Christopher Lum
lum@uw.edu

Lecture 04c
3D Plotting

Lecture is on YouTube

The YouTube video(s) covering this lecture are located at:

-3D Plotting in Matlab (https://youtu.be/OUwfE_-tcfo)

-3D Plotting in Mathematica (https://youtu.be/s_ehZc5N7Lg)

Outline

-Mathematical Theory

-3D Plots in Mathematica

-3D Plots in Matlab

Mathematical Theory

Consider the function

$$f(x_1, x_2) = \cos(x_1)\, x_2$$

The function f accepts two, separate inputs (x_1 and x_2) and produces a single, scalar output.

We can visualize f by plotting it over a range of x_1 and x_2 values. This will result in a 3D plot where the

"x-axis" represents the values of x_1
"y-axis" represents the values of x_2
"z-axis" represents the corresponding value of f at a location (x_1, x_2)

3D Plots in Mathematica

In order to plot this in Mathematica, we first define the function

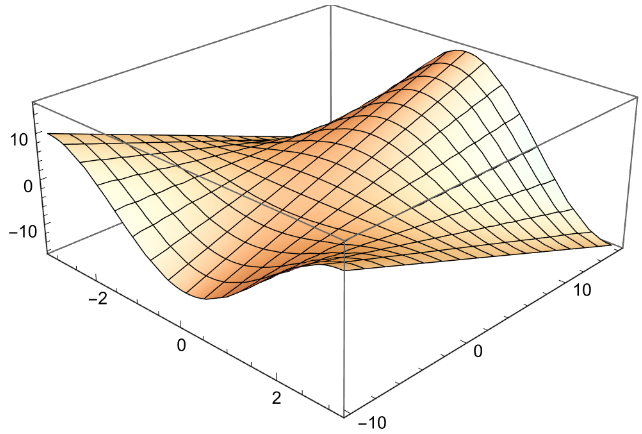
```
f[x1_, x2_] = Cos[x1] x2;
```

We can then use Mathematica's 'Plot3D' function to visualize this. For example, we can consider the domain of

$$x_1 \in [-\pi, \pi]$$

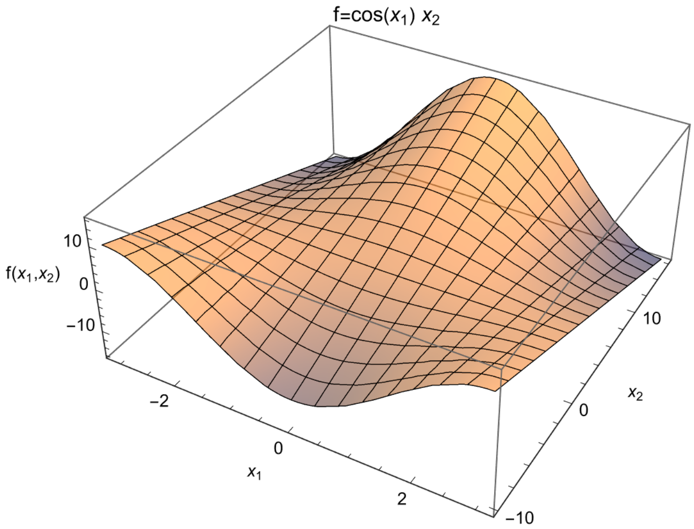
$x_2 \in [-10, 16]$

```
Plot3D[f[x1, x2], {x1, -π, π}, {x2, -10, 16}]
```



Just like any other Mathematica plot, we can add options to add features to the plot

