3.

$w_{n+1}(x) = \prod_{j=0}^{n}(x-x_j)$ from (8.6)

$w_{n+1}'(x) = \sum_{k=0}^{n} \prod_{\substack{j=0\\ j\neq k}}^{n} (x-x_j)$

Let $x=x_i$, and every term has the factor $(x_i-x_i)=0$ except for the term $k=i$, so we get

$w_{n+1}'(x_i) = \prod_{\substack{j=0\\ j\neq i}}^{n} (x_i-x_j)$ #

Check (8.5):

From (8.3), we have $l_i(x) = \prod_{\substack{j=0\\ j\neq i}}^{n} \frac{x-x_j}{x_i-x_j}$

And use below to rewrite it:

$$w_{n+1}(x) = (x-x_i)\prod_{\substack{j=0\\ j\neq i}}^{n}(x-x_j)$$

$$w_{n+1}'(x_i) = \prod_{\substack{j=0\\ j\neq i}}^{n}(x_i-x_j)$$

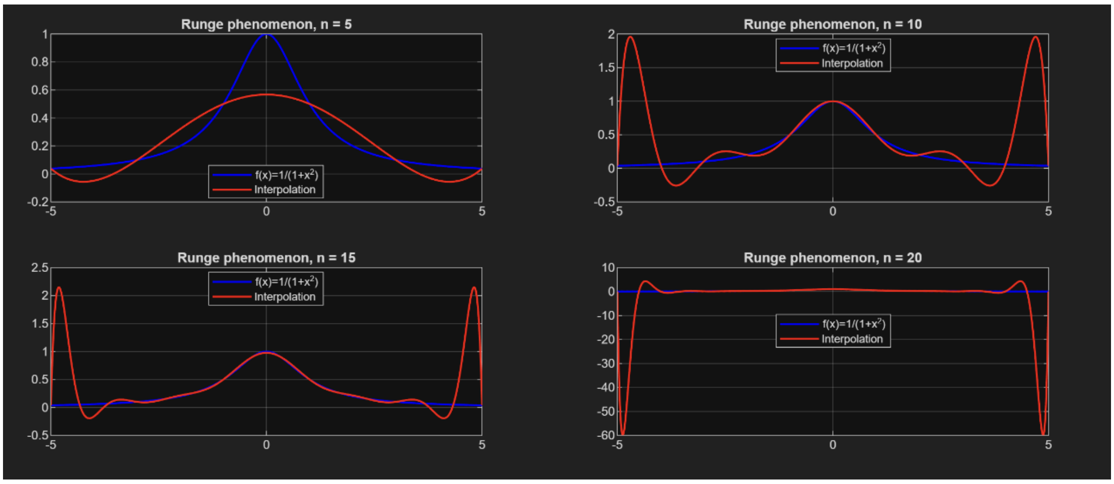
So we get $l_i(x) = \dfrac{\prod_{\substack{j=0\\ j\neq i}}^{n}(x-x_j)}{\prod_{\substack{j=0\\ j\neq i}}^{n}(x_i-x_j)} = \dfrac{w_{n+1}(x)}{(x-x_i)w_{n+1}'(x_i)}$

