1. Compute \( D_h(x_0) \) and \( \hat{D}_h(x_0) \) for some function \( f(x) \) at \( x_0 = 2 \). \( f(x) \in [-3, 3] \), \( f^{(1)}(x) \in [-4, 4] \), \( f^{(2)}(x) \in [-5, 5] \), and \( f^{(3)}(x) \in [-6, 6] \) for \( x \in [1, 3] \). Maximal values of \( h \) that guarantee error no greater than \( 10^{-4} \) for each method using Taylor polynomial based error bounds.

Start with first order Taylor polynomial expansion of \( f(x_0 + h) \) centered around \( x_0 \) with remainder term.

\[
f(x_0 + h) = f(x_0) + f^{(1)}(x_0)h + \frac{1}{2} f^{(2)}(\xi)h^2
\]

Now use this result in the expression for \( D_h(x_0) \).

\[
D_h(x_0) = \frac{f(x_0 + h) - f(x_0)}{h}
= f(x_0) + f^{(1)}(x_0)h + \frac{1}{2} f^{(2)}(\xi)h^2 - f(x_0)
= f^{(1)}(x_0) + \frac{1}{2} f^{(2)}(\xi)h
\]

Now express the error as a difference between the actual derivative and the approximation above.

\[
\left| f^{(1)}(x_0) - D_h(x_0) \right| = \left| f^{(1)}(x_0) - f^{(1)}(x_0) - \frac{1}{2} f^{(2)}(\xi)h \right|
= \left| -\frac{1}{2} f^{(2)}(\xi)h \right|
\]

Bounding the error in terms of the second derivative along interval \([x_0, x_0 + h]\).

\[
e_h \leq \max_{\xi \in [x_0, x_0 + h]} \left| \frac{1}{2} f^{(2)}(\xi)h \right| \approx \frac{1}{2} \left| f^{(2)}(x_0) \right| h
\] (1)

Now, observe the total interval of \( x \) considered for this problem. It is known that \( x_0 = 2 \) and that \( f^{(2)}(x_0) \in [-5, 5] \) for all \( x \in [1, 3] \). This means that \( |h| \leq 1 \) to keep \((x_0 + h) \in [1, 3]\). If \( |h| > 1 \), then \((x_0 + h) \notin [1, 3]\), and the given range for \( f^{(2)}(x_0) \) would no longer be valid.

Assuming \( |h| \leq 1 \), the maximum value of \( |f^{(2)}(x_0)| \) for \( x \in [1, 3] \) is just the absolute value of
either of the two extremes in the range $[-5, 5]$, because $|−5| = |5|$. Now, find the maximal
$h$ to satisfy an error no greater than $10^{-4}$.

$$\hat{e}_h \leq \left| \frac{5}{2}h \right| \quad \text{for } |h| \leq 1$$

$$\frac{5}{2}h \leq 10^{-4}$$

$$h \leq 4 \cdot 10^{-5}$$

$$h_{max} = 4 \cdot 10^{-5}$$

Now repeat the process but for $\hat{D}_h(x_0)$ this time. Taking a second order Taylor polynomial
with remainder term for both $f(x_0 + h)$ and $f(x_0 - h)$

$$f(x_0 + h) = f(x_0) + f^{(1)}(x_0)h + \frac{1}{2}f^{(2)}(x_0)h^2 + \frac{1}{6}f^{(3)}(\xi_1)h^3 \quad \text{for } \xi_1 \in [x_0, x_0 + h] \quad (2)$$

$$f(x_0 - h) = f(x_0) - f^{(1)}(x_0)h + \frac{1}{2}f^{(2)}(x_0)h^2 - \frac{1}{6}f^{(3)}(\xi_2)h^3 \quad \text{for } \xi_2 \in [x_0 - h, x_0] \quad (3)$$

Now subtract 3 from 2 for $\hat{D}_h(x_0)$.

