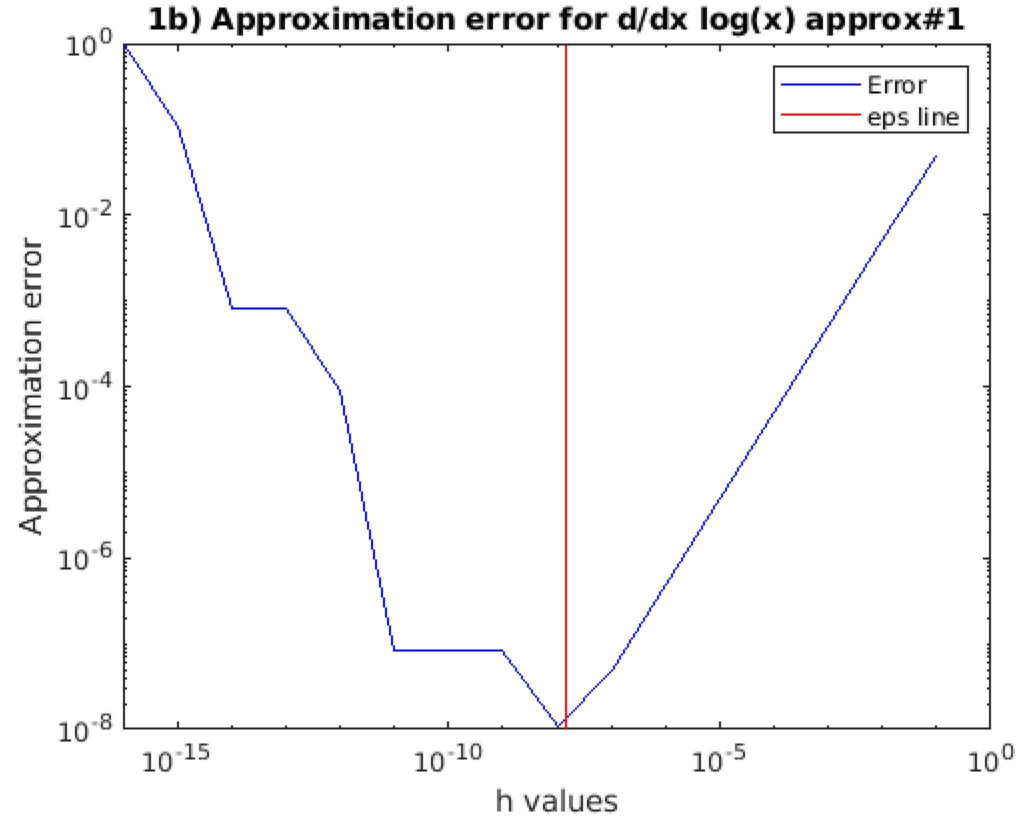
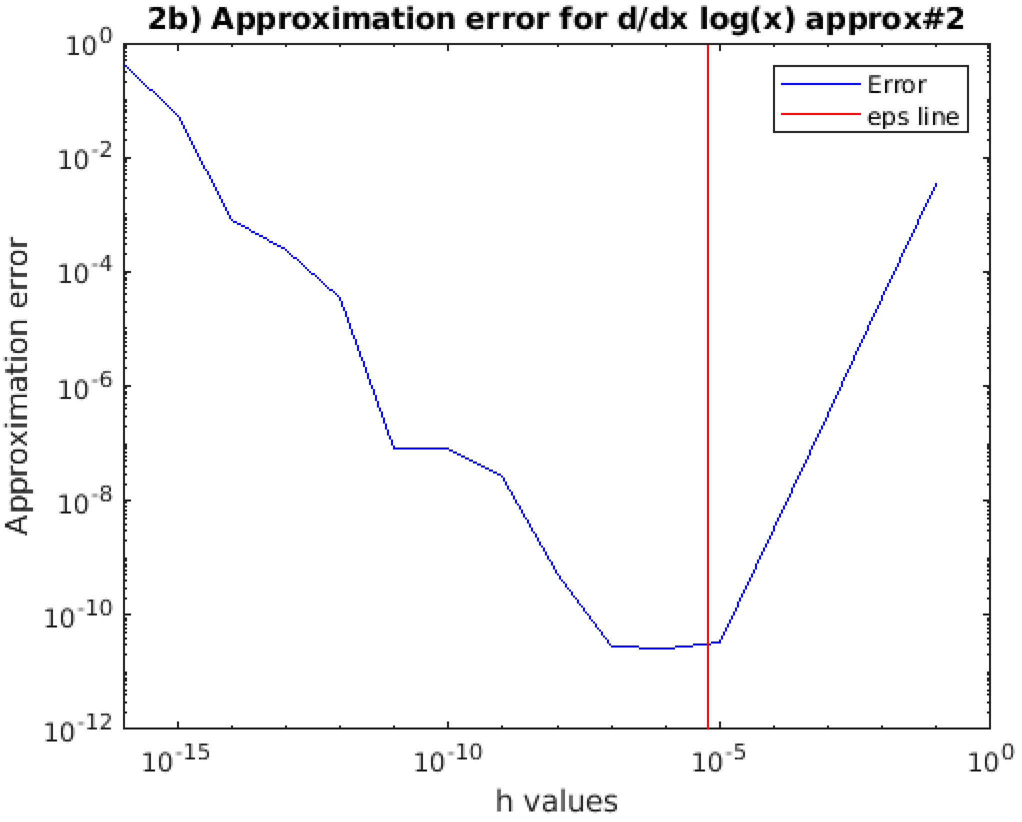
```matlab
% Anthony Galczak - agalczak@unm.edu
% CS375 - HW2

clc
close all

% 1b)
h = logspace(-1, -16, 16);
a = 1;

% Since f(x) = log(x), f'(x) = 1/x, f'(a) = 1/1 = 1
error_list = approx_1(@log, a, h, 1);
title1 = "1b) Approximation error for d/dx log(x) approx#1";
plot_error(error_list, h, title1, sqrt(eps));


% 1b) Explanation
% I used loglog to plot this as h is in logspace and loglog shows the
% error the best.
% My observation on plotting the approximate derivative for a=1 is
% that the error is the smallest at 10e-8. The error is much larger
% at the extremes (10e-1, 10e-15).

% 2b)
error_list = approx_2(@log, a, h, 1);
title2 = "2b) Approximation error for d/dx log(x) approx#2";
plot_error(error_list, h, title2, eps^(1/3));

% 2b) Explanation
% The error is similar to approximation 1, except that the error
% is closer to 1e-5 instead of 1e-8

% 3) Looks like we need 4 more plots to analyze both rounding
% error and truncation error.

% f(x) = x, a = 1, approx#1
% The truncation error for approx#1 is f''(eps), but the 2nd
% derivative of x is always 0. Thus, we are only dealing with
% rounding error here. No truncation error.
ident = @(x) x;
error_list = approx_1(ident, a, h, 1);
title3 = "3i.Approximation error for d/dx x approx#1";
% Since f(x) = x, f'(x) = 1, f'(a) = 1
plot_error(error_list, h, title3, sqrt(eps));

% f(x) = x, a = 1, approx#2
% The truncation error for approx#2 is f'''(eps), but the 2nd
% derivative of x is always 0. So again, we are only dealing
% with rounding error in this plot.
error_list = approx_2(ident, a, h, 1);
title4 = "3ii.Approximation error for d/dx x approx#2";
