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
T = table (x, y);
T.Properties.VariableNames = {'X','Y'};
disp(T)

% Polynomial output
fprintf('P(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:

```
    <strong>   X   </strong><strong>      Y   </strong>
    <strong>_______</strong>    <strong>______</strong>

    0.49332    -15.13
     1.9651    12.067
     2.2932    26.613
     2.9483    66.555
     3.0564    75.252

P(x) = -15.130 + 18.479(x-0.49) + 14.365(x-0.49)(x-1.97) + 1.040(x-0.49)(x-1.97)(x-2.29) + 2.684(x-0.49)(x-1.97)
(x-2.29)(x-2.95)
```

---

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
T1 = table (x, y);
T1.Properties.VariableNames = {'X','Y'};
disp(T1)
disp(' ')

T2 = table (xq, yq);
T2.Properties.VariableNames = {'XQ','YQ'};
```

```
disp(T2)
```

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

```
    <strong>   X   </strong><strong>       Y   </strong>
    <strong>_______</strong>    <strong>_______</strong>

    0.11176     -12.16
    0.45398    -14.943
    0.51094    -15.201
     1.0097    -13.496
     1.3944    -6.5483
     1.4523    -5.0808
     2.1697     20.808
     2.5631      40.92
     2.6118     43.774
     2.6196     44.245
     2.7133      50.05
     3.2957     97.597
     3.3665     105.13
     3.9283     183.75
     3.9465     186.93
     4.1159     218.59
     4.2983     257.05
     4.3603     271.16
      4.646     343.32
     4.7311     367.08


    <strong>  XQ   </strong><strong>      YQ   </strong>
    <strong>_______</strong>    <strong>_______</strong>

    0.85698    -14.859
    0.88151    -14.693
     1.2266     -10.19
     1.5016    -3.7519
     1.8461     7.4677
      2.712     49.972
     2.9196     64.373
     3.3432      102.6
     4.4181     284.83
     4.7899     384.11
```

---

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
T1 = table (x, y);
T1.Properties.VariableNames = {'X','Y'};
disp(T1)
disp(' ')
T2 = table (xq, yq);
T2.Properties.VariableNames = {'XQ','YQ'};
disp(T2)

%% 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,...
```

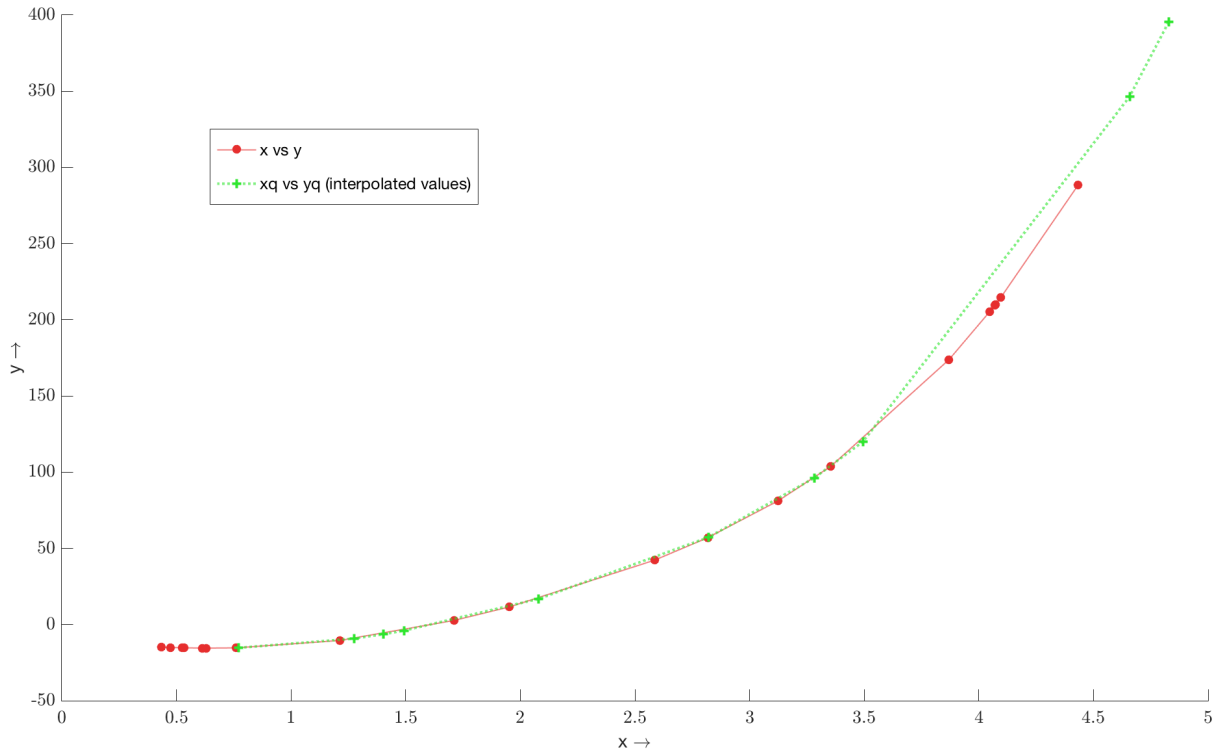
```
    'Color',[0.9 0.18 0.18 .6],...
 'LineStyle','-',...
 'LineWidth',1,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.9 0.18 0.18],...
    'Marker','o');
hold on
p2 = plot(xq,yq,...
    'Color',[0.18 0.9 0.18 .6],...
 'LineStyle',':',...
 'LineWidth',2,...
    'MarkerSize',6,...
 'MarkerFaceColor',[0.18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