$$\hat{D}_h(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{2f^{(1)}(x_0)h + \frac{1}{2}f^{(3)}(\xi_1)h^3 + \frac{1}{6}f^{(3)}(\xi_2)h^3}{2h}$$

$$= f^{(1)}(x_0) + \frac{1}{12}(f^{(3)}(\xi_1) + f^{(3)}(\xi_2))h^2$$

Again, from the problem statement $h$ is limited by the interval of $x \in [1, 3]$ that is considered.
With $|h| \leq 1$, it holds that $f^{(3)}(x_0 \pm h) \in [-6, 6]$ which then means that the maximum
will be one of the two extremes along the range of $f^{(3)}(x_0 \pm h)$ which will be $|−6| = |6| = 6$.

$$\left| f^{(1)}(x_0) - \hat{D}_h(x_0) \right| \leq \left| f^{(1)}(x_0) - f^{(1)}(x_0) - \frac{1}{12}(f^{(3)}(\xi_1) + f^{(3)}(\xi_2))h^2 \right|$$

$$= \left| -\frac{1}{12}(f^{(3)}(\xi_1) + f^{(3)}(\xi_2))h^2 \right|$$

$$\leq \frac{1}{12}f^{(3)}(\xi_1)h^2 + \frac{1}{12}f^{(3)}(\xi_2)h^2$$

$$\hat{e}_h \leq \frac{1}{12} \left[ \max_{\xi_1 \in [x_0, x_0 + h]} \left| f^{(3)}(\xi_1) \right| + \max_{\xi_2 \in [x_0 - h, x_0]} \left| f^{(3)}(\xi_2) \right| \right] h^2 \approx \frac{1}{6} \left| f^{(3)}(x_0) \right| h^2 \quad (4)$$
Now, find the maximal $h$.

\[
\hat{e}_h \leq \frac{6}{12} h^2 + \frac{6}{12} h^2 \quad \text{for } |h| \leq 1 \\
\leq |h^2| \\
|h^2| \leq 10^{-4} \\
h \leq 10^{-2} \\
\hat{h}_{\text{max}} = 10^{-2}
\]

2. With calculator, compute $D_h(x_0)$ and $\hat{D}_h(x_0)$ for $f(x) = e^{-1.5x}$ at $x_0 = 1$, with $h = 0.001$. Obtain error bounds with Taylor polynomials. What is the actual derivative and actual error in each approximation?

Start by evaluating $D_h(x_0)$ and $\hat{D}_h(x_0)$ directly with the given function and values of $x_0$ and $h$.

\[
f(x) = e^{-1.5x} \quad h = 0.001 \\
D_h(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} = \frac{e^{-1.5(1+0.001)} - e^{-1.5(1)}}{0.001} \\
\boxed{D_h(x_0) = -0.334444344} \\
\hat{D}_h(x_0) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} = \frac{e^{-1.5(1+0.001)} - e^{-1.5(1-0.001)}}{2(0.001)} \\
\boxed{\hat{D}_h(x_0) = -0.334695366}
\]

To obtain error bounds with each of the approximation methods, the first three derivatives must be taken.

\[
f(x) = e^{-1.5x} \\
f^{(1)}(x) = -1.5e^{-1.5x} \\
f^{(2)}(x) = 2.25e^{-1.5x} \\
f^{(3)}(x) = -3.375e^{-1.5x}
\]

Now, starting with $D_h(x_0)$, the expression for the error bound, equation 1, will still hold. Note that $|f(x)|$, $|f^{(1)}(x)|$, $|f^{(2)}(x)|$, and $|f^{(3)}(x)|$ are all decreasing in magnitude with increasing $x$, so choose $\xi^*$ to be the left most value.
\[ |f^{(1)}(x_0) - D_h(x_0)| \leq \max_{\xi \in [x_0, x_0 + h]} \left| \frac{1}{2} f^{(2)}(\xi) h \right| \]

Maximized for taking \( \xi^* = x_0 \)

\[ |f^{(1)}(x_0) - D_h(x_0)| \leq \frac{1}{2} f^{(2)}(x_0) \cdot 0.001 \]

\[ \leq \frac{1}{2} \cdot 2.25e^{-1.5} \cdot 0.001 \]

\[ \leq 2.5102 \cdot 10^{-4} \]

Also acceptable to use the alternate form of error bound in equation 1.

\[ |f^{(1)}(x_0) - D_h(x_0)| \leq \frac{1}{2} f^{(2)}(x_0) \cdot h \]

\[ \leq \frac{1}{2} \cdot 2.25e^{-1.5} \cdot 0.001 \]

\[ e \leq 2.5102 \cdot 10^{-4} \]

