1 Univariate Barycentric Formulation

We have seen that Newton's interpolation polynomials appear to have an advantage in that they can be updated more efficiently than for instance Lagrange's interpolation polynomial. However, the Lagrange form can be written more efficiently in the so-called barycentric form, where the evaluation is faster.
Let us introduce the following quantities

$$\varphi(x) = \prod_{j=0}^{n}(x - x_j) \text{ and } w_i = \frac{1}{\prod_{j\neq i}(x_i - x_j)}.$$

Exercise 1

(a) Show that the value at x of the polynomial p can be written as

$$p(x) = \varphi(x)\sum_{i=0}^{n}\frac{w_i}{x - x_i}y_i.$$

The original Lagrange's interpolation polynomial is given as

$$p(x) = \sum_{i=0}^{n} y_i\varphi_i(x), where\, \varphi_i(x) = \prod_{j=0,\, j\neq i}^{n}\frac{x - x_j}{x_i - x_j} \Rightarrow p(x) = \sum_{i=0}^{n}\left(y_i\prod_{j=0,\, j\neq i}^{n}\frac{x - x_j}{x_i - x_j}\right)$$

The Barycentric form is given above. We will substitute phi(x) and wi into the formula as follows:

$$p(x) = \varphi(x)\sum_{i=0}^{n}\frac{w_i}{x - x_i}y_i = \prod_{j=0}^{n}(x - x_j)\sum_{i=0}^{n}\frac{\frac{1}{\prod_{j=0,\, j\neq i}^{n}(x_i - x_j)}}{x - x_i}y_i$$

$$\Rightarrow p(x) = \left(\prod_{j=0}^{n}(x - x_j)\right)\sum_{i=0}^{n}\left(y_i\frac{1}{(x - x_i)\prod_{j=0,\, j\neq i}^{n}(x_i - x_j)}\right) = \sum_{i=0}^{n}\left(y_i\frac{\prod_{j=0}^{n}(x - x_j)}{(x - x_i)\prod_{j=0,\, j\neq i}^{n}(x_i - x_j)}\right)$$

$$\Rightarrow p(x) = \sum_{i=0}^{n}\left(y_i\frac{(x - x_i)\prod_{j=0,\, j\neq i}^{n}(x - x_j)}{(x - x_i)\prod_{j=0,\, j\neq i}^{n}(x_i - x_j)}\right) = \sum_{i=0}^{n}\left(y_i\frac{\prod_{j=0,\, j\neq i}^{n}(x - x_j)}{\prod_{j=0,\, j\neq i}^{n}(x_i - x_j)}\right) = \sum_{i=0}^{n}\left(y_i\prod_{j=0,\, j\neq i}^{n}\left(\frac{x - x_j}{x_i - x_j}\right)\right)$$

$\Rightarrow p(x) = \sum_{i=0}^{n} y_i\varphi_i(x)$, equal to the original Lagrange's interpolation polynomial and value at x of the polynomial p can be written as the formula above.

(b) Derive the barycentric formula

$$p(x) = \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i \right) \Big/ \left(\sum_{i=0}^{n} \frac{w_i}{x - x_i} \right).$$

Lagrange's interpolation polynomial

$$p(x) = \sum_{i=0}^{n} y_i \varphi_i(x), where\, \varphi_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \Rightarrow p(x) = \sum_{i=0}^{n} \left(y_i \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j} \right)$$

$$\Rightarrow p(x) = \sum_{i=0}^{n} \left(y_i \left(\frac{\prod_{j=0}^{n} x - x_j}{x - x_i} \right) \left(\prod_{j=0, j\neq i}^{n} \frac{1}{x_i - x_j} \right) \right) = \sum_{i=0}^{n} \left(y_i \left(\frac{\prod_{j=0}^{n} x - x_j}{x - x_i} \right) \left(\frac{1}{\prod_{j=0, j\neq i}^{n} x_i - x_j} \right) \right)$$

and we have $w_i = \dfrac{1}{\prod_{j=0, j\neq i}^{n} (x_i - x_j)}$ $\Rightarrow p(x) = \sum_{i=0}^{n} \left(y_i \left(\frac{\prod_{j=0}^{n} x - x_j}{x - x_i} \right) w_i \right) = \sum_{i=0}^{n} \left(\left(\prod_{j=0}^{n} x - x_j \right) \frac{w_i}{x - x_i} y_i \right)$

