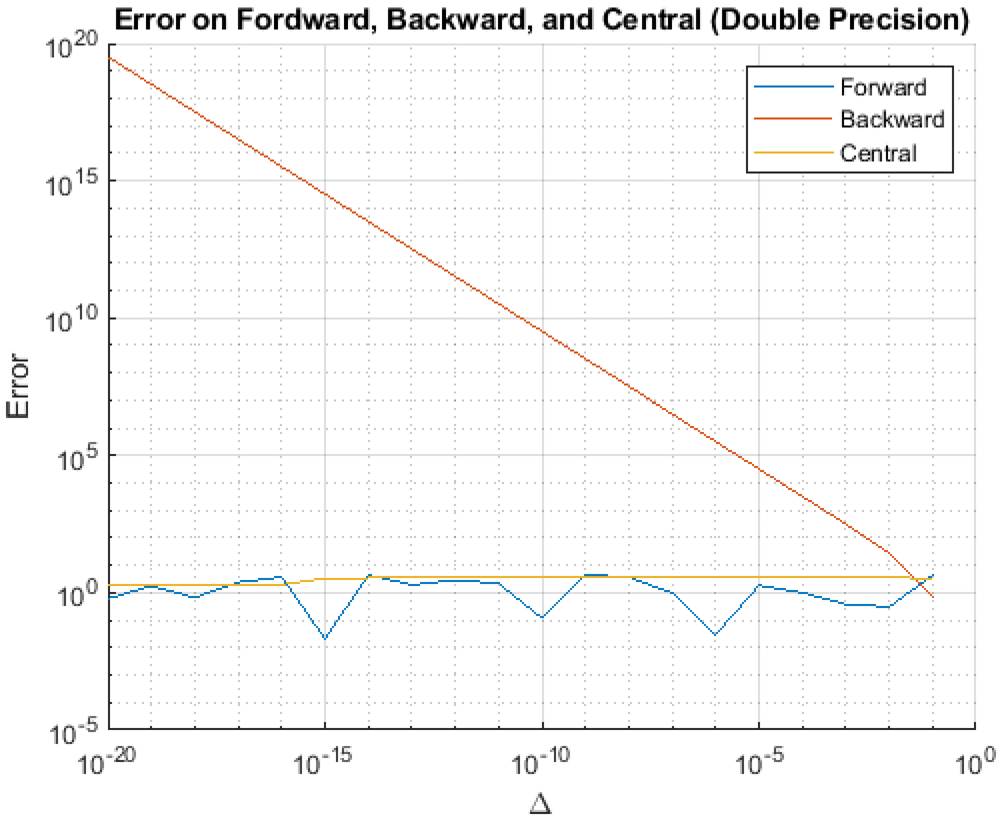
Double-Precision

loop through the exponents for each delta x value and evaluate the derivative

```
for i = 1:20
del(i,1) = 10^(-i);

% Backward Finite Difference
backward(i,1) = (f(x)-f(x-del(i,1))/del(i,1)) ;
```

```
% Forward Finite Difference
forward(i,1) = (f(x+del(i,1)+f(x)/del(i,1))) ;

% Central Finite Difference
central(i,1) = (f(x + del(i,1)) - f(x - del(i,1))) / (2 * del(i,1)) ;

% calculate the errors compared to exact (analytical) derivative
backward_error(i,1) = abs(backward(i,1) - df(x)) ;
forward_error(i,1) = abs(forward(i,1) - df(x)) ;
central_error(i,1) = abs(central(i,1) - df(x)) ;

end

% Plot the error vs del x for each method on the log scale
figure
hold on
% Scale of log for axis
set(gca, 'XScale', 'log', 'YScale', 'log')
% the loglog() function is the same as plot() but on log scale for both axes
loglog(del,forward_error)
loglog(del,backward_error)
loglog(del,central_error)
xlabel('\Delta'), ylabel('Error')
title('Error on Fordward, Backward, and Central (Double Precision)')
legend('Forward','Backward','Central')
grid on, hold off

% assemble the results into a table (actually a matrix)
Table = [del,forward_error,backward_error,central_error];
% print to screen

fprintf("=========== Double Precision Results ===========\n")
disp(Table)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
=========== Double Precision Results ===========
   1.0000e-01   4.3715e+00   7.2661e-01   3.4326e+00
   1.0000e-02   3.0757e-01   2.8562e+01   3.4878e+00
   1.0000e-03   3.8307e-01   3.2014e+02   3.4884e+00
   1.0000e-04   1.0131e+00   3.2357e+03   3.4884e+00