Moving on to \( \hat{D}_h(x_0) \), use the expression for bounding derived in the previous problem again. This time, choose \( \xi^*_1 \) to be left most in the interval \([x_0, x_0 + h]\) and \( \xi^*_2 \) to be left most in the interval \([x_0 - h, x_0]\), because \(|f^{(3)}(x)|\) is decreasing in magnitude with increasing \( x \).

\[ |f^{(1)}(x_0) - \hat{D}_h(x_0)| \leq \max_{\xi_1 \in [x_0, x_0 + h]} \left| \frac{1}{12} f^{(3)}(\xi_1) h^2 \right| + \max_{\xi_2 \in [x_0 - h, x_0]} \left| \frac{1}{12} f^{(3)}(\xi_2) h^2 \right| \]

Maximized for taking \( \xi^*_1 = x_0 \); \( \xi^*_2 = x_0 - h \)

\[ |f^{(1)}(x_0) - \hat{D}_h(x_0)| \leq \frac{1}{12} f^{(3)}(1) \cdot 0.001^2 \]

\[ + \frac{1}{12} f^{(3)}(1 - 0.001) \cdot 0.001^2 \]

\[ \leq \frac{1}{12} \cdot (-3.375e^{-1.5}) \cdot 0.001^2 \]

\[ + \frac{1}{12} \cdot (-3.375e^{-1.5}(1-0.001)) \cdot 0.001^2 \]

\[ \leq \frac{1}{12} \cdot 0.001^2 \cdot 3.375(e^{-1.5} + e^{-1.5-0.999}) \]

\[ \hat{e} \leq 1.2560 \cdot 10^{-7} \]

Also acceptable to use the alternate form of error bound in equation 4.

\[ |f^{(1)}(x_0) - \hat{D}_h(x_0)| \leq \frac{1}{6} f^{(3)}(x_0) \cdot h^2 \]

\[ \leq \frac{1}{6} \cdot (-3.375e^{-1.5}) \cdot 0.001^2 \]

\[ \hat{e} \leq 1.2551 \cdot 10^{-7} \]
Finally for evaluating the actual derivative and the actual errors, it will simply be an computation of $f^{(1)}(x_0)$ at $x_0 = 1$ then taking the difference of each of the two approximations.

\[
f^{(1)}(x_0) = -1.5e^{-1.5}
\]

\[
f^{(1)}(x_0) = -0.334695240
\]

\[
e = \left| f^{(1)}(x_0) - D_h(x_0) \right|
= \left| -0.334695240 + 0.334444344 \right|
\]

\[
e = 2.5090 \cdot 10^{-4}
\]

\[
\hat{e} = \left| f^{(1)}(x_0) - \hat{D}_h(x_0) \right|
= \left| -0.334695240 + 0.334695366 \right|
\]

\[
\hat{e} = 1.2551 \cdot 10^{-7}
\]

Note that the actual errors using both methods are less than or equal to their respective error bounds as expected.

3. With same $f(x)$, $x_0$, and $h$ as previous problem, assume machine carries only 9 digits. Within $\pm 1$, estimate the number of digits which would be correct in approximation of $D_h(x_0)$ based on error in catastrophic subtraction. With aid of calculator, compute $D_h(x_0)$ keeping 9 digits at each step and find actual error.

As before, evaluate $D_h(x_0)$ but carrying 9 digits at each calculation step.

\[
D_h(x_0) = \frac{f(x_0 + h) - f(x_0)}{h}
= \frac{e^{-1.5(1+0.001)} - e^{-1.5(1)}}{0.001}
= \frac{0.222795716 - 0.223130160}{0.001}
= \frac{-0.000334444}{0.001}
= -0.334444
\]

There were 3 digits of precision lost in this calculation, so the best number of digits to hope to be correct is 6 due to catastrophic subtraction. Now for the error.

\[
\left| f^{(1)}(x_0) - D_h(x_0) \right| = \left| -0.334695240 + 0.334444 \right|
= \left| -0.000251240 \right|
\]

\[
e = 0.00025140
\]
Note that this error is not contained within the error bound of $2.5102 \cdot 10^{-4}$ computed in the previous problem. This is due to catastrophic subtraction adding to the accumulated error.