We can see that $\prod_{j=0}^{n} x - x_j$ is independent of i and can be treated as a constant, so we can put it outside of the sum as follows:

$$\Rightarrow p(x) = \left(\prod_{j=0}^{n} x - x_j \right) \left(\sum_{i=0}^{n} \left(\frac{w_i}{x - x_i} y_i \right) \right)$$

Now we can try adding interpolation with the constant function 1 besides the original data. In other words, the interpolant of the constant function is itself: p(x) = yi => Both sides can be divided by p(x)

$$\Rightarrow 1 = \left(\prod_{j=0}^{n} x - x_j \right) \left(\sum_{i=0}^{n} \left(\frac{w_i}{x - x_i} \times 1 \right) \right) \Rightarrow \prod_{j=0}^{n} x - x_j = \frac{1}{\sum_{i=0}^{n} \left(\frac{w_i}{x - x_i} \right)}$$

$$\Rightarrow p(x) = \left(\frac{1}{\sum_{i=0}^{n} \left(\frac{w_i}{x - x_i} \right)} \right) \left(\sum_{i=0}^{n} \left(\frac{w_i}{x - x_i} y_i \right) \right) = \frac{\sum_{i=0}^{n} \frac{w_i}{x - x_i} y_i}{\sum_{i=0}^{n} \frac{w_i}{x - x_i}}$$

This is the derivation of the barycentric formula of the Lagrange's interpolation polynomial

(c) Show that the updated weights w_k can be computed in $O(n)$ arithmetic operations after each added point x_{n+1}.

The formula of wk is given as: $w_k = \dfrac{1}{\prod\limits_{j=0,\, j\neq k}^{n} \left(x_k - x_j\right)}$

For each added point x_{n+1}, the weights wk can be updated in a for-loop of range n

$w_{n+1} = 1$

for k in range[0, n]:

$$w_k = \frac{w_k}{x_k - x_{n+1}}$$

$$w_{n+1} = \frac{w_{n+1}}{\left(x_{n+1} - x_k\right)}$$

end

=> The updated weights wk can be computed in O(n) arithmetic operations after adding new point

EXERCISE 2

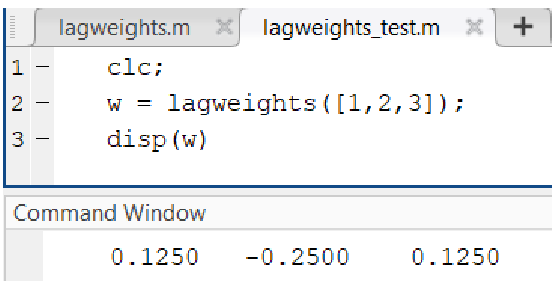
(a) Write a function lagweights.m that computes the weights w_k for the nodes x_k.

The Matlab code for lagweights.m is

```
% Compute the weights of the barycentric formula
function [w] = lagweights(x)
    w = ones([1,length(x)]);
    for k = 1:length(x)
        for j = 1:length(x)
            if k ~= j
                w(k) = w(k) * 1/(x(k) - x(j));
            end
        end
    end
end
```

The Matlab code for lagweights_test.m is

```
clc;
w = lagweights([1,2,3]);
disp(w)
```



(b) Write a function specialsum.m that computes the quantity $\sum_{i=0}^{n} \frac{z_i}{t-x_i}$, when x and z are arrays of size n and t is an array of size s. The output has to be an array of size s. That is, t has the values where the interpolation polynomial is evaluated.

