Code

Function file (lagrangepol.m)

```
function pval = lagrangepol(X, Y, Xp)
    n = length(X);
    pval = zeros(size(Xp));

    for k = 1:length(Xp)
        x = Xp(k);
        p = 0;

        for i = 1:n
            term = Y(i);
            for j = 1:n
                if i ~= j
                    term = term * (x - X(j)) / (X(i) - X(j));
                end
            end
            p = p + term;
        end

        pval(k) = p;
    end
end
```

Script file:

```
clc;
clear all;
close all;

N = 10;
xdata = linspace(-pi, pi, N);
ydata = xdata .* sinh(xdata);
xval = linspace(-pi, pi, 100);

tic
yvalAprox = lagrangepol(xdata, ydata, xval);
toc

plot(xval, xval .* sinh(xval), 'g', 'linewidth', 4);
hold on
plot(xdata, ydata, 'R+');
plot(xval, yvalAprox, 'k');
legend('true value', 'data points', 'approximation');
xlabel('x value');
ylabel('f(x)');
hold off
maxApproxError = max(abs(xval .* sinh(xval) - yvalAprox)) / max(abs(xval .*
sinh(xval)));
rmsError = sqrt(sum((xval .* sinh(xval) - yvalAprox).^2) / length(yvalAprox));
disp(['Root mean square error is: ', num2str(rmsError)]);
disp(['Maximum approximate error is: ', num2str(maxApproxError)]);
```

In–Lab Exercise 02

Write a MATLAB function m-file called Newtonpol.m that computes the Newton polynomials for any k. Follow the same signature and structure as the previous question. Complete the Table:2 for Newton interpolating polynomials.

Programs

Function file (newtonpol.m)

```
function [yint] = newtonpol(x, y, xint)
    n = length(x);
    a(1) = y(1);

    % Calculate divided differences
    for i = 1:n-1
        divDIF(i, 1) = (y(i+1) - y(i)) / (x(i+1) - x(i));
    end

    for j = 2:n-1
        for i = 1:n-j
            divDIF(i, j) = (divDIF(i+1, j-1) - divDIF(i, j-1)) / (x(j+i) - x(i));
        end
    end

    % Set coefficients
    for j = 2:n
        a(j) = divDIF(1, j-1);
    end

    % Evaluate the Newton polynomial at xint
    yint = a(1);
    xn = 1;

    for k = 2:n
        xn = xn * (xint - x(k-1));
        yint = yint + a(k) * xn;
    end

    % Display the result
    disp(['Interpolated value at x = ', num2str(xint), ' is ', num2str(yint)]);
end
```

Script file:

```
clc;
clear all;
close all;

N = 10;
xdata = linspace(-3, 3, N);
ydata = 2 * sin(pi * xdata / 6);
xval = linspace(-3, 3, 100);

% Preallocate yvalAprox for better performance
yvalAprox = zeros(size(xval));

tic
for i = 1:length(xval)
    yvalAprox(i) = newtonpol(xdata, ydata, xval(i));
end
toc

% Plotting
plot(xval, 2 * sin(pi * xval / 6), 'g', 'linewidth', 4);
hold on
plot(xdata, ydata, 'R+');
plot(xval, yvalAprox, 'k');
legend('true value', 'data points', 'approximation');
xlabel('x value');
ylabel('f(x)');
hold off

% Error Calculation
maxApproxError = max(abs(2 * sin(pi * xval / 6) - yvalAprox)) / max(abs(2 * sin(pi *
xval / 6)));
rmsError = sqrt(sum((2 * sin(pi * xval / 6) - yvalAprox).^2) / length(yvalAprox));

disp(['Root mean square error is: ', num2str(rmsError)]);
disp(['Maximum approximate error is: ', num2str(maxApproxError)]);
```