4. MATLAB computation of $\hat{D}_h(x_0)$ for some given $f(x)$.
   Using the MATLAB Code:

```matlab
%%
% Problem 4
% Language: MATLAB
%
% Objective: This script will obtain a first derivative approximation
% for a defined function that is called out during the
% run execution. The derivative approximation will be done by
% a central difference method. The step size will vary by a
% factor of $10^{-1}$ per iteration to see how the approximation
% differs with decreasing step size, $h$.
%
% Input Variables:
% None
%
% Output Variables:
% None
%
% Local Functions:
% noidea(x): Function Defined in Problem Statement
% f_Dh_hat(x,h,f): Central Difference Method Formula
%
% Define Input Constants/Vectors:
x0 = 1;
j = [1, 2, 3, 4];
n = length(j); %Reference for length of total j vector
h = 10.^(-j); %Define h with element-wise exponentiation operation over j
%
% Define Function Handle
handle = @noidea;
%
% Create Loop
% We want to loop FOR EACH value that exists in our j vector, but MATLAB
% has static FOR loop logic of {start:increment:end}, so we will define our
% logic manually as {1:1:n}
for count = 1:1:n
    % Now we can call out the "noidea(x)" function in our loop to
    % approximate the derivative by our Dh_hat definition
    Dh_hat = f_Dh_hat(x0,h(count),handle);
    % We can print our results as the loop executes to look at the console
    fprintf(’Dh_hat for h = %.9f’,h(count));
    fprintf(’\n’,Dh_hat);
end
%
% "No Idea" Function Definition
% Inputs:
% x: Point of evaluation
% Outputs:
```
% f: Evaluation of function

function f = noidea(x)
c1 = 5;
for i1=1:3
c1 = log(sin(c1)+3+x);
end
f = c1;
end

% %
% % Finite Difference Formulas
% %
% % Central Difference
% Inputs:
% x: Point of evaluation
% h: Step size
% f: Function handle for evaluation
% Outputs:
% D: Approximated value of derivative
% %
function D = f_Dh_hat(x, h, f)
    D = (f(x+h) - f(x-h))/(2*h);
end

The console output is:
Dh_hat for h = 0.100000000 is 0.199231680
Dh_hat for h = 0.010000000 is 0.199175859
Dh_hat for h = 0.001000000 is 0.199175301
Dh_hat for h = 0.000100000 is 0.199175295
Therefore, the results are:

<table>
<thead>
<tr>
<th>h</th>
<th>(\hat{D}_h(x_0))</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^{-1}</td>
<td>0.199231680</td>
</tr>
<tr>
<td>10^{-2}</td>
<td>0.199175859</td>
</tr>
<tr>
<td>10^{-3}</td>
<td>0.199175301</td>
</tr>
<tr>
<td>10^{-4}</td>
<td>0.199175295</td>
</tr>
</tbody>
</table>

5. With \(f(x) = e^{-1.5x}\) and \(x_0 = 1\), compute \(D_h(x_0)\), \(\hat{D}_h(x_0)\), and \(\tilde{D}_h(x_0)\) for \(h = 10^{-j}\), for \(j \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}\). Plot \(\log_{10}[\varepsilon_h(x_0)]\), \(\log_{10}[\hat{\varepsilon}_h(x_0)]\), and \(\log_{10}[\tilde{\varepsilon}_h(x_0)]\) versus \(\log_{10}[h]\). Indicate where Taylor polynomial error dominate and where catastrophic subtraction errors dominate. Write name centered vertically and touching the right hand edge of plot window. What do linear sections of slopes indicate?

The following code will plot the results:

1  %
2  % Problem 5
3  % Language: MATLAB
4  %
5  % Objective: This script will compare the error magnitudes obtained by
6  % different finite difference methods as the step size decreases. Then, a plot will be obtained which illustrates
7  % where the polynomial error dominates and where the

catastrophic subtraction error dominates.