The Matlab code for specialsum.m is

```
%{
x and z are arrays of size n and t is an array of size s. The
output has to be an array of size s. That is, t has the values where
the interpolation polynomial is evaluated
%}
function [sum] = specialsum(x,z,t)
    sum = zeros([1,length(t)]);
    for i = 1:length(x)
        sum = sum + z(i)./(t - x(i));
    end
end
```

The Matlab code for specialsum_test.m is

```
clc;
S = specialsum((0:3),(1.5:0.5:3), -4:0);
disp(S)
```

```
specialsum.m    specialsum_test.m    +
1 -   clc;
2 -   S = specialsum((0:3),(1.5:0.5:3), -4:0);
3 -   disp(S)
```

```
Command Window
   -1.6202   -2.0000   -2.6417   -4.0833       Inf
```

(c) Write a program lagpolint.m that computes the barycentric form of p at points t.

The Matlab code for lagpolint.m is

```
% computes the barycentric form of p at points t
function [P] = lagpolint(X, T, fun)
    Y=fun(X);
    W=lagweights(X);
    S1=specialsum(X, W.*Y, T);
    S2=specialsum(X, W, T);
    P=S1./S2;
    plot(X,Y,'.b',T,P,'r')
end
```

(d) Test lagpolint.m by sampling from the function $y = \sqrt{|t|}$ on $[-1, 1]$. Try first 9 uniform points and then 101 Chebyshev points

$$x_j = -\cos(j\pi/n), \quad j = 0, 1, \ldots, 100 := n.$$

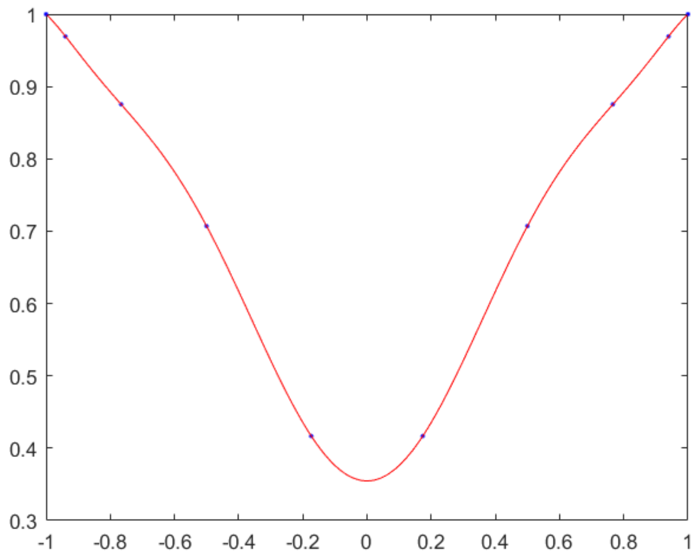
Plot the polynomials.

The Matlab code for lagpolint_test.m is

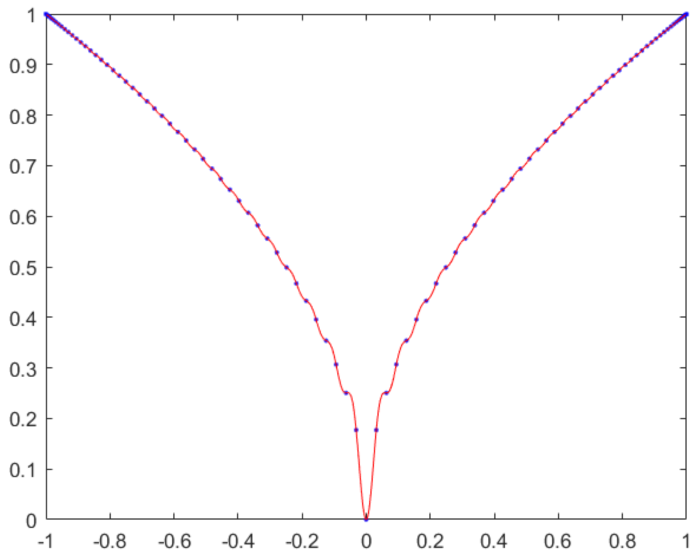
```
%{ Test lagpolint.m by sampling from the function y = sqrt(|t|) on
[?1; 1]. Try first 9 uniform points and then 101 Chebyshev points
xj = -cos(j*pi/n); j = 0; 1;...; 100 := n: Plot the polynomials. %}
clc;
fun = @(t)sqrt(abs(t)); % y = sqrt(|t|)
T=-1:0.002:1;
% 9 Chebyshev points
figure(1)
j=0:9;
X = -cos(j.*pi/9);
P = lagpolint(X, T, fun); disp(P)
% 101 Chebyshev points
figure(2)
j=0:100;
X = -cos(j.*pi/100);
P = lagpolint(X, T, fun); disp(P)
```

