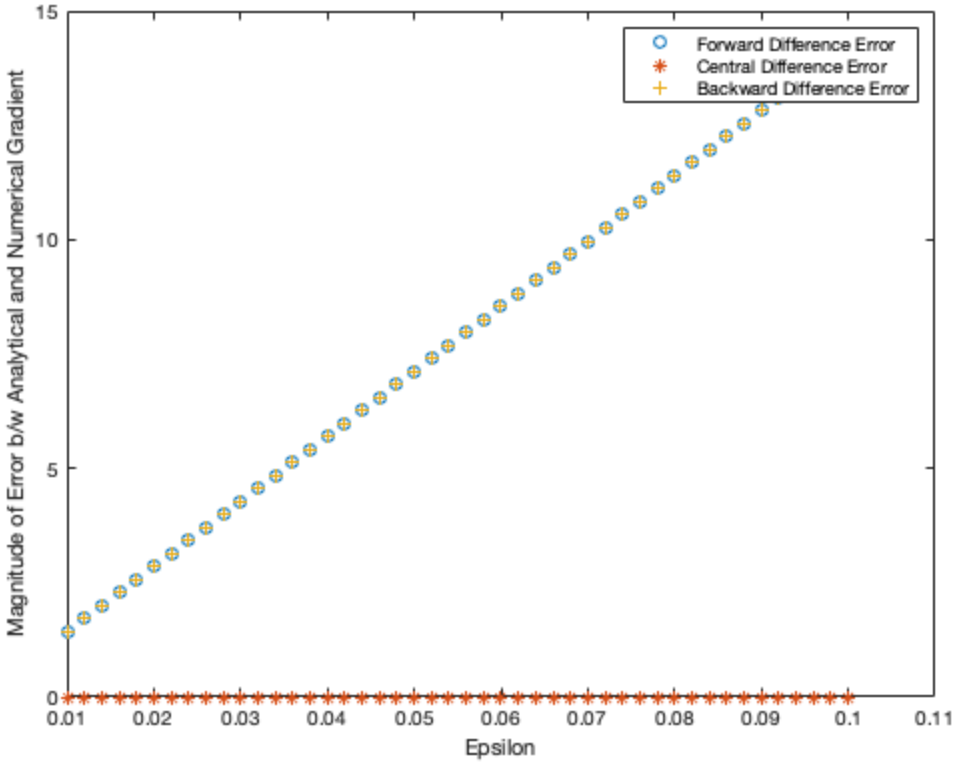
```
function [] = Gradient()
clear all;
close all;
disp('=======================================');
disp('     GRADIENT by Finite Differences Assignment Q1     ');
disp('     AML 771: Design Optimization and Decision Theory   ');
disp('     Student: Debjit Hore       ');
disp('=======================================');
% computes the gradient of a function supplied by user in #Subroutine
 Getfun#
% using forward, backward and central differences, and compares this
 with
% analytical gradient provided by the user in #Subroutine Gradient#

      N = 2;              % N: Dimension of the space
      epsilon = [0.01:0.002:0.1];        %
%     ===== READ POINT (Initial point)
      X = [5, 6]';
      [F0] = GETFUN (X);
% ----- GRADIENTS
      [G] = GRAD (X);
      for J=1:length(epsilon)
%----- FORWARD DIFFERENCE EVALUATION
      for I = 1:N
      A = X(I);
      DX = epsilon(J);      % DX: Purturbation of X
      if X(I) ~= 0.
        DX =  epsilon(J) * abs(X(I));
      end
      X(I) = X(I) + DX;
      [F] = GETFUN (X);
      X(I) = A;
      GF(J,I) = (F - F0) / DX;          % Gradient of F calculation
      end
% ----- BACKWARD DIFFERENCE EVALUATION
      for I = 1: N
      A = X(I);
      DX =  epsilon(J);
        if X(I) ~= 0.
          DX =  epsilon(J) * abs(X(I));
      end
      X(I) = X(I) - DX;
      [F] = GETFUN (X);
      X(I) = A;
      GB(J, I) = (F0 - F) / DX;
      end
% ----- CENTRAL DIFFERENCE EVALUATION
      for I = 1: N
        A = X(I);
      DX =  epsilon(J);
      if X(I) ~= 0.
        DX =  epsilon(J) * abs(X(I));
```

```
      end
      X(I) = X(I) - .5 * DX;              %purturbing back
      [F1] = GETFUN (X);
      X(I) = A;
      X(I) = X(I) + .5 * DX;              %purturbing forward
      [F] = GETFUN (X);
      X(I) = A;
      GC(J,I) = (F - F1) / DX;
      end
      end
     GF_Error= sqrt((GF(:,1)-G(1)).^2+(GF(:,2)-G(2)).^2);
     GB_Error= sqrt((GB(:,1)-G(1)).^2+(GB(:,2)-G(2)).^2);
     GC_Error= sqrt((GC(:,1)-G(1)).^2+(GC(:,2)-G(2)).^2);
     disp(' ')
     disp('  Analytical Gradient  Forw. Diff.  Backw. Diff.  Central
 Diff.')
     for I = 1: N
        disp(sprintf(' %15.4E %15.4E %15.4E %15.4E\n',G(I), GF(1,I),
 GB(1,I), GC(1,I)));
    end
     figure;
     plot(epsilon, GF_Error, 'o');
     hold on;
     plot(epsilon, GC_Error,  '*');
     plot(epsilon, GB_Error,  '+');
     hold off;
     ylabel('Magnitude of Error b/w Analytical and Numerical
 Gradient');
     xlabel('Epsilon');
     legend('Forward Difference Error', 'Central Difference
 Error', 'Backward Difference Error');

      function [F] = GETFUN(X)
% ===== DEFINE FUNCTION
      F = 12.069 * X(1)^2 + 21.504 * X(2)^2 - 1.7321 * X(1) - X(2);


      function [G] = GRAD(X)
% ===== GRADIENTS
      G(1) = 24.138 * X(1) - 1.7321;
      G(2) = 43.008 * X(2) - 1;

========================================
     GRADIENT by Finite Differences Assignment Q1
     AML 771: Design Optimization and Decision Theory
     Student: Debjit Hore
========================================

  Analytical Gradient  Forw. Diff.  Backw. Diff.  Central Diff.
      1.1896E+02      1.1956E+02      1.1835E+02      1.1896E+02

      2.5705E+02      2.5834E+02      2.5576E+02      2.5705E+02
```



Published with MATLAB® R2021a