Problems

1) We want to construct a rational function of the form

$$R(z) = \frac{\alpha + \beta z}{1 + \gamma z}$$

that interpolates the data $(z_1, f_1), (z_2, f_2), (z_3, f_3)$ at distinct points z_1, z_2, and z_3. In other words, we seek α, β, and γ such that

$$R(z_j) = f_j, \quad j = 1, 2, 3.$$

Show how you can determine α, β, and γ by setting a linear system $Ax = b$ for the unknown vector $x = (\alpha, \beta, \gamma)^T$.

First, we need

$$R(z_i) = \frac{\alpha + \beta z_i}{1 + \gamma z_i} = f_i$$

which implies

$$\begin{aligned} \alpha + \beta z_i &= f_i + \gamma f_i z_i \\ \implies \alpha + \beta z_i - \gamma f_i z_i &= f_i \end{aligned}$$

for $i = 1, 2, 3$. We can rewrite this system as

$$\begin{pmatrix} 1 & z_1 & -f_1 z_1 \\ 1 & z_2 & -f_2 z_2 \\ 1 & z_3 & -f_3 z_3 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}.$$

Solving this system will give us the desired α, β, and γ.

2) We studied in class interpolation of functions defined in 1D. We can adapt the technique to higher dimensions. For instance, let

$$f(x, y) = e^x \sin(y).$$

We want to construct a polynomial of the form

$$p(x, y) = c_0 + c_1 x + c_2 y + c_3 xy + c_4 x^2 + c_5 y^2$$

that interpolates f at the points (x_i, y_i):

$$p(x_i, y_i) = f(x_i, y_i); \quad i = 1, \ldots, 5. \tag{1}$$

(i) Set up a linear system $Ac = f$ to determine the coefficients $c_0, \ldots, c_5$.

Using (1), we obtain the linear system:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 4 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 2 & 2 & 1 & 4 \\ 1 & 2 & 1 & 2 & 4 & 1 \\ 1 & 2 & 3 & 6 & 4 & 9 \end{pmatrix} \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{pmatrix} = \begin{pmatrix} e^0 \sin(0) \\ e^0 \sin(2) \\ e^1 \sin(0) \\ e^1 \sin(2) \\ e^2 \sin(1) \\ e^2 \sin(3) \end{pmatrix}.$$

(ii) Write a MATLAB code to determine c when the data points are

$$(0,0),(0,2),(1,0),(1,2),(2,1),(2,3)$$

Using my code which is attached to the end of the PDF I obtained c as

$$c = \begin{pmatrix} 0 \\ -9.491631052234917e-01 \\ 5.059193000070850e+00 \\ 7.812146225895686e-01 \\ 9.491631052234921e-01 \\ -2.302272143329005e+00 \end{pmatrix}.$$

(iii) Plot your polynomial p over $x \in [-1,3], y \in [-1,3]$. Compare this plot to the similar plot for f.

