APPM3021 Numerical Methods • Lab 4 • 6/4/2018
Tyson Cross 1239448

Exercises:

---

exercise1.m :

```
% APPM3021 Lab 4 Exercise 1
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

N = 5;
syms f a;
f(a) = 1/3*a.^4 + 2*a.^3 - 5*sin(2*a) + exp(-a/2) - 12;
x = sort(unifrnd(0,5,N,1));
y = double(f(x));
[co, T] = NewtonInterpSimple(x,y);

% Display results
fprintf('% 8s % 8s \n','X','Y')
fprintf('_____________________ \n\n')
for i=1:length(x)
 fprintf(' % 8.2f % 8.2f\n',x(i),y(i))
end

% Polynomial output
fprintf('\nP(x) = ')
for i=1:length(co)
    fprintf('%.3f',co(i));
    if i ~=1
        for j=1:i-1
            fprintf('(x-%.2f)',x(j));
        end
    end

    if i~=length(co)
        fprintf(' + ')
    else
        fprintf('\n')
    end
end
```

When exercise1.m is run in the workspace, the following output is displayed to the command window:

```
       X        Y
_____________________

     2.51    38.11
     3.10    79.08
     3.22    89.97
     3.28    95.83
     4.04   203.70

P(x) = 38.107 + 69.752(x-2.51) + 31.450(x-2.51)(x-3.10) + 12.297(x-2.51)(x-3.10)(x-3.22) + -0.203(x-2.51)(x-3.10)
(x-3.22)(x-3.28)
```

---

exercise2.m :

```
% APPM3021 Lab 4 Exercise 2
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

N = 20;
syms f a;
f(a) = 1/3*a.^4 + 2*a.^3 - 5*sin(2*a) + exp(-a/2) - 12;

x = sort(unifrnd(0,5,N,1));
y = double(f(x));
xq = sort(unifrnd(0,5,round(N/2),1));
yq = NewtonInterp(x,y,xq);

% Display results
fprintf('% 8s % 8s    |% 8s % 8s \n','X','Y','XQ','YQ')
fprintf('_____________________________________________ \n\n')
for i=1:length(x)
    if i <= length(xq)
        fprintf(' % 8.2f % 8.2f   |  % 7.2f % 8.2f \n',x(i),y(i),xq(i),yq(i))
```

```
    else
        fprintf(' % 8.2f % 8.2f    | \n',x(i),y(i))
    end
end
```

When exercise2.m is run in the workspace, the following output is displayed to the command window:

```
       X        Y    |      XQ       YQ
_____________________________________________

     0.27   -13.68   |     0.26   -13.60
     0.50   -15.15   |     1.06   -12.87
     0.57   -15.38   |     1.17   -11.21
     0.61   -15.46   |     1.61    -0.48
     1.00   -13.65   |     1.63     0.19
     1.12   -12.02   |     2.27    25.36
     2.20    22.05   |     2.40    31.76
     2.44    34.21   |     3.87   174.52
     2.82    57.11   |     3.87   174.52
     3.28    95.87   |     3.90   179.45
     3.28    95.91   |
     3.38   106.29   |
     3.51   122.07   |
     3.74   152.71   |
     3.76   155.98   |
     3.80   163.20   |
     4.30   258.46   |
     4.51   308.02   |
     4.59   327.12   |
     4.70   357.70   |
```

---

exercise3.m :

```
% APPM3021 Lab 4 Exercise 3
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

%% Calculations
N = 20;
syms f a;
f = @(a) 1/3*a.^4 + 2*a.^3 - 5*sin(2*a) + exp(-a/2) - 12;
x = sort(unifrnd(0,5,N,1));
y = double(f(x));
xq = sort(unifrnd(0,5,round(N/2),1));
yq = NewtonInterp(x,y,xq);

% Display results
fprintf('% 8s % 8s    |% 8s % 8s \n','X','Y','XQ','YQ')
fprintf('_____________________________________________ \n\n')
for i=1:length(x)
    if i <= length(xq)
        fprintf(' % 8.2f % 8.2f   |  % 7.2f % 8.2f \n',x(i),y(i),xq(i),yq(i))
    else
        fprintf(' % 8.2f % 8.2f   | \n',x(i),y(i))
    end
end

%% Display setting and output setup
scr = get(groot,'ScreenSize');                              % screen resolution
phi = (1 + sqrt(5))/2;
ratio = phi/3;
offset = [ scr(3)/4 scr(4)/4];
fig1 =  figure('Position',...                               % draw figure
        [offset(1) offset(2) scr(3)*ratio scr(4)*ratio]);
set(fig1,'numbertitle','off',...                            % Give figure useful title
        'name','Exercise 3',...
        'Color','white');
fontName='Helvetica';
set(0,'defaultAxesFontName', fontName);                     % Make fonts pretty
set(0,'defaultTextFontName', fontName);
set(groot,'FixedWidthFontName', 'ElroNet Monospace')

% Draw plots

p1 = plot(x,y,...
    'Color',[0.9 0.18 0.18 .6],...
 'LineStyle','-',...
 'LineWidth',1);
hold on
p2 = plot(xq,yq,...
    'Color',[0.18 0.9 0.18 .6],...
 'LineStyle',':',...
 'LineWidth',2,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.18 0.9 0.18],...
    'Marker','o');
```