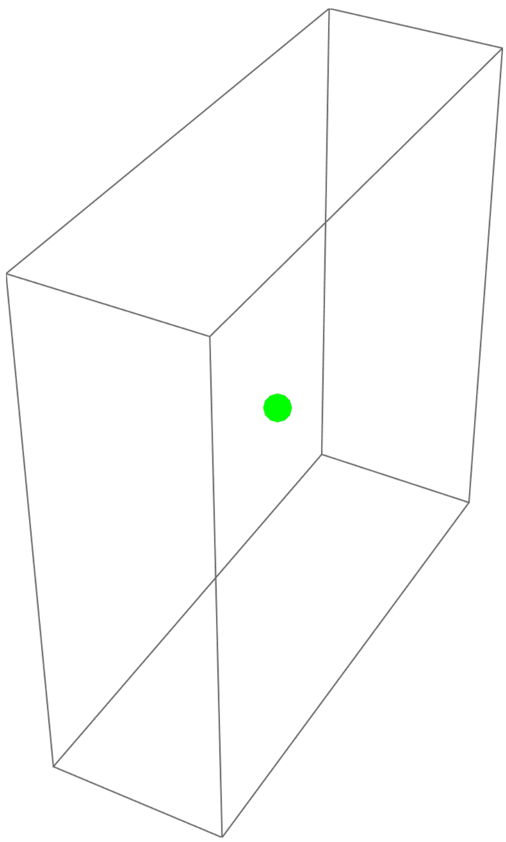
```
Plot3D[f[x1, x2], {x1, -π, π}, {x2, -10, 16},
 (*Plot options*)
 AxesLabel → {"x₁", "x₂", "f(x₁,x₂)"},
 PlotLabel → "f=cos(x₁) x₂",
 ColorFunction → "RustTones",
 PlotStyle → Opacity[0.5]]
```



We may want to add single points to this plot. We can plot single points in 3D in Mathematica using ‘Graphics3D’

```
(*Define the point of interest*)
x1o = 0;
x2o = 3;
P = {x1o, x2o, f[x1o, x2o]};

(*Plot using Graphics3D*)
plotPoint = Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{P[[1]], P[[2]], P[[3]]}]
  }
 ]
```



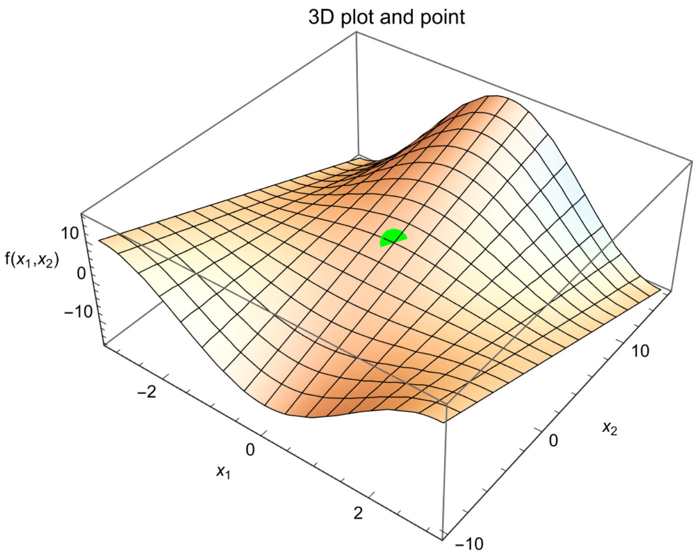
We can simultaneously display two plots using ‘Show’

```
Show[
 (*Plot the surface*)
 Plot3D[f[x1, x2], {x1, -π, π}, {x2, -10, 16}],

 (*Plot the point*)
 plotPoint,

 (*Plot options*)
 PlotLabel → "3D plot and point",
 AxesLabel → {"x1", "x2", "f(x1,x2)"}

]
```