% Input Variables:
% None

% Output Variables:
% None

% Local Functions:
% myfunction5(x): Evaluates the function at specified point for the function in problem statement 5
% myfunction5_prime(x): Evaluates the function derivative at specified point for function in problem statement 5
% f_Dh(x,h,f): Evaluates Forward Difference approximation of function 'f' at point 'x' with step size 'h'
% f_Dh_hat(x,h,f): Evaluates Central Difference approximation of function 'f' at point 'x' with step size 'h'
% f_Dh_tilde(x,h,f): Evaluates Central Difference approximation (Fourth Order) of function 'f' at point 'x' with step size 'h'

% Define Input Constants/Vectors:
x0 = 1;
j = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
n = length(j); %Reference for length of total j vector
h = 10.^(−j); %Define h with element-wise exponentiation operation over j

% Pre-Allocate Vectors for Loop
Dh = zeros(1,n);
Dh_hat = zeros(1,n);
Dh_tilde = zeros(1,n);

% Evaluate First Derivative at x0
df = myfunction5_prime(x0);

% Create Function Handle for Loop
handle = @myfunction5;

% Create Loop
for count = 1:1:n
    % Evaluate each approximation method and save to each vector
    Dh(count) = f_Dh(x0,h(count),handle);
    Dh_hat(count) = f_Dh_hat(x0,h(count),handle);
    Dh_tilde(count) = f_Dh_tilde(x0,h(count),handle);
end

% Evaluate the error of each approximation
eh = abs(df−Dh);
eh_hat = abs(df−Dh_hat);
eh_tilde = abs(df−Dh_tilde);

% Plot the Results
% Open new Figure
```matlab
figure;

% Begin Plotting and Formatting all Results in Log10 Scale
plot(log10(h), log10(eh), 'r');
hold on;
plot(log10(h), log10(eh_hat), 'b');
hold on;
plot(log10(h), log10(eh_tilde), 'g');

% Draw Arrows to Indicate Regions of Dominance
quiver([-2 -2 -2], [-2 -4 -8], [-10 -6 -3], [-10 -12 -10], 'k--');
quiver(-5, -13, -5.5, 'm--');

% Format Plot
I = legend({'$e_h$', '$\hat{e}_h$', '$\tilde{e}_h$'},...
    'Polynomial Error Dominant', 'Catastrophic Subtraction Dominant'});
set(I, 'Interpreter', 'Latex', 'Location', 'NorthWest', 'FontSize', 14);
xlabel({'$\log_{10}(h)$'}, 'Interpreter', 'latex');
ylabel({'$\log_{10}(e)$'}, 'Interpreter', 'latex');
title('Error of Finite Difference Methods for Varying Step Size');

% Problem 5 Function Definition
% Inputs:
% x: Point of Evaluation
% Outputs:
% f: Evaluation of Function
function f = myfunction5(x)
    % Evaluate the Function
    f = exp(-1.5*x);
end

% Problem 5 Function Derivative Definition
% Inputs:
% x: Point of Evaluation
% Outputs:
% df: Evaluation of Function Derivative
function df = myfunction5_prime(x)
    % Evaluate the Derivative
    df = -1.5*exp(-1.5*x);
end

% Finite Difference Formulas
% Forward Difference
% Inputs:
% x: Point of evaluation
% h: Step size
% f: Function handle for evaluation
% Outputs:
% D: Approximated value of derivative

function D = f_Dh(x, h, f)
    D = (f(x+h)-f(x))/h;
end
```
% Central Difference
% Inputs:
% x: Point of evaluation
% h: Step size
% f: Function handle for evaluation
% Outputs:
% D: Approximated value of derivative

function D = fDh_hat(x, h, f)
    D = (f(x+h) - f(x-h)) / (2*h);
end

% Central Difference (Fourth Order)
% Inputs:
% x: Point of evaluation
% h: Step size
% f: Function handle for evaluation
% Outputs:
% D: Approximated value of derivative

function D = fDh_tilde(x, h, f)
    D = (8*(f(x+h) - f(x-h)) - (f(x+2*h) - f(x-2*h))) / (12*h);
end

The resulting figure 1 will display associated errors with each method for a varying step size $h$. Note that the black dashed lines show the linear slope region corresponding to the sections where the Taylor polynomial associated error dominates (with slope equal to the
order of the method of approximation). The magenta dashed line shows what happens when catastrophic subtraction dominates the error. The error no longer decreases with decreasing $h$ but rather begins to increase with decreasing $h$ due to catastrophic subtraction induced error.