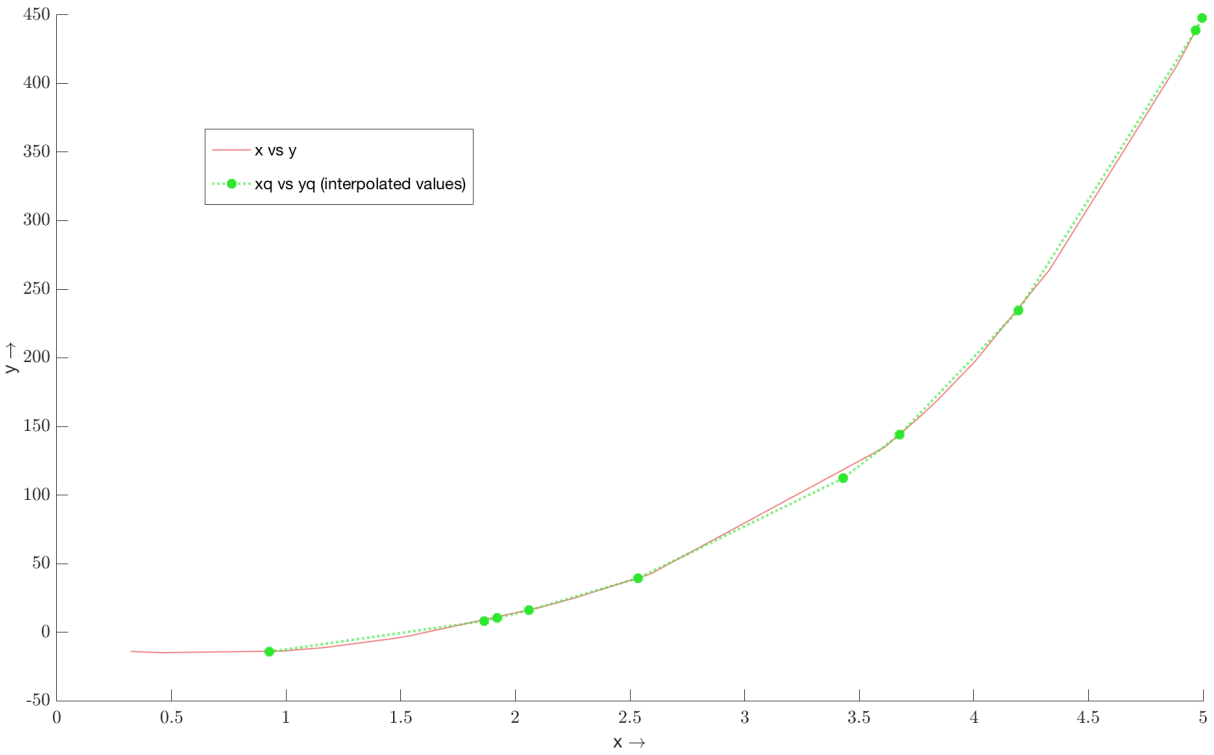
```
hold on

% Axes and labels
ax1 = gca;;
box(ax1,'off');
set(ax1,'FontSize',14,...
    'YMinorTick','off',...
    'XMinorTick','off',...
    'TickLabelInterpreter','latex');
hold on
ylabel('y \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);%,...
xlabel('x \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);
% Legend
legend1 = legend({'x vs y','xq vs yq (interpolated values)'},...
     'Location','best',...
     'Position',[0.19 0.7 0.2 0.09],...
     'Box','on');
hold on

% Adjust figure
pos = get(ax1, 'Position');
pos(1) = 0.08;
pos(3) = pos(3)*1.1;
set(ax1, 'Position', pos)
hold off
```

When exercise3.m is run in the workspace, the following output is displayed to the command window:

```
       X         Y     |       XQ        YQ
_____________________________________________

     0.32    -14.08    |     0.93    -14.34
     0.45    -14.93    |     1.86      8.10
     0.98    -13.83    |     1.92     10.30
     1.17    -11.21    |     2.06     16.01
     1.47     -4.67    |     2.54     39.39
     1.52     -3.19    |     3.43    112.15
     1.54     -2.69    |     3.68    144.17
     2.09     17.42    |     4.19    234.66
     2.16     20.41    |     4.97    438.84
     2.27     25.28    |     5.00    447.89
     2.59     42.49    |
     3.61    134.71    |
     3.61    135.13    |
     3.76    155.86    |
     3.78    160.16    |
     3.84    169.43    |
     4.01    197.57    |
     4.33    264.10    |
     4.89    414.80    |
     4.98    442.58    |
```

When exercise3.m is run in the workspace, the following figure is generated:

Figure 1. Exercise 3



---

exercise_extra.m :

```
% APPM3021 Lab 4 Exercise 3
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

%% Calculations
N = 20;
syms f a poly b;
f = @(a)1/3*a.^4 + 2*a.^3 - 5*sin(2*a) + exp(-a/2) - 12;
x = sort(unifrnd(0,5,N,1));
y = double(f(x));
xq = sort(unifrnd(0,5,round(N/2),1));
poly = NewtonInterpPoly(x,y,xq);    % create a symbolic function of the polynomial
yq = poly(xq);                      % evaluate the polynomial at the requested values

% Display results
fprintf('% 8s % 8s    |% 8s % 8s \n','X','Y','XQ','YQ')
fprintf('_____________________________________________ \n\n')
for i=1:length(x)
    if i <= length(xq)
        fprintf(' % 8.2f % 8.2f   |  % 7.2f % 8.2f \n',x(i),y(i),xq(i),yq(i))
    else
        fprintf(' % 8.2f % 8.2f   | \n',x(i),y(i))
    end
end

%% Display setting and output setup
scr = get(groot,'ScreenSize');                              % screen resolution
phi = (1 + sqrt(5))/2;
ratio = phi/3;
offset = [ scr(3)/4 scr(4)/4];
fig1 =  figure('Position',...                               % draw figure
        [offset(1) offset(2) scr(3)*ratio scr(4)*ratio]);
set(fig1,'numbertitle','off',...                            % Give figure useful title
        'name','Extra Exercise',...
        'Color','white');
fontName='Helvetica';
set(0,'defaultAxesFontName', fontName);                     % Make fonts pretty
set(0,'defaultTextFontName', fontName);
set(groot,'FixedWidthFontName', 'ElroNet Monospace')

% Axes and labels
ax1 = gca;
box(ax1,'off');
set(ax1,'FontSize',14,...
```