3D Plots in Matlab

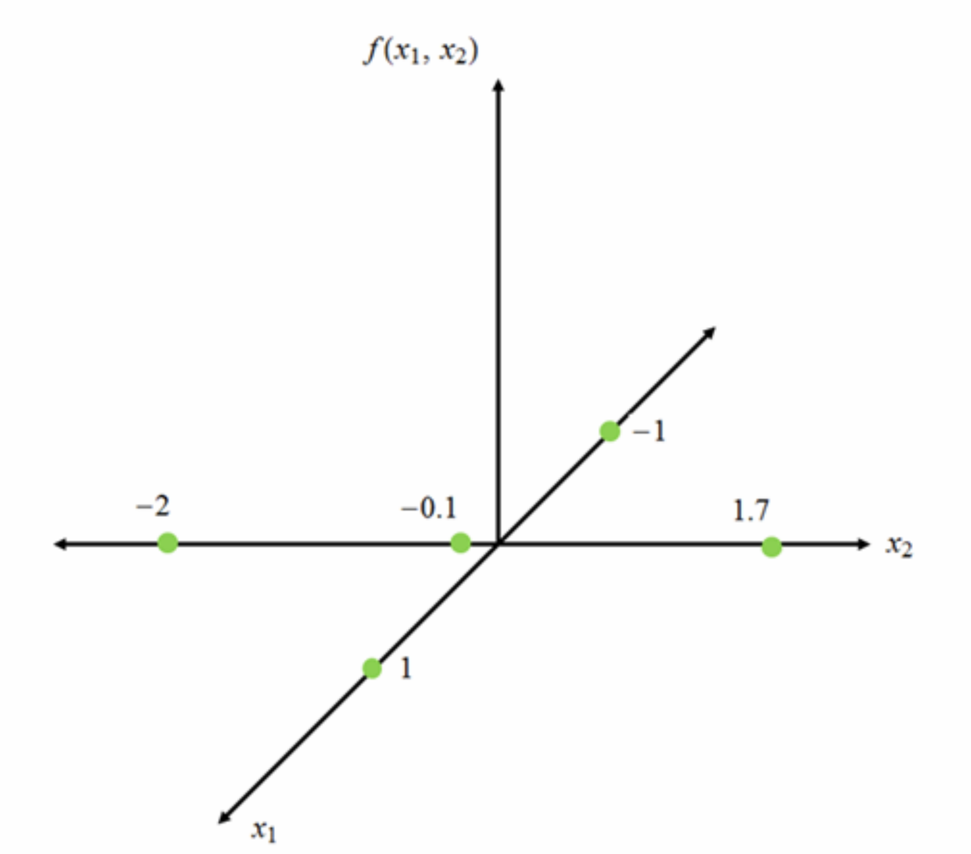
We consider the same function of two variables, $f(x_1, x_2)$. In order to plot this function, we need to plot its values at various values of x_1 and x_2. We can do this using Matlab in several steps.

Step 1: Define vectors of x_1 and x_2

Define the range of x_1 and x_2 points that we would like to evaluate the function at. For example, suppose that we are interested in plotting the function at values of

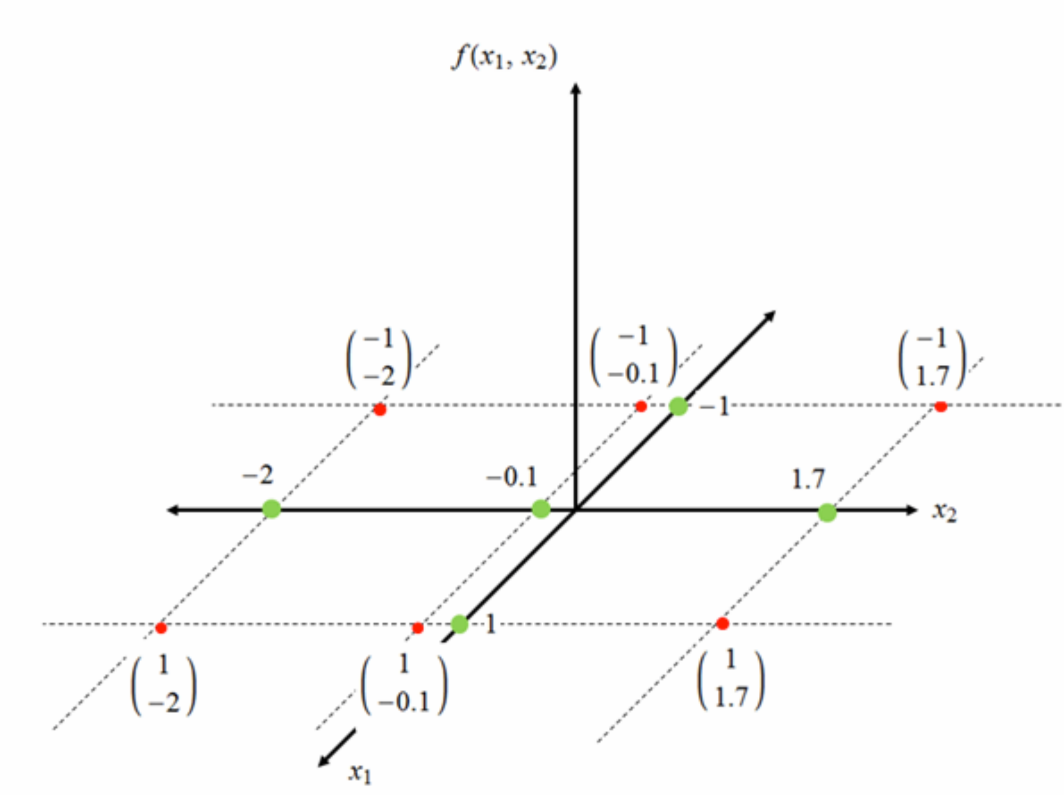
$$x_1 \in \{-1, 1\}$$
$$x_2 \in \{-2, -0.1, 1.7\}$$

We can visualize this as the green dots as shown below.