plot_error(error_list, h, title4, eps^(1/3));
```

```
% f(x) = x^2, a = 1, approx#1
% Since the truncation error in approx#1 is f''(eps),
% f(x)=x^2 will still have a derivative at a = 1.
% f''(1) = 2 so this plot still has rounding error.
square = @(x) x.^2;
% Since f(x) = x^2, f'(x) = 2*x, f'(a) = 2 * 1 = 2
error_list = approx_1(square, a, h, 2);
title5 = "3iii.Approximation error for d/dx x^2 approx#1";
plot_error(error_list, h, title5, sqrt(eps));

% f(x) = x^2, a = 1, approx#2
% Since truncation error in approx#2 is f'''(eps), we have
% eliminated the truncation error as d/dx(2) = 0. So again
% we are just dealing with rounding error.
error_list = approx_2(square, a, h, 2);
title6 = "3iv.Approximation error for d/dx x^2 approx#2";
plot_error(error_list, h, title6, eps^(1/3));

% 4a)
four_a_vals = linspace(1e-15,1e0,50);
for i=1:50
    x = four_a_vals(i);
    four_a_original(i) = (((exp(x)) - 1) / x) - 1;
    four_a_fixed(i) = x/factorial(2) + x^2/factorial(3) +...
        x^3/factorial(4) + x^4/factorial(5) +...
        x^5/factorial(6) + x^6/factorial(7);
end

figure
loglog(four_a_vals, four_a_original);
hold on
loglog(four_a_vals, four_a_fixed);
xlabel("Log scale values");
ylabel("Bad fn. vs. fixed fn.");
title("4a) Catastrophic cancellation ex.");
legend('Original', 'Fixed','Location','Southeast');
hold off

% 4b)
four_b_vals = linspace(1,100,100);
for i=1:100
    x = four_b_vals(i);
    four_b_original(i) = sqrt(1 + x^8) - x^4;
    four_b_fixed(i) = 1/(sqrt(1 + x^8) + x^4);
end

figure
loglog(four_b_vals, four_b_original);
hold on
loglog(four_b_vals, four_b_fixed);
xlabel("Values 1-100");
ylabel("Bad fn. vs. fixed fn.");
xlim([1 100]);
```