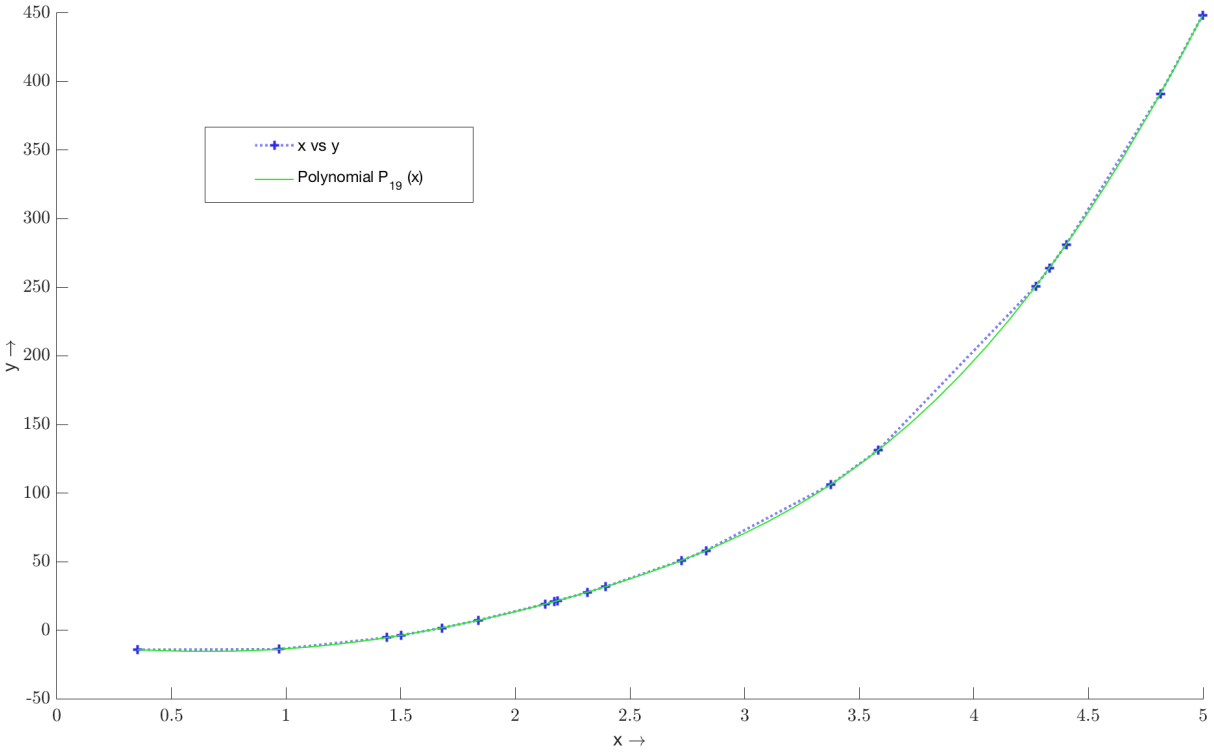
```
    'YMinorTick','off',...
    'XMinorTick','off',...
    'TickLabelInterpreter','latex');
hold on
ylabel('y \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);%,...
xlabel('x \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);

% Draw plots
p1 = plot(x,y,...
    'Color',[0.18 0.18 0.9 .6],...
 'LineStyle',':',...
 'LineWidth',2,...
    'MarkerSize',6,...
    'Marker','+');
hold on

f1 = fplot(ax1, poly,[min(x) max(x)],...
    'Color',[0.18 0.9 0.18 .5],...
 'LineStyle','-',...
 'LineWidth',1);
hold on

% Legend
poly_name = strcat('Polynomial P_{',num2str(length(x)-1),'} (x)');
legend1 = legend({'x vs y',poly_name,},...
     'Location','best',...
     'Position',[0.19 0.7 0.2 0.09],...
     'Box','on');
hold on

% Adjust figure
pos = get(ax1, 'Position');
pos(1) = 0.08;
pos(3) = pos(3)*1.1;
set(ax1, 'Position', pos)
hold off
```

When exercise_extra.m is run in the workspace, the following output is displayed to the command window:

```
       X         Y     |       XQ        YQ
_____________________________________________

     0.35    -14.31    |     0.23    -13.32
     0.97    -13.93    |     0.93    -14.29
     1.44     -5.44    |     1.19    -10.87
     1.50     -3.71    |     1.45     -5.26
     1.68      1.64    |     1.75      4.16
     1.84      7.23    |     2.17     20.73
     2.13     19.03    |     2.79     54.95
     2.17     20.89    |     3.00     70.27
     2.19     21.51    |     3.42    111.29
     2.31     27.67    |     4.89    413.22
     2.40     31.74    |
     2.73     50.86    |
     2.83     58.05    |
     3.38    106.31    |
     3.58    131.40    |
     4.27    250.90    |
     4.33    264.06    |
     4.40    281.19    |
     4.81    391.21    |
     5.00    448.33    |
```