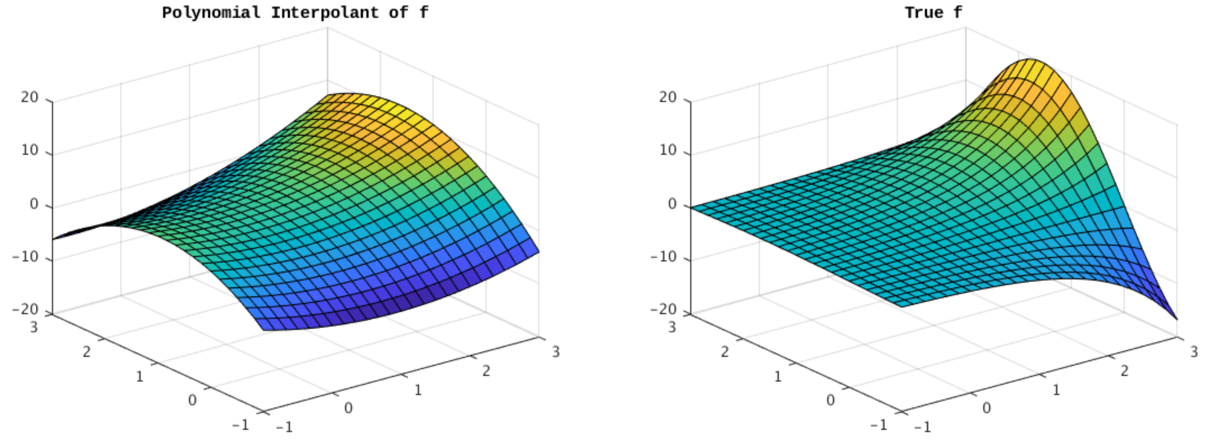

Figure 1: *Left*:Polynomial interpolant of $e^x\sin(y)$. *Right*: $e^x\sin(y)$

From Figure 1, we can see that our interpolant is similar in the middle of the plot. However, once we get near the boundaries of our plots, the accuracy of our interpolant degrades significantly (eyeball norm shows a difference). The degradation near the boundaries is due to the data not being given at the boundaries of our plot and so we are extrapolating. Likewise, the middle of the interpolant is relatively accurate because we have data points near the middle.

3) Recall the Lagrange basis functions $L_j(x)$ are defined by

$$L_j(x) = \prod_{i\neq j}\left(\frac{x - x_i}{x_j - x_i}\right)$$

for $\{x_i\}_{i=1}^n$.

(i) Prove that for any $n \geq 1$,

$$\sum_{j=0}^{n} L_i(x) = 1$$

for all $x \in \mathbb{R}$.
Proof:

Suppose we are creating an nth degree interpolating polynomial of the function $f(x) = 1$ over the data points $x_0, x_1, \ldots, x_n$ using Lagrange polynomials. Then our polynomial would be given by

$$p(x) = \sum_{j=0}^{n} (1) L_j(x) = \sum_{j=0}^{n} L_j(x).$$

Then, because $f(x) = 1$ is a 0th degree polynomial and because polynomial interpolants are unique for any given set of data points, $p(x) = 1$. Therefore

$$p(x) = \sum_{j=0}^{n} L_j(x) = 1.$$

□

(ii) Define $\Psi(x)K(x - x_0) \cdots (x - x_n)$. Show that the polynomial interpolant of degree n that interpolates the data $(x_i, f(x_i))$ for $i = 0, \ldots, n$ can be written in the form

$$p_n(x) = \sum_{i=0}^{n} \frac{\Psi(x)}{(x - x_i)\Psi'(x_i)} f(x_i)$$

provided $x \neq x_i$ for all i.

To show our equality, we first need to compute the first derivative of $\Psi(x)$ as

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}x}\Psi(x) &= \frac{\mathrm{d}}{\mathrm{d}x} \prod_{j=0}^{n} (x - x_j) \\ &= \sum_{k=0}^{n} \prod_{j \neq k} (x - x_j). \end{aligned}$$

Then, evaluating $\Psi'(x)$ at $x = x_i$ for $i = 0, \ldots, n$, we have

$$\begin{aligned} \Psi'(x_i) &= \sum_{k=0}^{n} \prod_{j \neq k} (x_i - x_j) \\ &= \prod_{i \neq j} (x_i - x_j) \quad \text{from the product rule.} \end{aligned}$$

Next, using the Lagrange basis and with $x \neq x_i$ for each i, we can form the nth degree interpolating polynomial of our data with

$$\begin{aligned}
p_n(x) &= \sum_{i=0}^{n} f(x_i)L_i(x) \\
&= \sum_{i=0}^{n} \frac{x - x_i}{x - x_i} f(x_i)L_i(x) \\
&= \sum_{i=0}^{n} \frac{x - x_i}{x - x_i} f(x_i) \prod_{i \neq j} \frac{x - x_j}{x_i - x_j} \\
&= \sum_{i=0}^{n} \left(\frac{1}{x - x_i} \prod_{i \neq j} \frac{1}{x_i - x_j} \right) \left((x - x_i) \prod_{i \neq j} (x - x_j) \right) f(x_i) \\
&= \sum_{i=0}^{n} \left(\frac{1}{(x - x_i)\Psi'(x_i)} \right) \left(\prod_{j=0}^{n} (x - x_j) \right) f(x_i) \\
&= \sum_{i=0}^{n} \left(\frac{1}{(x - x_i)\Psi'(x_i)} \right) (\Psi(x))\, f(x_i) \\
&= \sum_{i=0}^{n} \frac{\Psi(x)}{(x - x_i)\Psi'(x_i)} f(x_i).
\end{aligned}$$