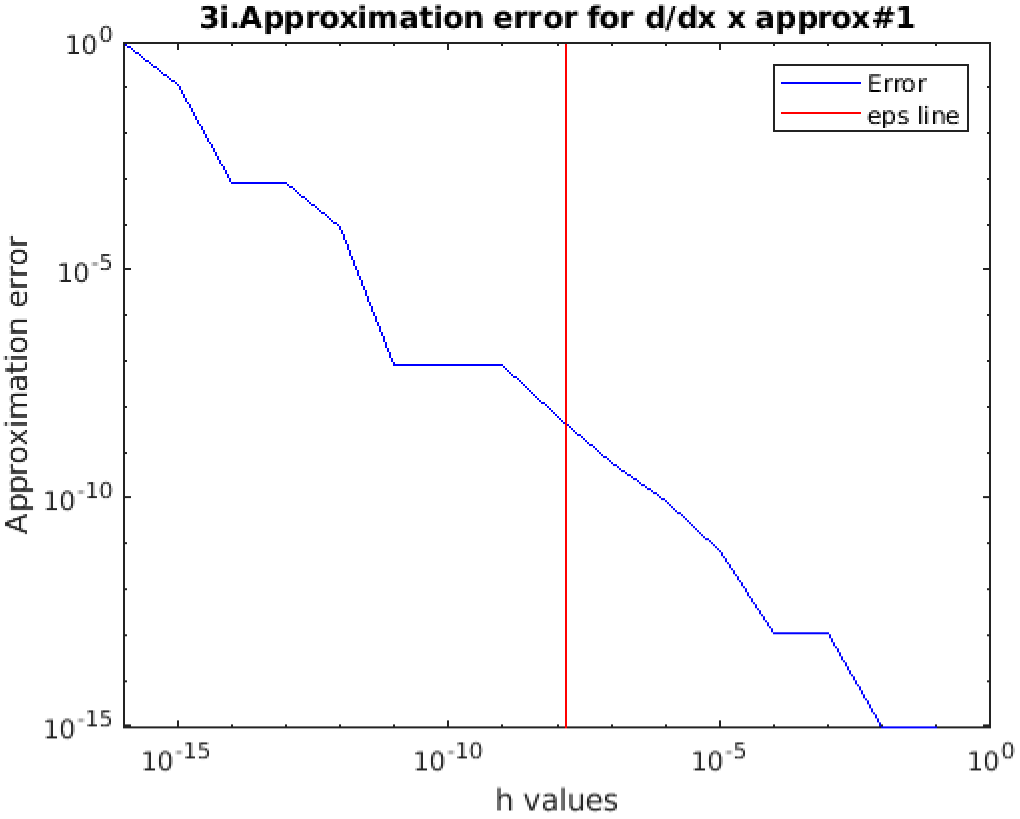
```
title("4b) Loss of Significance ex.");
legend('Original','Fixed');
hold off

% First approximation function, for 1b
function e = approx_1(fun, a, h, real_deriv)
    for i=1:16
        approx_list(i) = (fun(a + h(i)) - fun(a)) / h(i);
        error_list(i) = abs(real_deriv - approx_list(i));
    end
    e = error_list;
end

% Second approximation function, for 2b
function e = approx_2(fun, a, h, real_deriv)
    for i=1:16
        approx_list(i) = (fun(a + h(i)) - fun(a - h(i))) / (2*h(i));
        error_list(i) = abs(real_deriv - approx_list(i));
    end
    e = error_list;
end

% Function serves for plotting approximation errors
function p = plot_error(errors, h, title_str, eps_error)
    figure
    error_plot = loglog(h, errors,'Color','Blue');
    hold  on
    xlabel("h values");
    ylabel("Approximation error");
    title(title_str);
    eps_line = line([eps_error eps_error],[get(gca,'ylim')],...
        'Color','Red');
    legend([error_plot eps_line], ["Error", "eps line"], ...
       'Location','northeast');
    hold off
end
```