When exercise_extra.m is run in the workspace, the following figure is generated:

Figure 2. Extra Exercise



Questions:

Question 1. 1)

```
% APPM3021 Lab 4 Question 1_1
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');


R = [1101.0;911.3;636.0;451.1;233.5];
T = [25.113;30.131;40.120;50.128;60.136];

[co, Table] = NewtonInterpSimple(R,T);

% Display results
fprintf('% 8s % 8s \n','R(ohm)','T(°C)')
fprintf('_____________________ \n\n')
for i=1:length(R)
 fprintf(' % 8.2f % 8.2f\n',R(i),T(i))
end


disp(' ')
disp('Table of divided difference')
disp(T)
disp(' ')
disp('Newton''s polynomial coeffients:')
disp(co)


% Polynomial output
fprintf('\n%dth order polynomial P(x) = ',numel(R))
for i=1:length(co)
    fprintf('%.3f',co(i));
    if i ~=1
        for j=1:i-1
            fprintf('(x-%.2f)',R(j));
        end
    end

    if i~=length(co)
        fprintf(' + ')
    else
        fprintf('\n')
    end
end
```

When question1_1.m is run in the workspace, the following output is displayed to the command window:

```
  R(ohm)    T(°C)
_____________________

  1101.00    25.11
   911.30    30.13
   636.00    40.12
   451.10    50.13
   233.50    60.14

Table of divided difference
   25.1130
   30.1310
   40.1200
   50.1280
   60.1360


Newton's polynomial coeffients:
   25.1130   -0.0265    0.0000   -0.0000   -0.0000


5th order polynomial P(x) = 25.113 + -0.026(x-1101.00) + 0.000(x-1101.00)(x-911.30) + -0.000(x-1101.00)(x-911.30)
(x-636.00) + -0.000(x-1101.00)(x-911.30)(x-636.00)(x-451.10)
```

Questions:

Question 1. 2)

```
% APPM3021 Lab 4 Question 1_2
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

R = [1101.0;911.3;636.0;451.1;233.5];
```

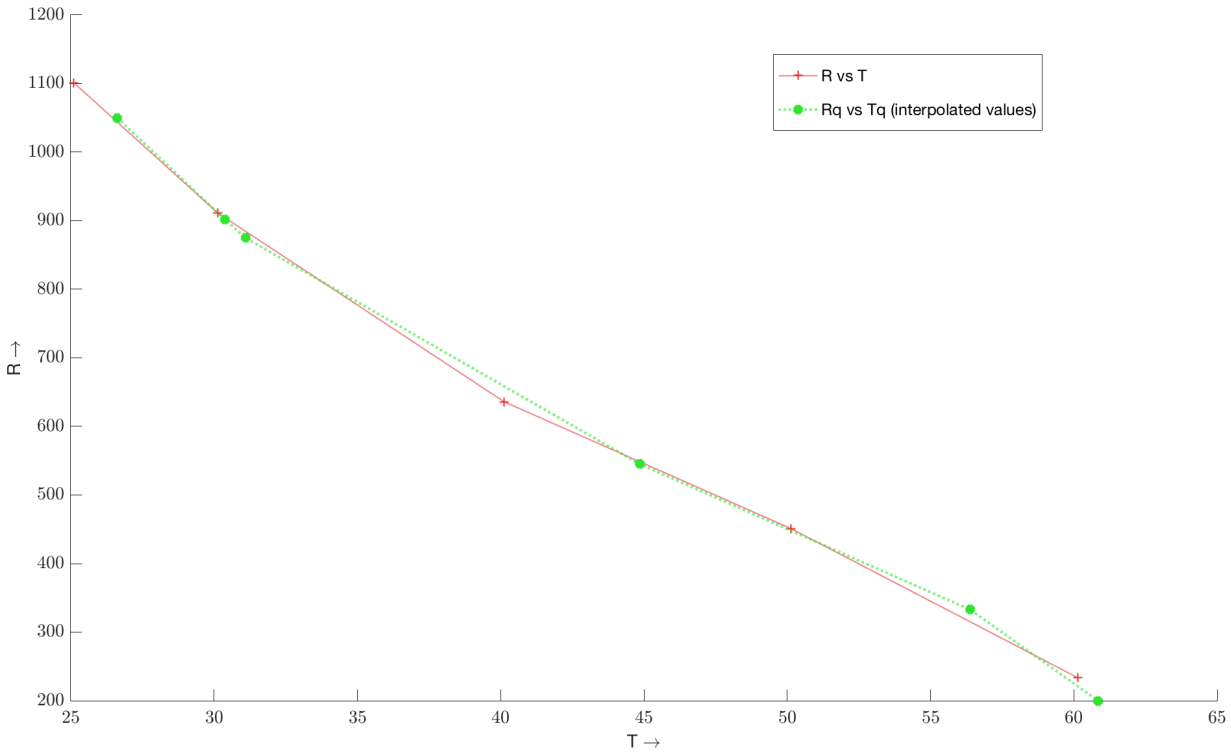
```
T = [25.113;30.131;40.120;50.128;60.136];

Rq = [1050.1;901.56;875.11;545.27;333.1;200];

[co, Table] = NewtonInterpSimple(R,T);
Tq = NewtonInterp(R,T,Rq);

% Display results
fprintf('% 8s % 8s    |% 8s % 8s \n','R(ohm)','Temp(°C)','Rq','Tq')
fprintf('_____________________________________________ \n\n')
for i=1:length(T)
    if i <= length(Rq)
        fprintf(' % 8.2f % 8.2f   |  % 8.2f % 8.2f \n',R(i),T(i),Rq(i),Tq(i))
    else
        fprintf(' % 8.2f % 8.2f   | \n',T(i),T(i))
    end
end

%% Display setting and output setup
scr = get(groot,'ScreenSize');                              % screen resolution
phi = (1 + sqrt(5))/2;
ratio = phi/3;
offset = [ scr(3)/4 scr(4)/4];
fig1 =  figure('Position',...                               % draw figure
        [offset(1) offset(2) scr(3)*ratio scr(4)*ratio]);
set(fig1,'numbertitle','off',...                            % Give figure useful title
        'name','Question 1. 2)',...
        'Color','white');
fontName='Helvetica';
set(0,'defaultAxesFontName', fontName);                     % Make fonts pretty
set(0,'defaultTextFontName', fontName);
set(groot,'FixedWidthFontName', 'ElroNet Monospace')

% Draw plots

p1 = plot(T,R,...
    'DisplayName','R vs T',...
    'Color',[0.9 0.18 0.18 .6],...
 'LineStyle','-',...
 'LineWidth',1,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.18 0.9 0.18],...
    'Marker','+');
hold on
p2 = plot(Tq,Rq,...
 'DisplayName','Rq vs Tq (interpolated values)',...
    'Color',[0.18 0.9 0.18 .6],...
 'LineStyle',':',...
 'LineWidth',2,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.18 0.9 0.18],...
    'Marker','o');
hold on

% Axes and labels
ax1 = gca;;
box(ax1,'off');
set(ax1,'FontSize',14,...
    'YMinorTick','off',...
    'XMinorTick','off',...
    'TickLabelInterpreter','latex');
hold on
ylabel('R \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);%,...
xlabel('T \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);
% Legend
legend1 = legend(ax1,'show');
set(legend1,...
     'Location','best',...
     'Position',[0.602162162162162 0.788098918083462 0.2 0.09],...
     'Box','on');
hold on

% Adjust figure
pos = get(ax1, 'Position');
pos(1) = 0.08;
pos(3) = pos(3)*1.1;
set(ax1, 'Position', pos)
hold off
```