(iii) Define next

$$w_i = \frac{1}{\Psi'(x_i)}.$$

Show that the polynomial interpolant of degree n that interpolates the data $(x_i, f(x_i))$ for $i = 0, \ldots, n$ can be written in the form

$$p_n(x) = \frac{\sum_{i=0}^n \frac{w_i f(x_i)}{x - x_i}}{\sum_{i=0}^n \frac{w_i}{x - x_i}}$$

provided $x \neq x_i$ for all i.

First, let's rewrite $\Psi(x)$ using our result from part (i):

$$\begin{aligned}
\Psi(x) &= \prod_{i=0}^n (x - x_i)/1 \\
&= \prod_{i=0}^n (x - x_i) \frac{1}{\sum_{j=0}^n L_j(x)} \\
&= \frac{1}{\prod_{i=0}^n \frac{1}{x - x_i}} \frac{1}{\sum_{j=0}^n L_j(x)} \\
&= \frac{1}{\sum_{j=0}^n \left(\prod_{i=0}^n \frac{1}{x - x_i} \right) L_j(x)} \\
&= \frac{1}{\sum_{j=0}^n \left(\prod_{i=0}^n \frac{1}{x - x_i} \right) \left(\prod_{k \neq j} \frac{x - x_k}{x_j - x_k} \right)} \\
&= \frac{1}{\sum_{j=0}^n \left(\prod_{i=0}^n \frac{1}{x - x_i} \right) \left(\prod_{k \neq j} (x - x_k) \right) w_j} \\
&= \frac{1}{\sum_{j=0}^n \frac{1}{x - x_j} w_j} \\
&= \frac{1}{\sum_{j=0}^n \frac{w_j}{x - x_j}}
\end{aligned}$$

Finally, using our results from the previous line and part (ii), we have

$$\begin{aligned}
p_n(x) &= \sum_{i=0}^n \frac{\Psi(x)}{(x - x_i)\Psi'(x_i)} f(x_i) \\
&= \sum_{i=0}^n w_i \frac{\Psi(x)}{(x - x_i)} f(x_i) \\
&= \Psi(x) \sum_{i=0}^n \frac{w_i f(x_i)}{(x - x_i)} \\
&= \frac{1}{\sum_{i=0}^n \frac{w_i}{x - x_i}} \sum_{i=0}^n \frac{w_i f(x_i)}{(x - x_i)} \\
&= \frac{\sum_{i=0}^n \frac{w_i f(x_i)}{x - x_i}}{\sum_{i=0}^n \frac{w_i}{x - x_i}}.
\end{aligned}$$

Code Used

```
%%
% 2D polynomial interpolation
% Author: Caleb Jacobs
% Date last modified: 11-10-2021

function Problem_2
    %% Settings
    format longE

    %% Setup
    f  = @(x) exp(x(:,1)) .* sin(x(:,2));      % Function to interpolate
    xi = [0, 0; ...
          0, 2; ...
          1, 0; ...
          1, 2; ...
          2, 1; ...
          2, 3];                               % Nodes
    n = length(xi);                            % Number of nodes

    %% Create system of equations
    % Create LHS matrix
    A = [ones(n,1), xi(:,1), xi(:,2), xi(:,1) .* xi(:,2), xi(:,1).^2, xi(:,2).^2];
    b = f(xi);                                 % RHS function vector

    %% Find polynomial coefficients
    c = A \ b

    %% Setup for plotting interpolation
    [X, Y] = meshgrid(linspace(-1, 3, 25), linspace(-1, 3, 25));
    F = @(x,y) exp(x) .* sin(y);               % Original function

    P = @(x,y) c(1) + c(2)*x + c(3)*y + c(4)*x.*y + c(5)*x.^2 + c(6)*y.^2;

    %% Plot surfaces
    figure(1), clf
    surf(X, Y, P(X,Y))
    title(’Polynomial Interpolant of f’)
    zlim([-20.1,20.1])

    figure(2), clf
    surf(X, Y, F(X,Y))
    title(’True f’)
    zlim([-20.1,20.1])

end
```