- Plotting the polynomial

9 uniform Chebyshev points



100 uniform Chebyshev points



First values of interpolation polynomial P

Command Window

```
  Columns 1 through 14

       NaN    0.9991    0.9983    0.9974    0.9965    0.9956    0.9947    0.9937    0.9928    0.9918    0.9908    0.9898    0.9888    0.9878

  Columns 15 through 28

    0.9868    0.9858    0.9847    0.9837    0.9826    0.9816    0.9805    0.9794    0.9784    0.9773    0.9762    0.9751    0.9740    0.9729

  Columns 29 through 42

    0.9718    0.9707    0.9695    0.9684    0.9673    0.9662    0.9651    0.9639    0.9628    0.9617    0.9605    0.9594    0.9582    0.9571
```

2 Application to Interpolating Surfaces

Let us next consider a regular grid of points (x_i, y_j) and the surface values $z_{ij} = f(x_i, y_j)$. Let $\mathbf{x} = (x_0, \ldots, x_m) \in \mathbb{R}^{m+1}$ and $\mathbf{y} = (y_0, \ldots, y_n) \in \mathbb{R}^{n+1}$. In this setting the surface can be interpolated using a product (tensor product) of univariate interpolation polynomials. In the sequel we denote the y-dependent quantities with a bar, for instance, $\bar{l}_q(t)$ for the corresponding Lagrange basis polynomial in the y-direction.

EXERCISE 3

(a) Show that $P(s,t) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} l_p(s) \bar{l}_q(t)$ is an interpolation polynomial.

$$P(s,t) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} \ell_p(s) \bar{\ell}_q(t)$$

We have the following identities:

$$\ell(x) = \prod_{a=0}^{m} (x - x_a), w_p = \frac{1}{\prod_{u=0, u\neq p}^{m} (x_p - x_u)}$$

$$\bar{\ell}(y) = \prod_{b=0}^{n} (y - y_b), w_q = \frac{1}{\prod_{v=0, v\neq q}^{n} (y_q - y_v)}$$

$$\text{and } \ell_p(x) = \ell(x) \frac{w_p}{x - x_p}, \ \bar{\ell}_q(y) = \bar{\ell}(y) \frac{w_q}{y - y_q}$$

$$\Rightarrow P(x,y) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} \ell_p(x) \bar{\ell}_q(y) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} \ell(x) \frac{w_p}{x - x_p} \bar{\ell}(y) \frac{w_q}{y - y_q}$$

$$\Rightarrow P(x,y) = \sum_{p=0}^{m} \sum_{q=0}^{n} z_{pq} \frac{\prod_{a=0}^{m} (x - x_a)}{(x - x_p) \prod_{u=0, u\neq p}^{m} (x_p - x_u)} \frac{\prod_{b=0}^{n} (y - y_b)}{(y - y_q) \prod_{v=0, v\neq q}^{n} (y_q - y_v)} \Rightarrow$$

$$P(x,y)=\sum_{p=0}^{m}\sum_{q=0}^{n} z_{pq}\frac{\prod_{a=0,a\neq p}^{m}(x-x_a)}{\prod_{u=0,u\neq p}^{m}(x_p-x_u)}\frac{\prod_{b=0,b\neq q}^{n}(y-y_b)}{\prod_{v=0,v\neq q}^{n}(y_q-y_v)}=\sum_{p=0}^{m}\sum_{q=0}^{n}\left(z_{pq}\prod_{u=0,u\neq p}^{m}\left(\frac{x-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y-y_v}{y_q-y_v}\right)\right)$$