When question1_2.m is run in the workspace, the following output is displayed to the command window:

```
  R(ohm) Temp(°C)    |      Rq       Tq
_____________________________________________

  1101.00    25.11   |   1050.10    26.62
   911.30    30.13   |    901.56    30.39
   636.00    40.12   |    875.11    31.10
   451.10    50.13   |    545.27    44.84
   233.50    60.14   |    333.10    56.37
```

When question1_2.m is run in the workspace, the following figure is generated:

Figure 3. Question 1. 2)



Question 2

```
% APPM3021 Lab 4 Question 2
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

N = [5,10,20,30];
M = 101;
syms f a;
f(a) = 1/(1+25*a^2);
xq = linspace(-5,5,M)';
max_diff = zeros(1,numel(N));

for i=1:numel(N)
    x{:,i} = linspace(-5,5,N(i))';
    y{:,i} = double(f(x(i)));
    yq{:,i} = NewtonInterp(x{:,i},y{:,i},xq);
end

for i=1:numel(N)
    k=1;
    for j=1:numel(x{:,i})-1
        while k<numel(xq) && xq(k)>=x{:,i}(j) && xq(k)<=x{:,i}(j+1)
            current_diff = abs(max(abs(y{:,i}(j)))-max(abs(yq{:,i}(k))));
            if current_diff > max_diff(i);
                max_diff(i) = current_diff;
            end
            k=k+1;
        end
    end
end
disp(' ')

% Display results
fprintf('Max Differences \n')
fprintf('________________\n\n')
for i=1:length(max_diff)
fprintf('n = %2d   Max difference = %.3f \n\n',N(i),max_diff(i))
end

%% Display setting and output setup
scr = get(groot,'ScreenSize');                                  % screen resolution
phi = (1 + sqrt(5))/2;
ratio = phi/3;
```

```
offset = [ scr(3)/4 scr(4)/4];
fig1 =  figure('Position',...                                   % draw figure
        [offset(1) offset(2) scr(3)*ratio scr(4)*ratio]);
set(fig1,'numbertitle','off',...                                % Give figure useful title
        'name','Question 2',...
        'Color','white');
fontName='Helvetica';
set(0,'defaultAxesFontName', fontName);                         % Make fonts pretty
set(0,'defaultTextFontName', fontName);
set(groot,'FixedWidthFontName', 'ElroNet Monospace')
ax1 = gca;

% Draw plots
linS = {'--',':','-.'};

for i=1:numel(N)
    num_ax = strcat('ax',num2str(i));
    num_leg = strcat('legend',num2str(i));

    variable.(num_ax) = subplot(2,2,i);
    f1 = fplot(variable.(num_ax), f,[min(x{:,i}) max(x{:,i})],...
    'Color',[0.18 0.9 0.18 .5],...
 'LineStyle','-',...
 'LineWidth',1);
    hold on

    p1 = plot(variable.(num_ax),x{:,i},y{:,i},...
        'DisplayName','y',...
        'Color',[0.9 0.18 0.18 .5],...
        'Linestyle','-',...
        'LineWidth',1.5,...
        'MarkerSize',6,...
        'MarkerFaceColor',[0.18 0.9 0.18],...
        'Marker','x');
    hold on

    p2 = plot(variable.(num_ax),xq,yq{:,i},...
        'DisplayName','yq',...
        'Color',[0.18 0.18 0.9 .5],...
        'Linestyle',linS{mod(i,numel(linS))+1},...
        'LineWidth',1.5);
    hold on

    % Axes and labels
    box(variable.(num_ax),'off');
    set(variable.(num_ax),'FontSize',14,...
        'YMinorTick','off',...
        'XMinorTick','off',...
        'XTick',x{:,1},...
        'YLim',[min(yq{:,i}) ceil(max(yq{:,i}))],...
        'TickLabelInterpreter','latex');
    hold on
    ylabel('y \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);%,...
    xlabel('x \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);
    title(strcat('N=',num2str(N(i))),...
        'FontName',fontName,...
        'FontSize',14);

    % Legend
    variable.(num_leg) = legend('show');
    set(variable.(num_leg),...
         'Location','best',...
         'Box','off');
    hold on

    switch i
        case 1
            set(variable.(num_leg),...
                'Position',[0.260797293759768 0.609026492988499 0.105212355212355 0.0818895035093611]);
        case 2
            set(variable.(num_leg),...
                'Position',[0.752591052429539 0.767796055549154 0.184362934362934 0.0818895035093611]);
        case 3
            set(variable.(num_leg),...
                'Position',[0.248249031211505 0.255092594135115 0.105212355212355 0.0818895035093609]);
        case 4
            set(variable.(num_leg),...
                'Position',[0.689555191267665 0.248910213918732 0.105212355212355 0.081889503509361]);
        end
end
```