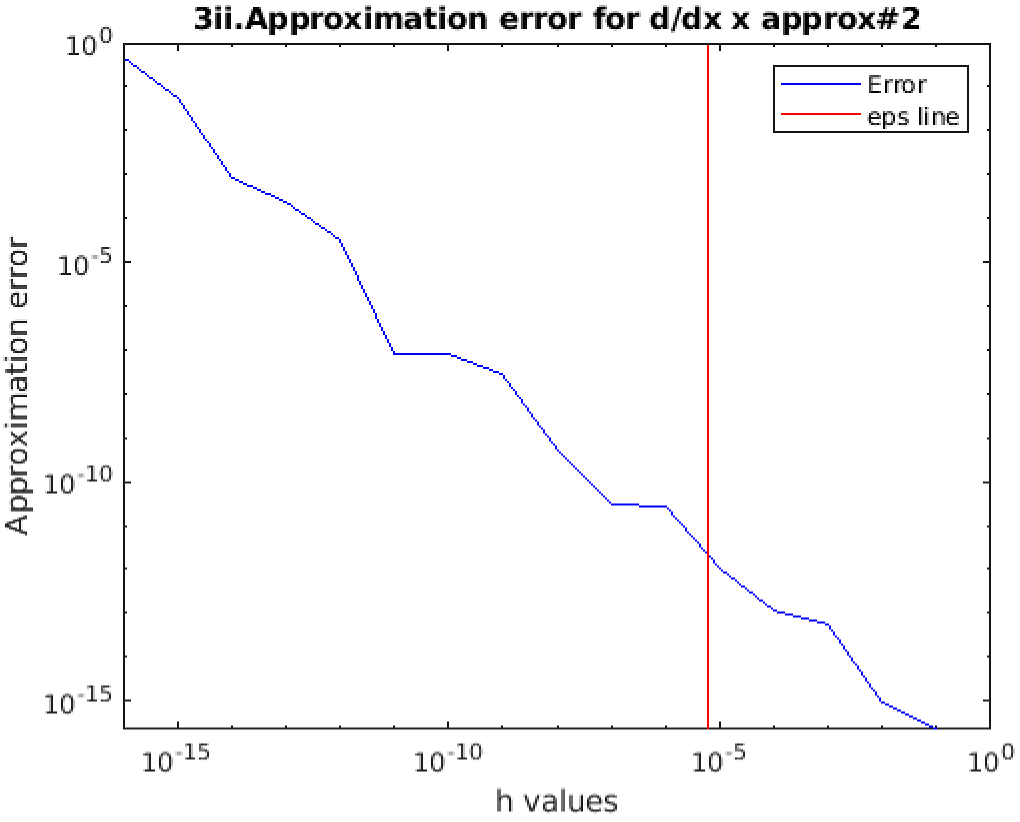


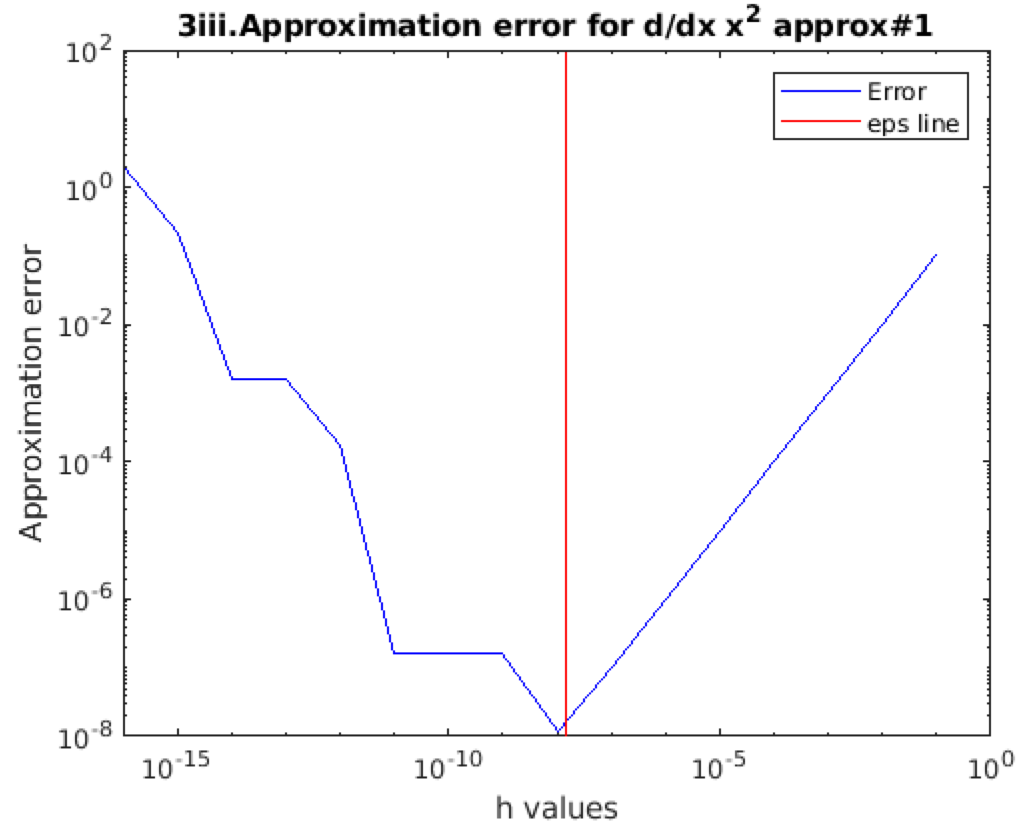
1b) Approximation error for d/dx log(x) approx#1