Then $\Pi_n(x) = \sum_{i=0}^{n} y_i l_i(x)$

$= \sum_{i=0}^{n} \dfrac{w_{n+1}(x)}{(x-x_i)w_{n+1}'(x_i)} y_i$ #

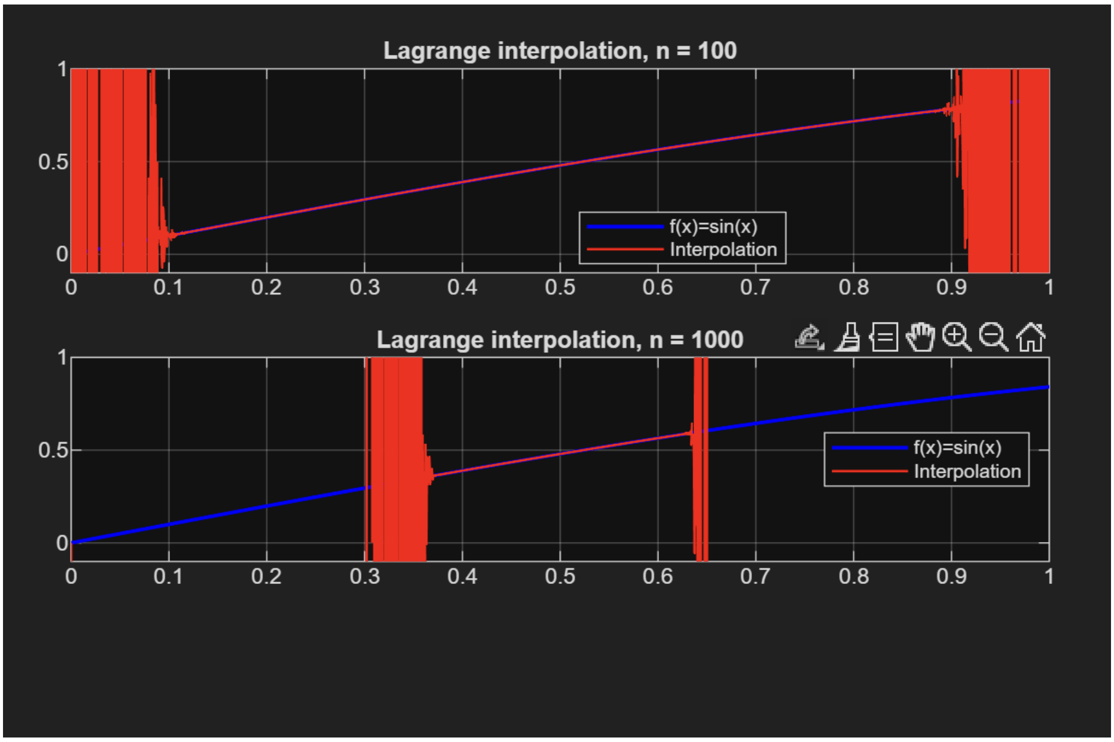
程式作業 1 Matlab 程式碼

```
clear; close all; clc;

f = @(x) 1 ./ (1 + x.^2);

x_plot = linspace(-5, 5, 1000);

y_true = f(x_plot);

n_values = [5, 10, 15, 20];

figure;
for k = 1:length(n_values)
    n = n_values(k);

    x_nodes = linspace(-5, 5, n+1);
    y_nodes = f(x_nodes);

    p_coeff = polyfit(x_nodes, y_nodes, n);
    y_interp = polyval(p_coeff, x_plot);

    subplot(2,2,k);
    plot(x_plot, y_true, 'b-', 'LineWidth', 1.5); hold on;
    plot(x_plot, y_interp, 'r-', 'LineWidth', 1.5);
    hold off;

    title(sprintf('Runge phenomenon, n = %d', n));
    legend('f(x)=1/(1+x^2)', 'Interpolation', 'Nodes', 'Location','Best');
    grid on;
end
```


Runge phenomenon, n = 5
1
0.8
0.6
0.4
0.2
0
-0.2
-5
0
5
f(x)=1/(1+x^2)
Interpolation
Runge phenomenon, n = 10
2
1.5
1
0.5
0
-0.5
-5
0
5
f(x)=1/(1+x^2)
Interpolation
Runge phenomenon, n = 15
2.5
2
1.5
1
0.5
0
-0.5
-5
0
5
f(x)=1/(1+x^2)
Interpolation
Runge phenomenon, n = 20
10
0
-10
-20
-30
-40
-50
-60
-5
0
5
f(x)=1/(1+x^2)
Interpolation