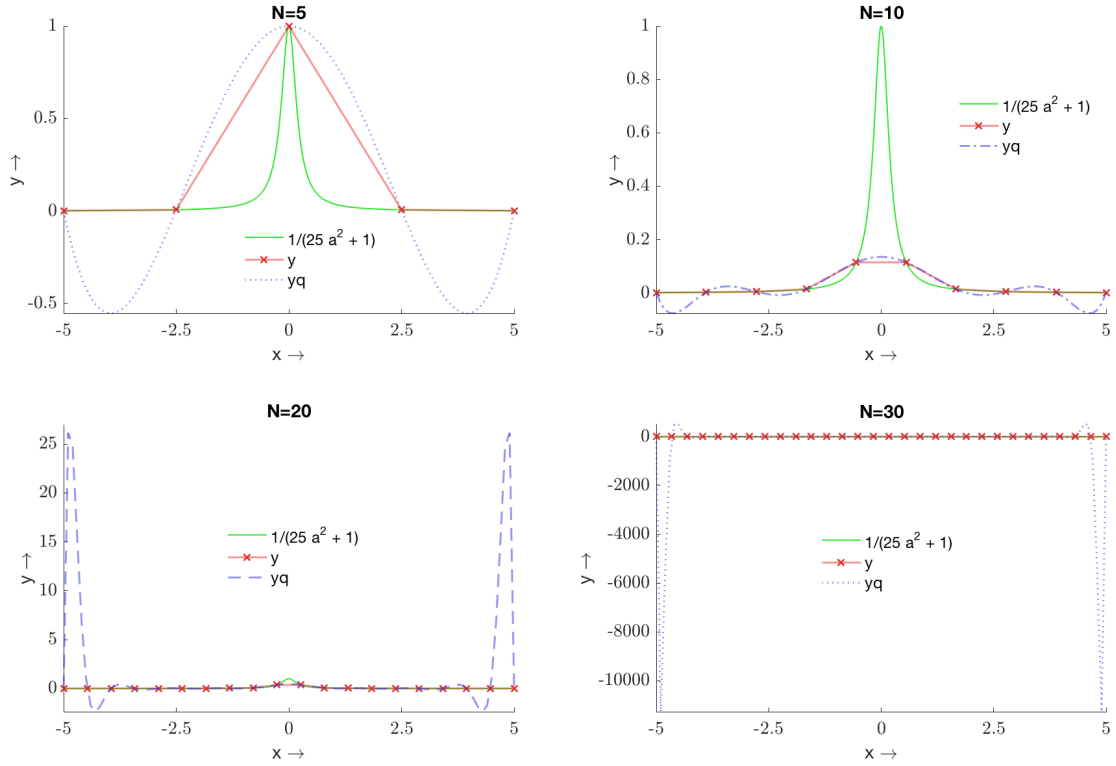
When question2.m is run in the workspace, the following output is displayed to the command window:

```
Max Differences
_______________

n =  5   Max difference = 0.994

n = 10   Max difference = 0.097
```

```
n = 20   Max difference = 26.141

n = 30   Max difference = 11269.111
```

When question2.m is run in the workspace, the following figure is generated:

Figure 4. Question 2




```
Weierstrass Approximation Theorem states that all continuous functions defined on a given interval can be
continously uniformly well approximated by a polynomial function of a sufficiently high degree.
However, Runge's phenomenon shows that for certain functions interpolated uniformlyover equidistance points,
that there can be an oscillation in the interpolated approximation, towards the positive and negative ends of
the interval, with an error that increases (towards infinity) as the degree of the polynomial increases.
Each nth derivative of the Runge function increases in magnitude rapidly, increasing the size of the error as
the degree of the approximation increases.
```

