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
______________________

     0.68   -15.48
     1.30    -8.61
     2.94    65.90
     3.51   122.42
     4.97   440.98

P(x) = -15.478 + 10.987(x-0.68) + 15.300(x-0.68)(x-1.30) + 3.035(x-0.68)(x-1.30)(x-2.94) + 1.509(x-0.68)(x-1.30)
(x-2.94)(x-3.51)
```

---

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
_____________________________________________

     0.07   -11.69   |     0.20   -12.99
     0.21   -13.12   |     0.67   -15.48
     0.72   -15.42   |     0.67   -15.48
     0.78   -15.26   |     0.82   -15.05
     0.99   -13.75   |     1.04   -13.11
     1.43    -5.61   |     2.97    67.90
     1.50    -3.86   |     3.94   185.57
     1.85     7.55   |     4.12   220.23
     1.94    10.93   |     4.60   330.93
     2.29    26.69   |     4.93   428.16
     3.28    96.48   |
     3.32   100.35   |
     3.33   101.15   |
     3.53   124.34   |
     3.57   130.08   |
     3.89   177.01   |
     4.15   224.49   |
     4.38   276.28   |
     4.65   345.24   |
     4.93   426.27   |
```

---

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','Newton-Gregory Divided Difference Interpolation',...
        '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,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.9 0.18 0.18],...
    'Marker','o');
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','+');
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
```

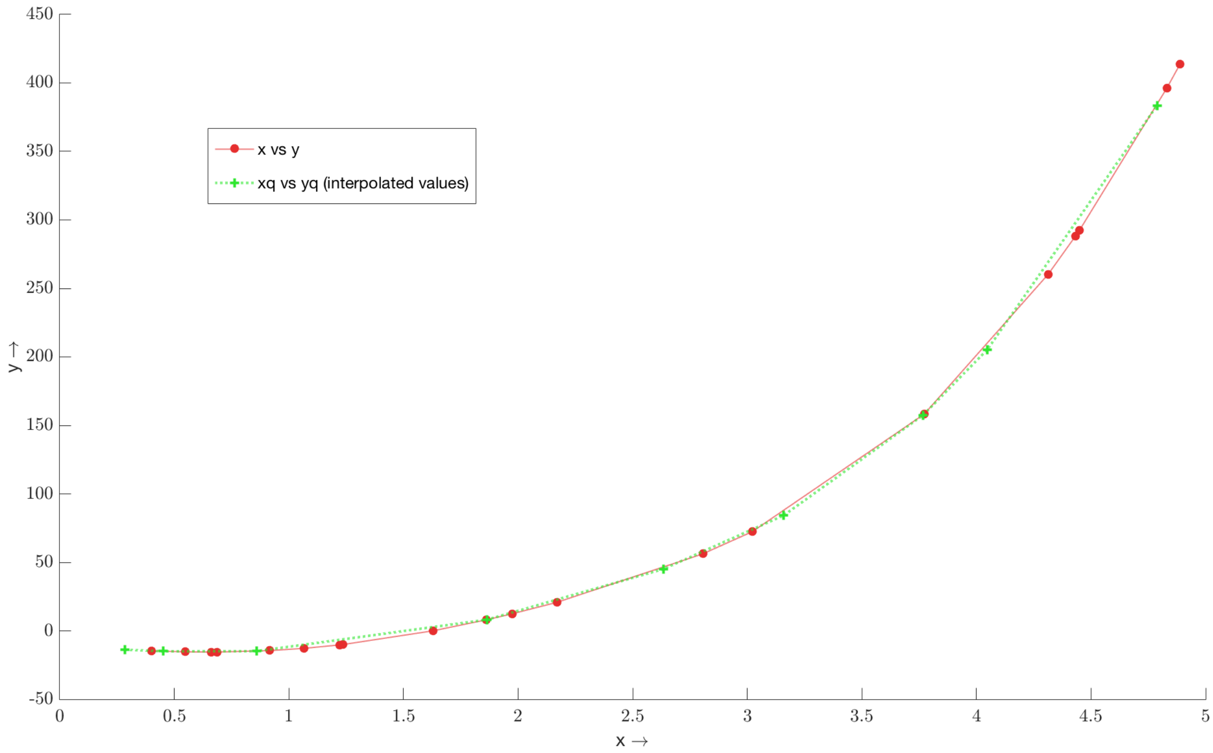
When exercise3.m is run in the workspace, the following output is displayed to the command window:

```
       X        Y     |      XQ       YQ

     0.40   -14.64    |     0.28   -13.78
     0.55   -15.33    |     0.45   -14.93
     0.66   -15.49    |     0.86   -14.85
     0.69   -15.47    |     1.87     8.19
     0.92   -14.42    |     2.63    45.14
     1.07   -12.78    |     3.16    84.28
     1.22   -10.27    |     3.77   157.42
     1.24   -10.00    |     4.05   205.23
     1.63     0.03    |     4.79   383.54
     1.86     8.03    |     4.79   383.64
     1.97    12.45    |
     2.17    20.86    |
     2.81    56.19    |
     3.02    72.41    |
     3.77   158.15    |
     4.31   260.30    |
     4.43   288.30    |
     4.45   292.44    |
     4.83   396.17    |
     4.89   413.64    |
```

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

Figure 1. Newton-Gregory Divided Difference Interpolation



---

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);

% Display results
fprintf('% 8s % 8s    |% 8s \n','X','Y','XQ')
fprintf('____________________________________ \n\n')
for i=1:length(x)
    if i <= length(xq)
        fprintf(' % 8.2f % 8.2f   |  % 7.2f \n',x(i),y(i),xq(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','Newton-Gregory Divided Difference Polynomial Interpolation',...
        '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','o');
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
```

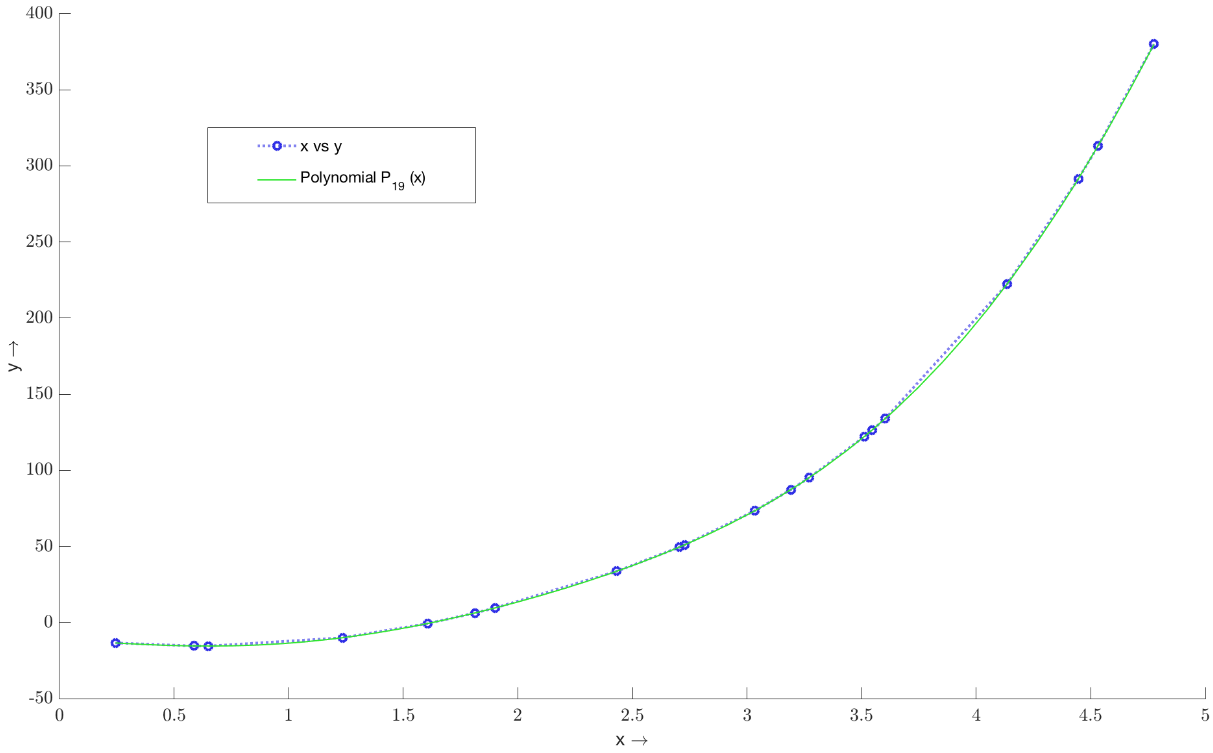
When exercise_extra.m is run in the workspace, the following output is displayed to the command window:

```
       X        Y    |      XQ
____________________________________

     0.24   -13.43   |     0.08
     0.59   -15.42   |     0.12
     0.65   -15.49   |     0.86
     1.23   -10.03   |     1.00
     1.61    -0.71   |     1.15
     1.81     6.24   |     1.78
     1.90     9.55   |     1.98
     2.43    33.66   |     2.82
     2.70    49.52   |     3.89
     2.73    50.96   |     3.95
     3.03    73.35   |
     3.19    87.26   |
     3.27    95.09   |
     3.51   122.13   |
     3.55   126.35   |
     3.60   133.92   |
     4.13   222.16   |
     4.45   291.40   |
     4.53   313.04   |
     4.78   379.96   |
```

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

Figure 2. Newton-Gregory Divided Difference Polynomial Interpolation



Questions:

Question 1 (a) (i)

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
```

Functions and Code:

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 vectorcontaining 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 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,:);
    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
```

Functions and Code:

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
```