6. With a difference approximation of $f^{(2)}(x_0)$ defined by:

$$\tilde{D}_h(x_0) = \frac{f(x_0 + h) + f(x_0 - h) - 2f(x_0)}{h^2}$$

Compute $\tilde{D}_h(0)$ for $f(x) = \ln(x^2 + 2)$ in order to approximate $f^2(0)$ for $h = 10^{-j}$, for $j \in \{0, 1, 2, 3, 4, 5\}$. Finding $\hat{e}_h$ and plotting $\log_{10}(\hat{e}_h)$ versus $\log_{10}(h)$, comment on the result and specifically what the order might be from the plot.

The following code will plot the results:

```matlab
% Problem 6
% Language : MATLAB
% Objective : This script will utilize a given second derivative approximation method in order to obtain the value of said approximation for a particular function at a specified point with varying step size. The approximation is then compared to the actual value of the second derivative to calculate the actual error. The logarithmic plot of the error versus step size is then created.

% Inputs : None
% Outputs : None

% Local Functions:
% myfunction6(x) : Evaluates the function at specified point for the function in problem statement 6
% myfunction6_double_prime(x) : Evaluates the function second derivative at specified point for function in problem statement 6

%Define Constants/Vectors
x0 = 0;
j = [0, 1, 2, 3, 4, 5];
n = length(j); %Reference for length of total j vector
h = 10.^(−j); %Define h with element-wise exponentiation operation over j

% Pre-allocate vectors for each approximation
Dh_breve = zeros(1,n);

% Evaluate Function and Derivative at x0
d2f = myfunction6_double_prime(x0);
```
% Create Function Handle for Easier Callouts in Loop
handle = @myfunction6;
%
% Create Loop
% We want to loop FOR EACH value that exists in our j vector, but MATLAB
% has static FOR loop logic of {start:increment:end}, so we will define our
% logic manually as {1:1:n}
for count = 1:n
    % Evaluate approximation method and save to vector
    Dh_breve(count) = f_Dh_breve(x0,h(count),handle);
end
%
% Evaluate the error of approximation
% Plot the Results
% Open new Figure
figure;
% Begin Plotting and Formatting all Results in Log10 Scale
plot(log10(h),log10(eh_breve),'r');
% Format Plot
I = legend({'$\breve{e}_h$'},'Location','NorthWest','FontSize',14);
set(I,'Interpreter','latex','Location','NorthWest','FontSize',14);
xlabel({'$\log_{10}(h)$'},'Interpreter','latex');
ylabel({'$\log_{10}(\breve{e}_h)$'},'Interpreter','latex');
title('Error of Discrete Laplacian Method for Varying Step Size');
%
% Problem 6 Function Definition
% Inputs:
% x: Point of Evaluation
% Outputs:
% f: Evaluation of Function
function f = myfunction6(x)
% Evaluate the Function
f = log(x^2+2);
end
%
% Problem 6 Function Second Derivative Definition
% Inputs:
% x: Point of Evaluation
% Outputs:
% d2f: Evaluation of Function Second Derivative
function d2f = myfunction6_double_prime(x)
% Evaluate the Second Derivative
d2f = (-2*x^2+4)/(x^2+2)^2;
end
%
% Laplacian Method (1-Dimensional)
% Inputs:
% x: Point of Evaluation
% h: Step size
% f: Function handle for evaluation
% Outputs:
The resulting plot of figure 2 illustrates the relation between the error and the step size $h$. The linear part of this plot has a slope that is roughly 2. Note that a logarithmic scale is used, so this corresponds to the order of the method used, i.e. $\hat{e}_h = \mathcal{O}(h^2)$. This makes sense, because when utilizing logarithmic scale, $\log_{10}(\hat{e}_h) = \log_{10}(\mathcal{O}(h^2))$. Then, the degree term is moved out to yield an approximate slope, $\log_{10}(\hat{e}_h) = 2\log_{10}(\mathcal{O}(h))$. Note that catastrophic subtraction begins to dominate the error with a step size of $h = 10^{-4}$ or smaller.