Question 3

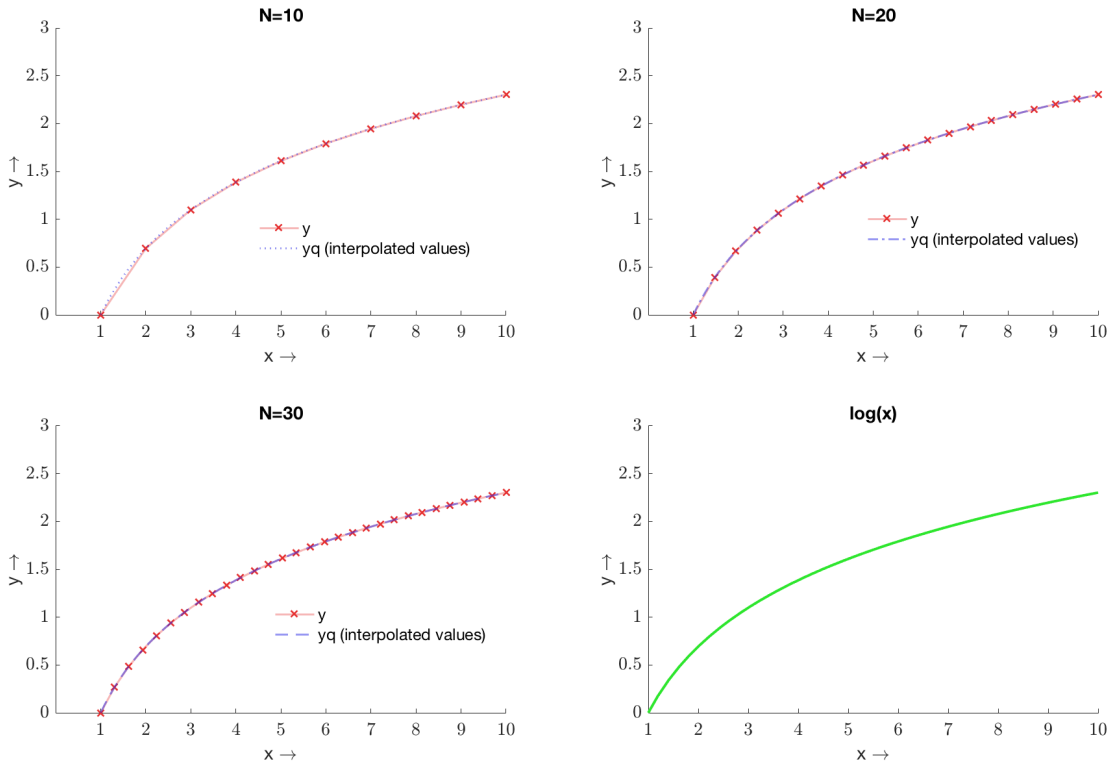
```
% APPM3021 Lab 4 Question 3
% Tyson Cross 1239448

clc; clear all;
format loose
rng('shuffle');

N = [10,20,30];
M = 101;
syms f a;
f(a) = log(a);
xq = linspace(1,10,M)';
max_diff = zeros(1,numel(N));

for i=1:numel(N)
    x{:,i} = linspace(1,10,N(i))';
    y{:,i} = double(f(x(i)));
    yq{:,i} = NewtonInterp(x{:,i},y{:,i},xq);
end

for i=1:numel(N)
    k=1;
    for j=1:numel(x{:,i})-1
        while k<numel(xq) && xq(k)>=x{:,i}(j) && xq(k)<=x{:,i}(j+1)
            current_diff = abs(max(abs(y{:,i}(j)))-max(abs(yq{:,i}(k))));
            if current_diff > max_diff(i);
                max_diff(i) = current_diff;
            end
            k=k+1;
        end
    end
```

```
end

% Display results
fprintf('Max Differences \n')
fprintf('________________\n\n')
for i=1:length(max_diff)
fprintf('n = %2d   Max difference = %.3f \n\n',N(i),max_diff(i))
end

%% Display setting and output setup
scr = get(groot,'ScreenSize');                              % screen resolution
phi = (1 + sqrt(5))/2;
ratio = phi/3;
offset = [ scr(3)/4 scr(4)/4];
fig1 =  figure('Position',...                               % draw figure
        [offset(1) offset(2) scr(3)*ratio scr(4)*ratio]);
set(fig1,'numbertitle','off',...                            % Give figure useful title
        'name','Question 3',...
        'Color','white');
fontName='Helvetica';
set(0,'defaultAxesFontName', fontName);                     % Make fonts pretty
set(0,'defaultTextFontName', fontName);
set(groot,'FixedWidthFontName', 'ElroNet Monospace')
ax1 = gca;

% Draw plots
linS = {'--',':','-.'};

for i=1:numel(N)
    num_ax = strcat('ax',num2str(i));
    num_leg = strcat('legend',num2str(i));

    variable.(num_ax) = subplot(2,2,i);

    p1 = plot(variable.(num_ax),x{:,i},y{:,i},...
        'DisplayName','y',...
        'Color',[0.9 0.18 0.18 0.35],...
        'Linestyle','-',...
        'LineWidth',1.5,...
        'MarkerSize',6,...
        'MarkerFaceColor',[0.18 0.9 0.18],...
        'Marker','x');
    hold on

    p2 = plot(variable.(num_ax),xq,yq{:,i},...
        'DisplayName',strcat('yq (interpolated values)'),...
        'Color',[0.18 0.18 0.9 .5],...
        'Linestyle',linS{mod(i,numel(linS))+1},...
        'LineWidth',1.5);
    hold on

    % Axes and labels
    box(variable.(num_ax),'off');
    set(variable.(num_ax),'FontSize',14,...
        'YMinorTick','off',...
        'XMinorTick','off',...
        'XTick',x{:,1},...
        'YLim',[min(y{:,i}) ceil(max(y{:,i}))],...
        'TickLabelInterpreter','latex');
    hold on
    ylabel('y \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);%,...
    xlabel('x \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);
    title(strcat('N=',num2str(N(i))),...
        'FontName',fontName,...
        'FontSize',14);

    % Legend
    variable.(num_leg) = legend('show');
    set(variable.(num_leg),...
         'Location','best',...
         'Box','off');
    hold on

        switch i
        case 1
            set(variable.(num_leg),...
                'Position',[0.277746799538157 0.648487789608799 0.16457528957529 0.0533054315296785]);
        case 2
            set(variable.(num_leg),...
                'Position',[0.729670720212078 0.65930695498747 0.16457528957529 0.0533054315296785]);
        case 3
            set(variable.(num_leg),...
                'Position',[0.289329811121169 0.188560935793336 0.16457528957529 0.0533054315296784]);
         end

end

ax4 = subplot(2,2,4);

f1 = fplot(ax4, f,[min(x{:,i}) max(x{:,i})],...
    'DisplayName','ln(x)',...
```

```
    'Color',[0.18 0.9 0.18 .2],...
 'LineStyle','-',...
 'LineWidth',2);
    hold on

        % Axes and labels
    box(ax4,'off');
    set(ax4,'FontSize',14,...
        'YMinorTick','off',...
        'XMinorTick','off',...
        'XTick',x{:,1},...
        'YLim',[min(y{:,i}) ceil(max(y{:,i}))],...
        'TickLabelInterpreter','latex');
    hold on
    ylabel('y \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);%,...
    xlabel('x \rightarrow',...
        'FontName',fontName,...
        'FontSize',14);
    title('log(x)',...
        'FontName',fontName,...
        'FontSize',14);
```

```
Max Differences
_______________

n = 10   Max difference = 0.688

n = 20   Max difference = 0.372

n = 30   Max difference = 0.239
```