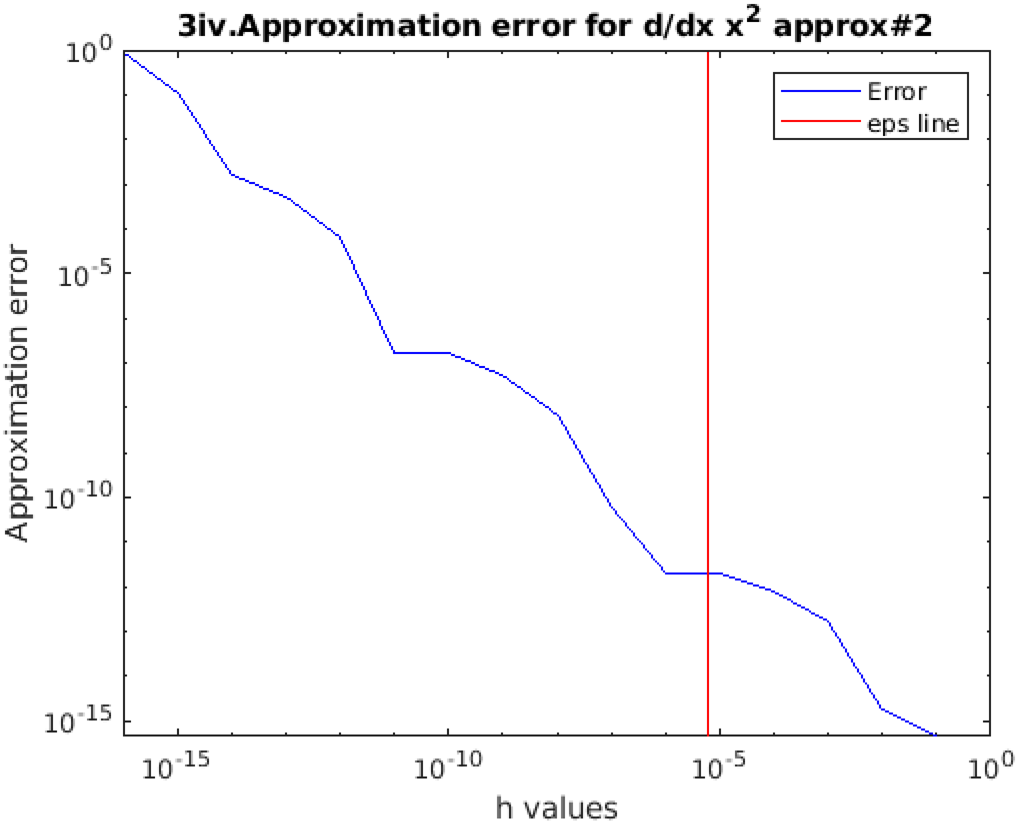


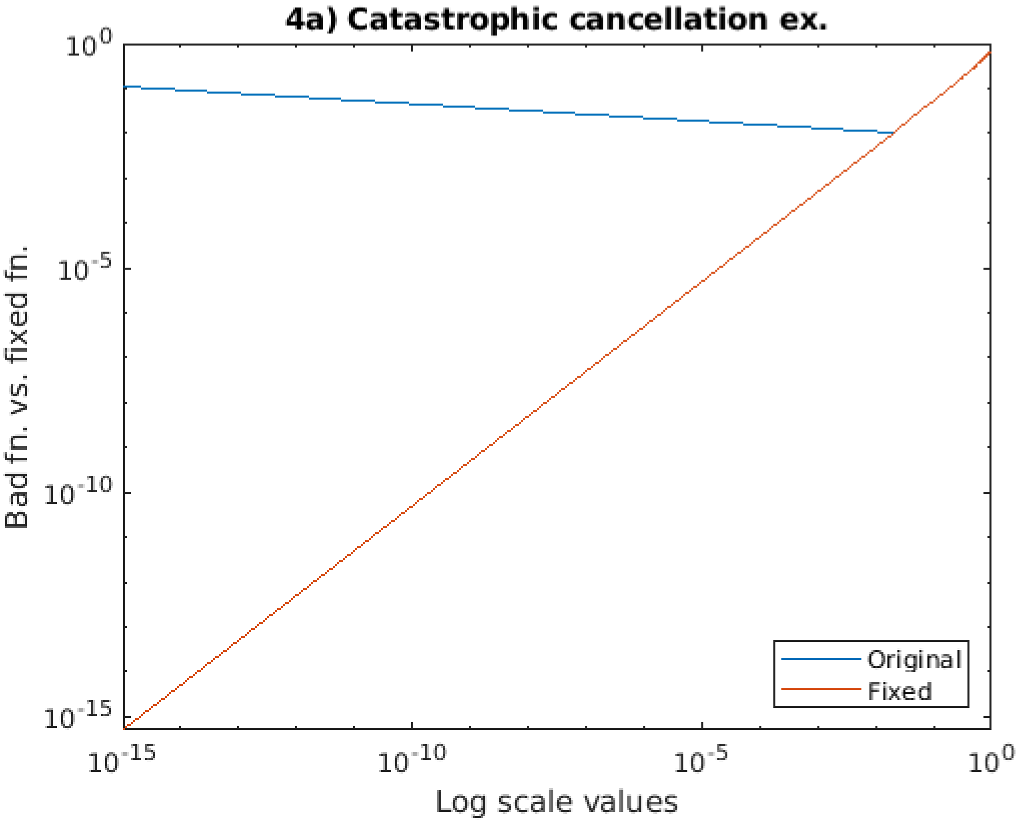
2b) Approximation error for d/dx log(x) approx#2