   1.0000e-05   1.8344e+00   3.2391e+04   3.4884e+00
   1.0000e-06   2.9895e-02   3.2395e+05   3.4884e+00
   1.0000e-07   9.7023e-01   3.2395e+06   3.4884e+00
   1.0000e-08   3.7672e+00   3.2395e+07   3.4884e+00
   1.0000e-09   4.4710e+00   3.2395e+08   3.4884e+00
   1.0000e-10   1.1921e-01   3.2395e+09   3.4884e+00
   1.0000e-11   2.2020e+00   3.2395e+10   3.4884e+00
   1.0000e-12   2.8644e+00   3.2395e+11   3.4884e+00
   1.0000e-13   1.9408e+00   3.2395e+12   3.4890e+00
   1.0000e-14   4.4646e+00   3.2395e+13   3.4618e+00
   1.0000e-15   2.0781e-02   3.2395e+14   3.3397e+00
   1.0000e-16   3.7590e+00   3.2395e+15   2.0074e+00
   1.0000e-17   2.4095e+00   3.2395e+16   1.8784e+00
   1.0000e-18   6.7016e-01   3.2395e+17   1.8784e+00
   1.0000e-19   1.7725e+00   3.2395e+18   1.8784e+00
   1.0000e-20   6.0790e-01   3.2395e+19   1.8784e+00
```



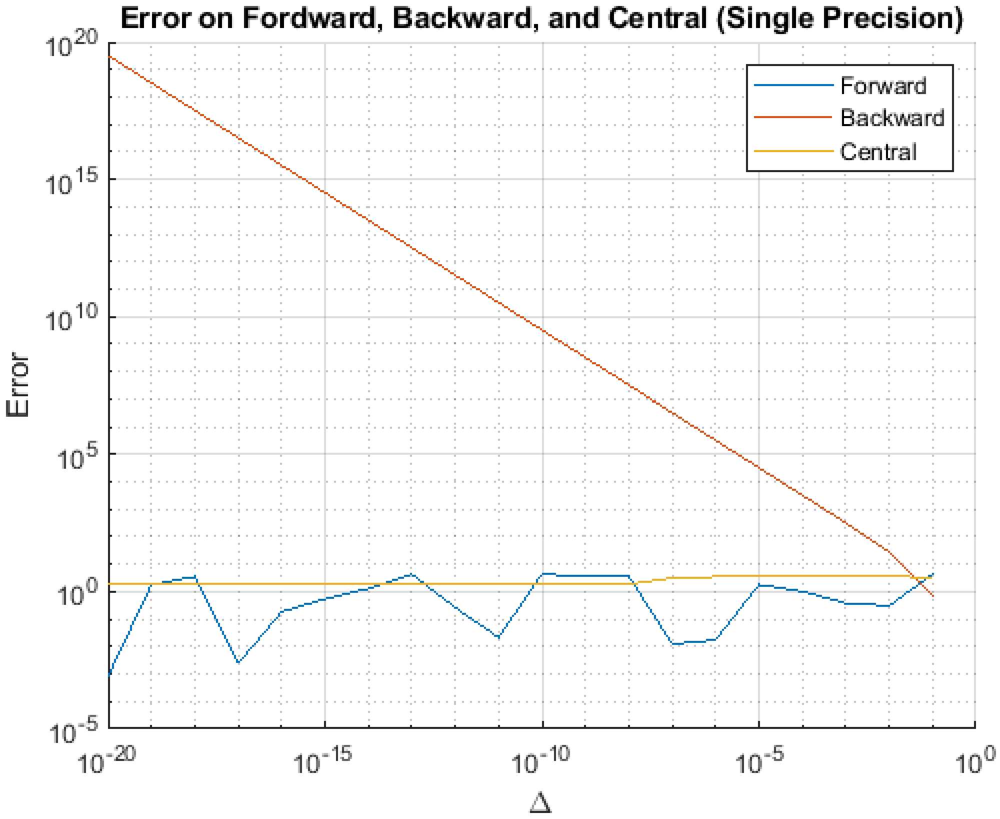
Repeat everything, but in single precision

Select x value to evaluate derivative

```
x = single(x);
for i = single(1:20)
del(i,1) = 10^(-i);

% Backward Finite Difference
backward(i,1) = (f(x)-f(x-del(i,1))/del(i,1)) ;

% Forward Finite Difference
forward(i,1) = (f(x+del(i,1)+f(x)/del(i,1))) ;

% Central Finite Difference
central(i,1) = (f(x + del(i,1)) - f(x - del(i,1))) / (2 * del(i,1)) ;

% calculate the errors compared to exact (analytical) derivative
backward_error(i,1) = abs(backward(i,1) - df(x)) ;
forward_error(i,1) = abs(forward(i,1) - df(x)) ;
```

```
central_error(i,1) = abs(central(i,1) - df(x)) ;

end

% Plot the error vs del x for each method on the log scale
figure
hold on
% Scale of log for axis
set(gca, 'XScale', 'log', 'YScale', 'log')