Figure 5. Question 3



The interpolated approximating polynomial closely matches the function ln(x). Unlike the polynomial approximating the Runge function in Question 2, the magnitude of the derivative of the function ln(x) gets smaller with each nth derivative, so the error reduces as the degree of the polynomial increases.

Functions and Code:

NewtonInterpSimple.m :

```
 function [co,T] = NewtonInterpSimple(x,y)
% NewtonInterpSimple() performs Newton-Gregory divided difference interpolation,
% using the column vector values for x and y, and returns a row vector
% containing the polynomial coefficient values in 'co' and the table of
% differences in matrix T

    % x entries are the x values
    % y entries are value at f(x)

    [n m] = size(x); % m is the number of data points
    T = zeros(n, n);
    T(:,1) = double(y)';
    for j=2:n
        for i=1:(n-j+1)
            T(i,j) = ( T(i+1,j-1) - T(i,j-1) ) / ( x(i+j-1) - x(i) );
        end
    end
    co = T(1,:);
end
```

NewtonInterp.m :

```
 function yq = NewtonInterp(x,y,xq)
% NewtonInterp() performs Newton-Gregory divided difference interpolation,
% using the column vector values for x and y, and query point xq
% Output is a row vector containing the interpolated y values

    % x entries are the x values
    % y entries are value at f(x)
    % xq values are the queries for interpolation
    % yq values is the resulting interpolated y values

    [n ~] = size(x); % m is the number of data points
    T = zeros(n, n);
    T(:,1) = double(y)';
    for j=2:n
        for i=1:(n-j+1)
            T(i,j) = ( T(i+1,j-1) - T(i,j-1) ) / ( x(i+j-1) - x(i) );
        end
    end

    co = T(1,:);
    yq = zeros(length(xq),1);

    for k=1:length(xq)
        mult = 1;
        sum = 0;
        for i=1:length(co)
            if i==1
                sum = co(i);
            else
                mult = 1;
                for j=1:i-1
                    mult = mult .* ( xq(k) - x(j) );
                end
                sum = sum + co(i) * mult;
            end
        end
        yq(k) = sum;
    end
 end
```

NewtonInterpPoly.m :

```
function poly = NewtonInterpPoly(x,y,xq)
% NewtonInterpPoly() performs Newton-Gregory divided difference interpolation,
% using the column vector values for x and y, and query point xq
% Output is a polynomial function of degree n-1 where n is the length of
% the x values

    % x entries are the x values
    % y entries are value at f(x)
    % xq values are the queries for interpolation
 % poly is the resulting polynomial
```

```
    [n m] = size(x); % m is the number of data points
    T = zeros(n, n);
    T(:,1) = double(y)';
    for j=2:n
        for i=1:(n-j+1)
            T(i,j) = ( T(i+1,j-1) - T(i,j-1) ) / ( x(i+j-1) - x(i) );
        end
    end

    co = T(1,:);

    for i=1:length(co)
    if i ==1
        str = strcat('@(b)',num2str(co(i)));
    else
        str = strcat(str,num2str(co(i)));
        for j=1:i-1
            str = strcat(str,'.*(b-',num2str(x(j)),')');
        end
    end
    if i~=length(co)
        str = strcat(str,'+');
    end
    end
    poly = str2func(str);

end
```