```
    <strong>   X   </strong><strong>       Y   </strong>
    <strong>_______</strong>    <strong>_______</strong>

    0.43462    -14.838
    0.47283    -15.037
     0.5251    -15.254
     0.5349    -15.287
    0.61259    -15.461
     0.6297    -15.478
    0.75928    -15.323
     1.2126    -10.451
     1.7113     2.6935
     1.9512     11.514
     2.5857     42.231
     2.8174     56.991
     3.1247      81.17
     3.3547     103.85
     3.8702      173.9
     4.0466     205.17
     4.0691     209.46
     4.0727     210.16
     4.0962     214.72
     4.4331     288.44


    <strong>  XQ   </strong><strong>      YQ   </strong>
    <strong>_______</strong>    <strong>_______</strong>

    0.77276    -15.277
     1.2744    -9.2467
     1.4023     -6.353
     1.4933    -3.9801
     2.0796     16.829
     2.8203     57.197
     3.2815     96.136
     3.4943     119.96
     4.6575     346.48
     4.8281     395.46
```

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
T1 = table (x, y);
T1.Properties.VariableNames = {'X','Y'};
disp(T1)
disp(' ')
disp(['XQ:', mat2str(round(xq',4))]);
disp(' ')


%% 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');
```

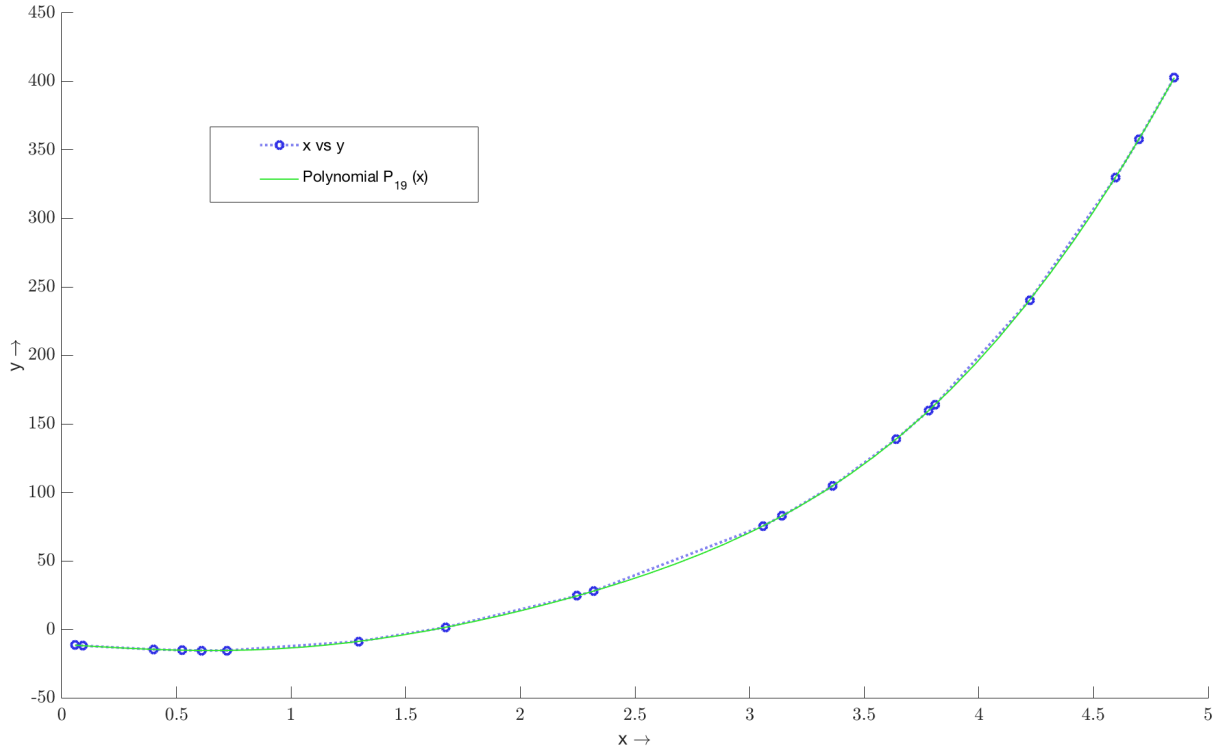
```
hold on
ylabel('y \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);%,...
xlabel('x \rightarrow',...
    'FontName',fontName,...
    'FontSize',14);

% 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:

```
    <strong>   X    </strong><strong>       Y   </strong>
    <strong>________</strong>    <strong>_______</strong>

    0.057997    -11.607
    0.092473    -11.963
     0.40077    -14.636
     0.52602    -15.257
     0.61082    -15.459
     0.72188     -15.42
      1.2947    -8.8216
      1.6757     1.5132
       2.248     24.441
      2.3207     27.964
      3.0581     75.399
      3.1424      82.76
       3.363     104.75
      3.6387     138.76
      3.7815     159.67
      3.8105     164.22
      4.2219     240.37
      4.5967     330.02
      4.6977     357.64
      4.8526     402.83


XQ:[0.5917 1.572 2.0367 2.1889 2.2876 2.8128 3.4519 3.9059 4.3573 4.3693]
```

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
```