If P(x,y) is the interpolation polynomial, if we insert x_p, y_q into P, we should get z_{pq}

Analysis of $P(x,y)=\sum_{p=0}^{m}\sum_{q=0}^{n}\left(z_{pq}\prod_{u=0,u\neq p}^{m}\left(\frac{x-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y-y_v}{y_q-y_v}\right)\right)$

1. When x = xp, y = yq => $z_{pq}\prod_{u=0,u\neq p}^{m}\left(\frac{x_p-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y_q-y_v}{y_q-y_v}\right)=z_{pq}\times 1\times 1=z_{pq}$

2. When x = xj, j != p and y = yi, i can be anything in [0, n] => $z_{ji}\prod_{u=0,u\neq p}^{m}\left(\frac{x_j-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y_i-y_v}{y_q-y_v}\right)$

- In the identity $\prod_{u=0,u\neq p}^{m}\left(\frac{x_j-x_u}{x_p-x_u}\right)$, we know that j != p, and also u != p => There exists u such that u = j. In other words, there exists xj – xu = 0. A product of xj -xu thus will then be reduced to 0

=> $z_{ji}\prod_{u=0,u\neq p}^{m}\left(\frac{x_j-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y_i-y_v}{y_q-y_v}\right)=z_{ji}\times 0\times\prod_{v=0,v\neq q}^{n}\left(\frac{y_i-y_v}{y_q-y_v}\right)=0$

3. When x = xi, i can be anything in [0, m] and y = yk, k != q => $z_{ik}\prod_{u=0,u\neq p}^{m}\left(\frac{x_i-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y_k-y_v}{y_q-y_v}\right)$

- In the identity $\prod_{v=0,v\neq q}^{n}\left(\frac{y_k-y_v}{y_q-y_v}\right)$, we know that k != q, and also v != q => There exists v such that v = k. In other words, there exists yk – yv = 0. A product of yk - yv thus will then be reduced to 0

=> $z_{ik}\prod_{u=0,u\neq p}^{m}\left(\frac{x_i-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y_k-y_v}{y_q-y_v}\right)=z_{ik}\times\prod_{u=0,u\neq p}^{m}\left(\frac{x_i-x_u}{x_p-x_u}\right)\times 0=0$

Now we plut x_p, y_q into P(x,y)

$$P(x_p,y_q)=\sum_{p=0}^{m}\sum_{q=0}^{n}\left(z_{pq}\prod_{u=0,u\neq p}^{m}\left(\frac{x-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y-y_v}{y_q-y_v}\right)\right)=z_{pq}+0+.....+0,\text{number of 0s is (m x n) - 1}$$

=> $P(x_p,y_q)=\sum_{p=0}^{m}\sum_{q=0}^{n}\left(z_{pq}\prod_{u=0,u\neq p}^{m}\left(\frac{x-x_u}{x_p-x_u}\right)\prod_{v=0,v\neq q}^{n}\left(\frac{y-y_v}{y_q-y_v}\right)\right)=z_{pq}$

Since we insert x_p, y_q into P(x,y) and we get z_{pq}, P(x,y) or P(s,t) is the interpolation polynomial

(b) Show that

$$P(s,t) = \left(\sum_{p=0}^{m} \sum_{q=0}^{n} \frac{w_p \bar{w}_q}{(s-x_p)(t-y_q)} z_{pq} \right) \Big/ \left(\sum_{p=0}^{m} \frac{w_p}{s-x_p} \sum_{q=0}^{n} \frac{\bar{w}_q}{t-y_q} \right).$$

It is known in part (a) that

$$w_p = \frac{1}{\prod_{u=0,u\neq p}^{m} \left(x_p - x_u\right)} \text{ and } \bar{w}_q = \frac{1}{\prod_{v=0,v\neq q}^{n} \left(y_q - y_v\right)}$$

$$\text{and } P(x,y) = \sum_{p=0}^{m} \sum_{q=0}^{n} \left(z_{pq} \prod_{u=0,u\neq p}^{m} \left(\frac{x-x_u}{x_p - x_u} \right) \prod_{v=0,v\neq q}^{n} \left(\frac{y-y_v}{y_q - y_v} \right) \right)$$