% the loglog() function is the same as plot() but on log scale for both axes
loglog(del,forward_error)
loglog(del,backward_error)
loglog(del,central_error)
xlabel('\Delta'), ylabel('Error')
title('Error on Fordward, Backward, and Central (Single Precision)')
legend('Forward','Backward','Central')
grid on, hold off

% assemble the results into a table (actually a matrix)
Table = [del,forward_error,backward_error,central_error];
% print to screen

fprintf("=========== Single Precision Results ===========\n")
disp(Table)
```

```
=========== Single Precision Results ===========
   1.0000e-01   4.3715e+00   7.2661e-01   3.4326e+00
   1.0000e-02   3.0759e-01   2.8562e+01   3.4878e+00
   1.0000e-03   3.8323e-01   3.2014e+02   3.4882e+00
   1.0000e-04   1.0155e+00   3.2357e+03   3.4902e+00
   1.0000e-05   1.8075e+00   3.2391e+04   3.4980e+00
   1.0000e-06   1.7354e-02   3.2395e+05   3.4861e+00
   1.0000e-07   1.2495e-02   3.2395e+06   3.1880e+00
   1.0000e-08   3.6364e+00   3.2395e+07   1.8784e+00
   1.0000e-09   3.4780e+00   3.2395e+08   1.8784e+00
   1.0000e-10   4.4077e+00   3.2395e+09   1.8784e+00
   1.0000e-11   2.1560e-02   3.2395e+10   1.8784e+00
   1.0000e-12   2.6267e-01   3.2395e+11   1.8784e+00
   1.0000e-13   4.3590e+00   3.2395e+12   1.8784e+00
   1.0000e-14   1.2901e+00   3.2395e+13   1.8784e+00
   1.0000e-15   5.4499e-01   3.2395e+14   1.8784e+00
   1.0000e-16   1.8562e-01   3.2395e+15   1.8784e+00
   1.0000e-17   2.4090e-03   3.2395e+16   1.8784e+00
   1.0000e-18   3.5497e+00   3.2395e+17   1.8784e+00
   1.0000e-19   1.7759e+00   3.2395e+18   1.8784e+00
   1.0000e-20   8.1635e-04   3.2395e+19   1.8784e+00
```



Published with MATLAB® R2024b