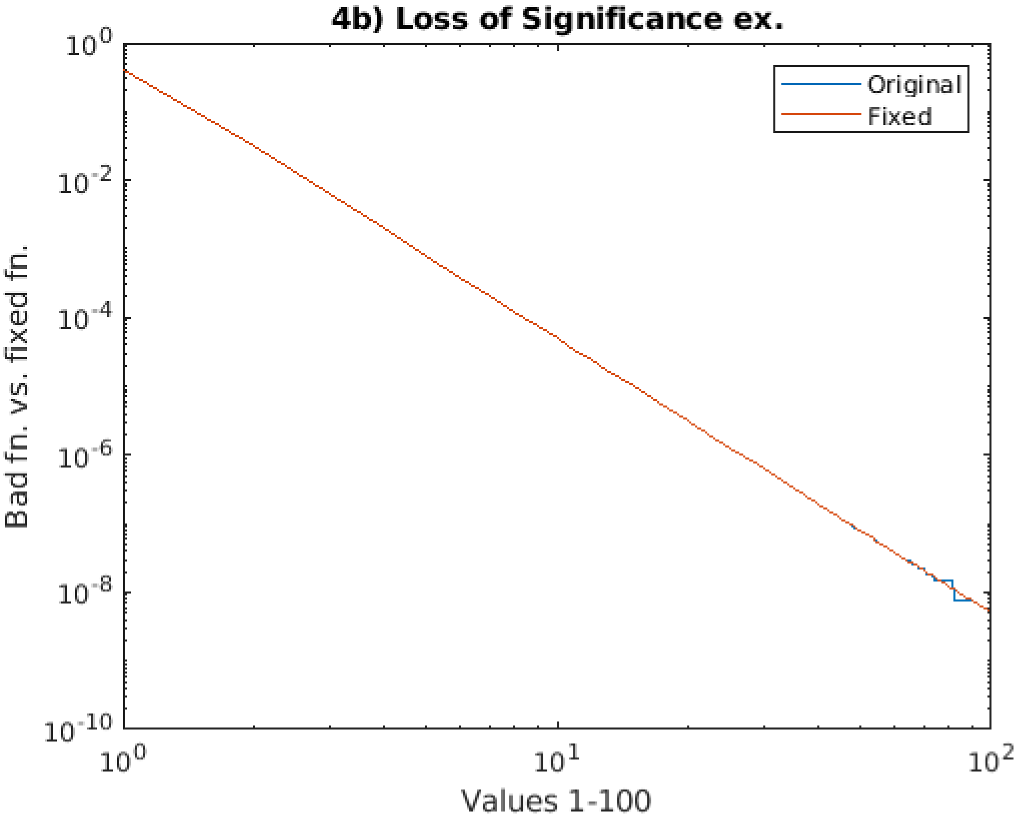

3i.Approximation error for d/dx x approx#1
Approximation error
h values
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-15}
10^{-10}
10^{-5}
10^{0}
Error
eps line


3ii.Approximation error for d/dx x approx#2
Approximation error
h values
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-15}
10^{-10}
10^{-5}
10^{0}
Error
eps line


3iii.Approximation error for d/dx x^2 approx#1
Approximation error
h values
10^{2}
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-15}
10^{-10}
10^{-5}
10^{0}
Error
eps line


3iv.Approximation error for d/dx x^2 approx#2
Approximation error
h values
10^{0}
10^{-5}
10^{-10}
10^{-15}
10^{-15}
10^{-10}
10^{-5}
10^{0}
Error
eps line


4a) Catastrophic cancellation ex.
Bad fn. vs. fixed fn.
10^0
10^-5
10^-10
10^-15
10^-15
10^-10
10^-5
10^0
Log scale values
Original
Fixed


4b) Loss of Significance ex.
Bad fn. vs. fixed fn.
10^0
10^-2
10^-4
10^-6
10^-8
10^-10
10^0
10^1
10^2
Values 1-100
Original
Fixed

Published with MATLAB® R2018a