First, we need to simplify the expression

$$\prod_{u=0,u\neq p}^{m} \left(\frac{x-x_u}{x_p-x_u} \right) = \left(\frac{\prod_{u=0}^{m} x - x_u}{x - x_p} \right) \left(\prod_{u=0,u\neq p}^{m} \frac{1}{x_p - x_u} \right) = \left(\frac{\prod_{u=0}^{m} x - x_u}{x - x_p} \right) \left(\frac{1}{\prod_{u=0,u\neq p}^{m} x_p - x_u} \right) = \left(\frac{\prod_{u=0}^{m} x - x_u}{x - x_p} \right) w_p$$

$$\prod_{v=0,v\neq q}^{n} \left(\frac{y-y_v}{y_q-y_v} \right) = \left(\frac{\prod_{v=0}^{n} y - y_v}{y - y_q} \right) \left(\prod_{v=0,v\neq q}^{n} \frac{1}{y_q - y_v} \right) = \left(\frac{\prod_{v=0}^{n} y - y_v}{y - y_q} \right) \left(\frac{1}{\prod_{v=0,v\neq q}^{n} y_q - y_v} \right) = \left(\frac{\prod_{v=0}^{n} y - y_v}{y - y_q} \right) \bar{w}_q$$

P(x,y) can be derived as follows

$$P(x,y) = \sum_{p=0}^{m} \sum_{q=0}^{n} \left(z_{pq} \left(\frac{\prod_{u=0}^{m} x - x_u}{x - x_p} \right) w_p \left(\frac{\prod_{v=0}^{n} y - y_v}{y - y_q} \right) \bar{w}_q \right) = \prod_{u=0}^{m} (x - x_u) \prod_{v=0}^{n} (y - y_v) \sum_{p=0}^{m} \sum_{q=0}^{n} \left(\frac{w_p \bar{w}_q}{\left(x - x_p\right)\left(y - y_q\right)} z_{pq} \right)$$

Now we can try adding interpolation with the constant function 1 besides the original data. The interpolant of the constant function is itself: p(x,y) = z_xy => Both sides can be divided by p(x,y)

$$\Rightarrow 1 = \prod_{u=0}^{m} (x - x_u) \prod_{v=0}^{n} (y - y_v) \sum_{p=0}^{m} \sum_{q=0}^{n} \left(\frac{w_p \bar{w}_q}{\left(x - x_p\right)\left(y - y_q\right)} \times 1 \right)$$

$$\Rightarrow \prod_{u=0}^{m} (x - x_u) \prod_{v=0}^{n} (y - y_v) = \frac{1}{\sum_{p=0}^{m} \sum_{q=0}^{n} \left(\frac{w_p \bar{w}_q}{\left(x - x_p\right)\left(y - y_q\right)} \right)} = \frac{1}{\sum_{p=0}^{m} \sum_{q=0}^{n} \frac{w_p}{x - x_p} \frac{\bar{w}_q}{y - y_q}}$$

$$\Rightarrow \prod_{u=0}^{m} (x - x_u) \prod_{v=0}^{n} (y - y_v) = \frac{1}{\sum_{p=0}^{m} \frac{w_p}{x - x_p} \sum_{q=0}^{n} \frac{\bar{w}_q}{y - y_q}}$$

Plugging back into P(x,y), we have the identity

$$P(x,y)=\frac{1}{\sum_{p=0}^{m}\frac{w_p}{x-x_p}\sum_{q=0}^{n}\frac{\overline{w_q}}{y-y_q}}\sum_{p=0}^{m}\sum_{q=0}^{n}\left(\frac{w_p\overline{w_q}}{(x-x_p)(y-y_q)}z_{pq}\right)=\frac{\sum_{p=0}^{m}\sum_{q=0}^{n}\left(\frac{w_p\overline{w_q}}{(x-x_p)(y-y_q)}z_{pq}\right)}{\sum_{p=0}^{m}\frac{w_p}{x-x_p}\sum_{q=0}^{n}\frac{\overline{w_q}}{y-y_q}}$$