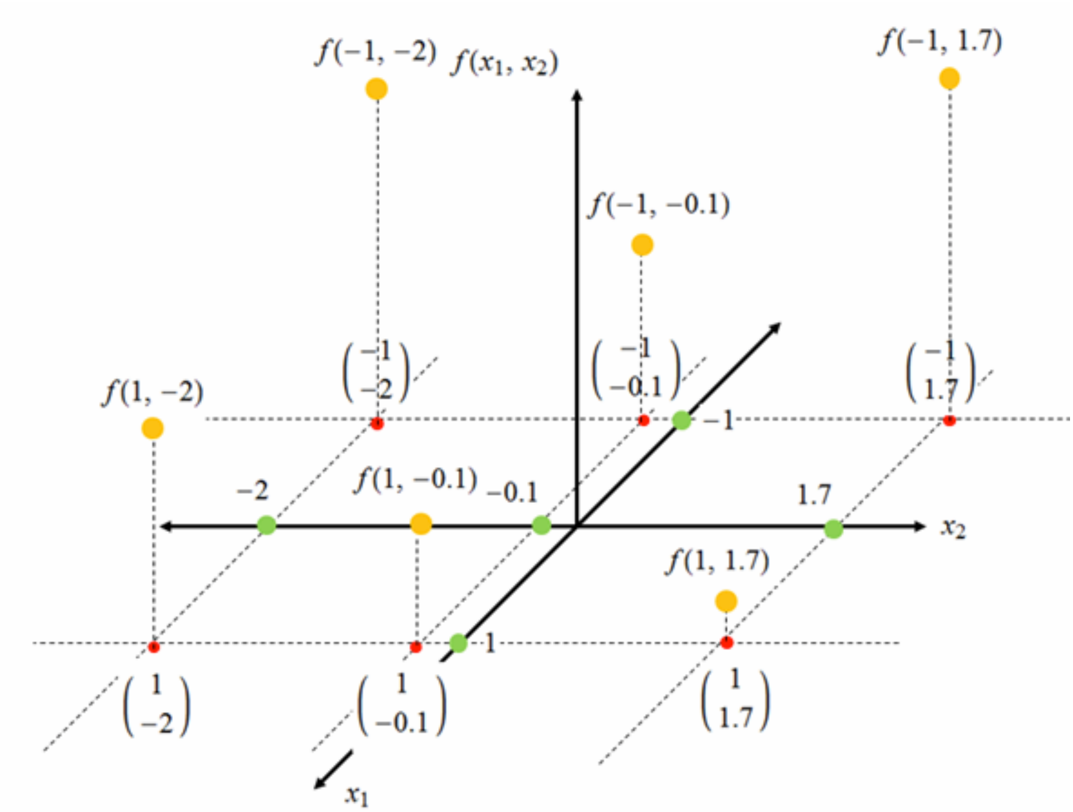
Step 2: Create mesh of points

We now can create a mesh of 2D points which represent the Cartesian product of these two vectors. These points are shown below in red dots