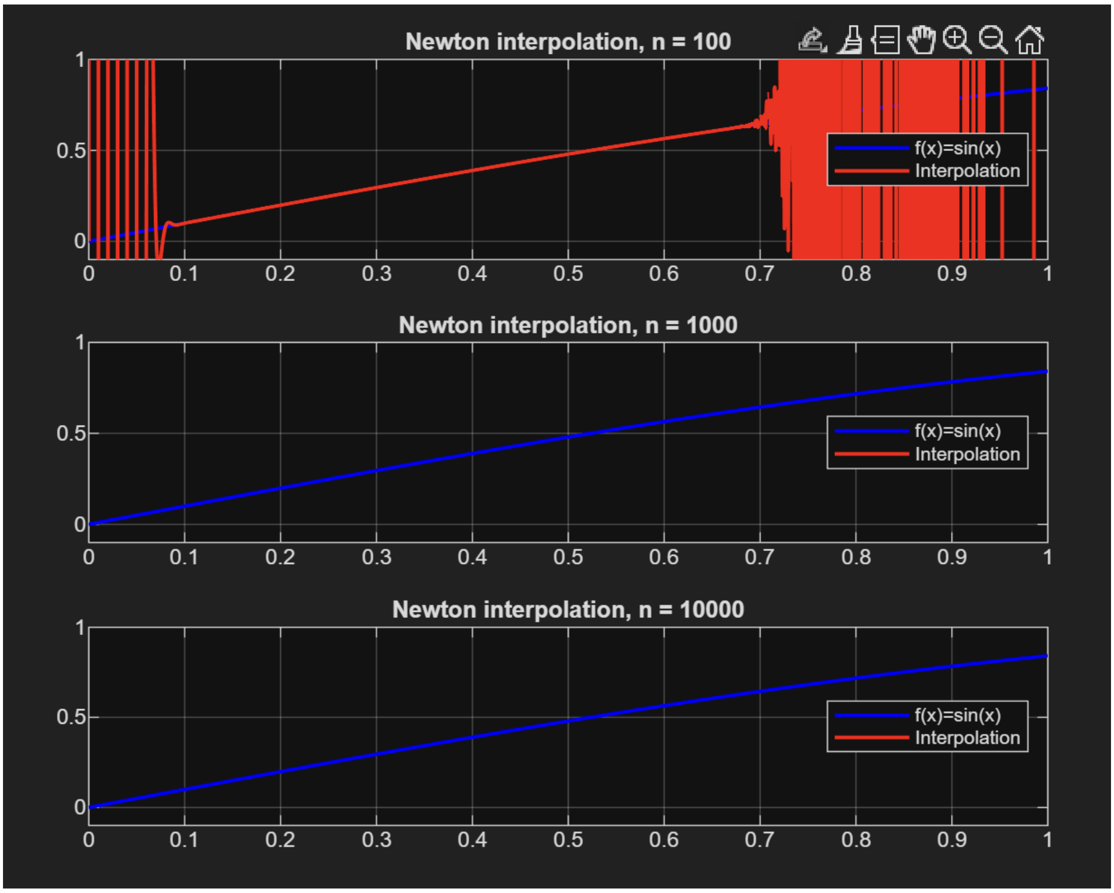
程式作業 2 Matlab 程式碼(Lagrange Interpolation n=10000 跑不出來)

```
clear; close all; clc;

f = @(x) sin(x);

a = 0; b = 1;
x_plot = linspace(a,b,2000);
y_true = f(x_plot);

n_values = [100, 1000];

figure;
for k = 1:length(n_values)
    n = n_values(k);
    x_nodes = linspace(a,b,n+1);
    y_nodes = f(x_nodes);

    y_interp = zeros(size(x_plot));
    for i = 1:length(x_nodes)
        l_i = ones(size(x_plot));
        for j = 1:length(x_nodes)
            if j ~= i
                l_i = l_i .* (x_plot - x_nodes(j)) / (x_nodes(i)-
x_nodes(j));
            end
        end
        y_interp = y_interp + y_nodes(i)*l_i;
    end


    subplot(3,1,k);
    plot(x_plot,y_true,'b-','LineWidth',1.5); hold on;
    plot(x_plot,y_interp,'r-','LineWidth',1);
    hold off;
    title(sprintf('Lagrange interpolation, n = %d', n));
    legend('f(x)=sin(x)','Interpolation','Nodes','Location','Best');
    grid on;
    axis([a b min(y_true)-0.05 max(y_true)+0.05]);
    axis([0 1 -0.1 1]);
end
```



Newton Divided Differences 插值(n=1000 以上無法計算)

```
clear; close all; clc;

f = @(x) sin(x);

a = 0; b = 1;
x_plot = linspace(a,b,2000);
y_true = f(x_plot);

n_values = [100, 1000, 10000];

figure;
for k = 1:length(n_values)
    n = n_values(k);
    x_nodes = linspace(a,b,n+1);
    y_nodes = f(x_nodes);

    n_nodes = n+1;
    dd = zeros(n_nodes, n_nodes);
    dd(:,1) = y_nodes(:);

    for j = 2:n_nodes
        for i = j:n_nodes
            dd(i,j) = (dd(i,j-1) - dd(i-1,j-1)) / (x_nodes(i) - x_nodes(i-
j+1));
        end
    end

    coeffs = diag(dd);

    y_interp = coeffs(end) * ones(size(x_plot));
    for j = n:-1:1
        y_interp = y_interp .* (x_plot - x_nodes(j)) + coeffs(j);
    end

    subplot(3,1,k);
    plot(x_plot,y_true,'b-','LineWidth',1.5); hold on;
    plot(x_plot,y_interp,'r-','LineWidth',1.5);
    hold off;

    title(sprintf('Newton interpolation, n = %d', n));
    legend('f(x)=sin(x)','Interpolation','Nodes','Location','Best');
    grid on;
    axis([0 1 -0.1 1]);
end
```


Newton interpolation, n = 100
f(x)=sin(x)
Interpolation
Newton interpolation, n = 1000
f(x)=sin(x)
Interpolation
Newton interpolation, n = 10000
f(x)=sin(x)
Interpolation