(proven)

This is a generalization of the Lagrange's barycentric formula to the 3-dimensional case

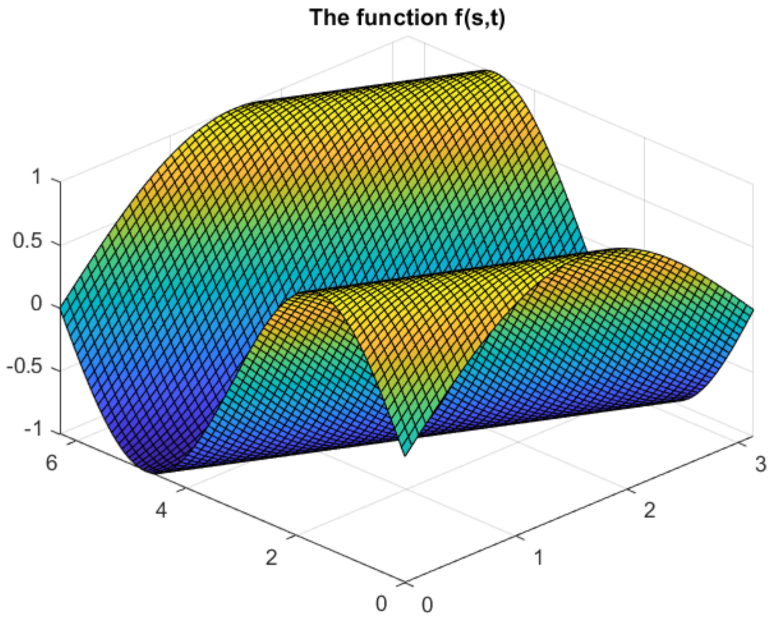
EXERCISE 4 Write a program interpolsurf.m such that given the grid points as x and y, and the sampled values $\mathbf{z} = (f(x_i, y_j)) \in \mathbb{R}^{(m+1)\times(n+1)}$, computes the values of the polynomial $P(s,t)$.
Test with $f(s,t) = \sin(s+t)$, $m = 3$, x a uniform partition of $[0,\pi]$, and $n = 7$, y a uniform partition of $[0, 2\pi]$. Plot $P(s,t)$, $f(s,t)$, and the difference $f - P$.