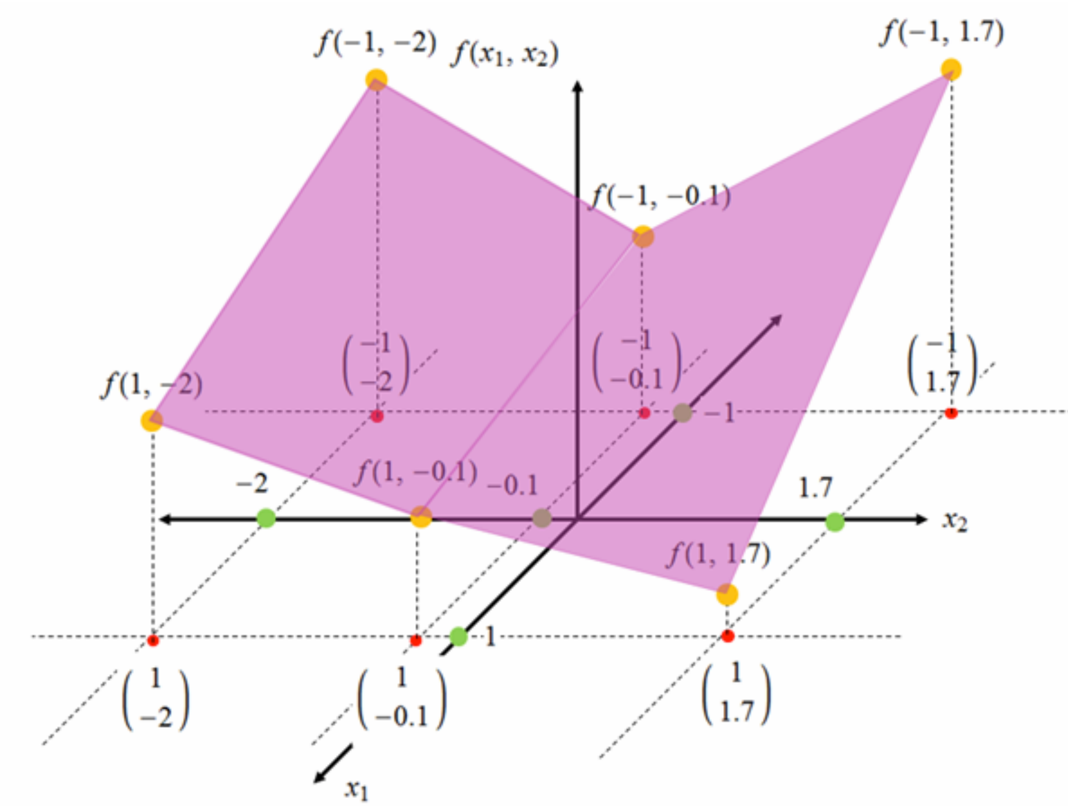
Step 3: Evaluate function at meshed points

We can evaluate the function at each of these meshed points (the red dots). The value of the function evaluated at each red dot is shown below as orange dots.



Step 4: Fill in vertices with polygon

Finally, we can connect the points of $f(x_1, x_2)$ (orange dots) with a polygon to attempt to see what the function looks like. This is shown below as the magenta polygon



Matlab provides various functions for these steps. Some useful ones are

> linspace (create a linear vector of points in the x_1 and x_2 direction)
> meshgrid (create a 2D grid of points)
> surf (connect points with polygons)

Recall that the function we were interested in was

$$z = f(x_1, x_2) = \cos(x_1)\, x_2$$

with $x_1 \in [-\pi, \pi]$
$x_2 \in [-10, 16]$

The relevant Matlab code to plot this is shown below

```
clear
clc
close all

%Define the range of x1 values
x1_start     = -pi;       %Starting value of the x1 vector
x1_end       = pi;        %Ending value of the x1 vector
Nx1          = 50;        %Number of points in the x1 vector
x1           = linspace(x1_start, x1_end, Nx1);

%Define the range of x2 values
x2_start     = -10;       %Starting value of the x2 vector
x2_end       = 20;        %Ending value of the x2 vector
Nx2          = 80;        %Number of points in the x2 vector
x2           = linspace(x2_start, x2_end, Nx2);

%Create 2D matrices of the x1 and x2 vectors
[X1,X2] = meshgrid(x1,x2);

%Evaluate the function at these points
Z = cos(X1).*X2;

%Define a point we are interested in linearizing
x1o = 0;
x2o = 5;
zo  = cos(x1o)*x2o;

%Plot the surface
figure
hold on
surf(X1,X2,Z)

%Plot the point
plot3(x1o, x2o, zo, 'ro', 'MarkerSize', 15, 'LineWidth', 3)

%Make the plot nice
xlabel('x_1')
ylabel('x_2')
zlabel('z=f(x_1,x_2)')
title('z=f(x_1,x_2)')
legend('f(x_1,x_2)','f(x_{1o},x_{2o})')
grid on
shading interp        %Make the 3D surface smooth
view([35 30])         %Change the view of the plot (can also use button on plot window)
```

This produces an output of