The Matlab code for interpolsurf.m is

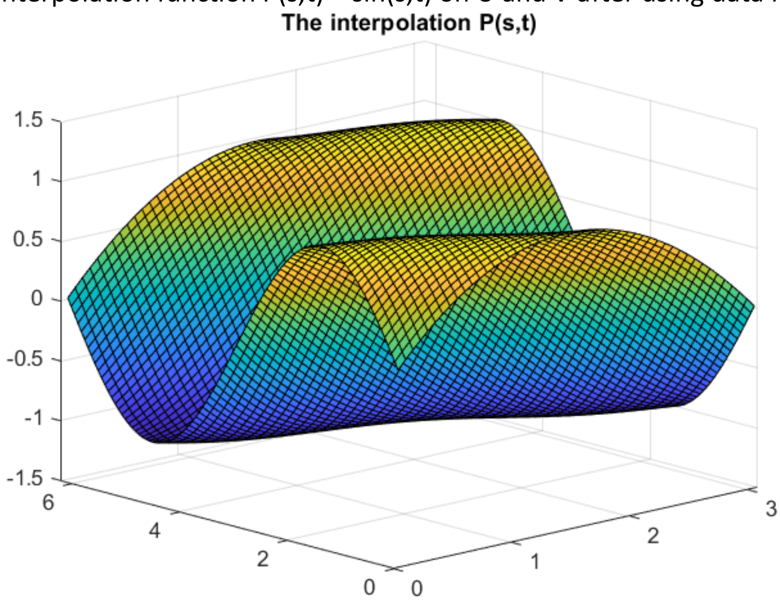
```
%{
Write a program interpolsurf.m such that given the grid
points as x and y, and the sampled values z = (f(xi; yj)) of
R(m+1)x(n+1), computes the values of the polynomial P(s,t).
Test with f(s,t) = sin(s + t), m = 3, x a uniform partition of
[0; pi],and n = 7, y a uniform partition of [0; 2pi]. Plot P(s,t),
f(s,t), and the difference f - P
The interpolation grid is given as U x V, where size(U) = 51 is
uniform partition on [0; pi] and size(V) = 101 is uniform partition
on [0; 2pi]
%}
clc;
f = @(X,Y) sin(X + Y);
m = 3; n = 7;
X = (0:1/m:1) * pi;
Y = (0:1/n:1) * 2 * pi;
[XX,YY] = meshgrid(X,Y);
ZZ = f(XX,YY);
U = (0:0.02:1) * pi;
V = (0:0.01:1) * 2 * pi;
W1 = lagweights(X);
W2 = lagweights(Y);

% numerator
Suv = zeros([length(V) length(U)]);
for i = 1:length(V)
    for j = 1:length(U)
        for q = 1:length(Y)
            for p = 1:length(X)
                Suv(i,j) = Suv(i,j) + (W1(p) * W2(q) * ZZ(q,
p))/((U(j) - X(p))*(V(i) - Y(q)));
            end
        end
    end
end
```

```
% denominator
SX = specialsum(X, W1, U);
SY = specialsum(Y, W2, V);
[SSX,SSY] = meshgrid(SX,SY);
Pst = Suv./(SSX .* SSY);
[UU,VV] = meshgrid(U,V);
% Plotting f(UU,VV) or f(s,t)
fst = f(UU,VV);
figure(1)
surf(UU,VV,fst)
title("The function f(s,t)")
figure(2)
surf(UU,VV,Pst)
title("The interpolation P(s,t)")
figure(3)
surf(UU,VV, Sf - Pst)
title("The difference f - P")
```

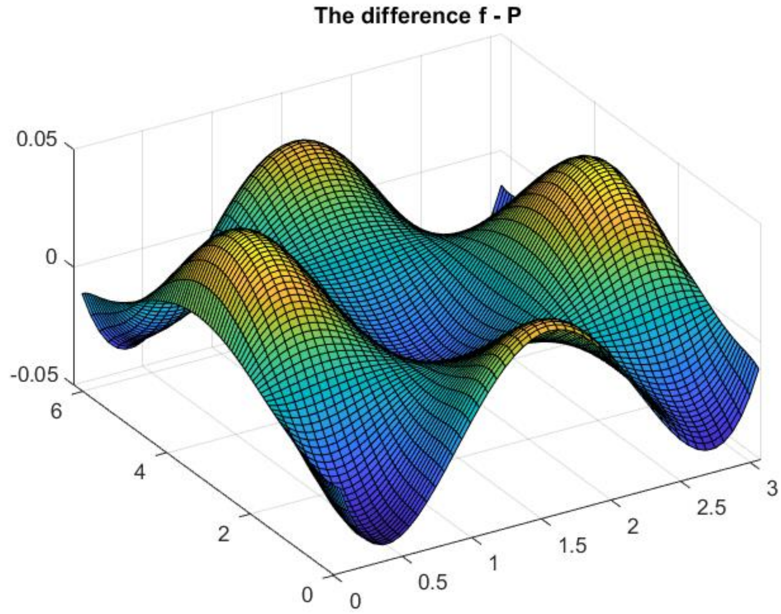
The original function f(s,t) = sin(s,t) on U and V



The interpolation function P(s,t) = sin(s,t) on U and V after using data from X and Y



The difference between the original function f(s,t) and the interpolation function P(s,t)



We can observe that the errors is quite significant, varying from near -0.05 to near 0.05. To reduce the errors, we can add more datapoints as X,Y to increase the accuracy of the interpolation surface.