Christopher Lum
lum@uw.edu

Lecture 04h
Gradient of a Function and the Directional Derivative

Lecture is on YouTube

The YouTube video entitled 'Gradient of a Function and the Directional Derivative' that covers this lecture is located at https://youtu.be/obeu4B8mXuw

Gradient of a Function

For a given scalar function $f(x, y, z)$, the gradient of f is the vector function defined by

$$\text{grad } f = \nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k} = \begin{pmatrix} \partial f / \partial x \\ \partial f / \partial y \\ \partial f / \partial z \end{pmatrix} \qquad \textbf{\textit{(Eq.1)}}$$

The ∇ is referred to as "nabla" or "del".

Note that the gradient is a function of the point $P = \begin{pmatrix} x_o & y_o & z_o \end{pmatrix}$ where it is evaluated at. To remind ourselves of this fact, sometimes it is easier to write the gradient as

$$\text{grad } f(x_o, y_o, z_o) = \nabla f(x_o, y_o, z_o) = \left(\frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}\right) \Big|_{x=x_o, y=y_o, z=z_o}$$

For many engineering applications, we may need to extend the idea of a function's gradient to more than 3 dimensions. In higher dimensions, the gradient of the function $f = f(x_1, x_2, ..., x_n)$ is given by

$$\text{grad } f = \nabla f = \begin{pmatrix} \partial f / \partial x_1 \\ \partial f / \partial x_2 \\ \vdots \\ \partial f / \partial x_n \end{pmatrix}$$

Intuitively, the gradient provides the direction in which to move if you want to increase the function value as quickly as possible.

Example: 2D function

Consider a function that returns the altitude (h) at a given x and y location.

```
In[ ]:= f[x_, y_] = 3 x^2 + y^3;
```

We can compute the gradient

In[∘]:=
```
gradF[x_, y_] = (D[f[x, y], x]
                 D[f[x, y], y]);
gradF[x, y] // MatrixForm
```

Out[∘]//MatrixForm=

$$\begin{pmatrix} 6\,x \\ 3\,y^2 \end{pmatrix}$$

If we consider a point

$$P = \begin{pmatrix} x_o \\ y_o \end{pmatrix} = \begin{pmatrix} -3 \\ 2 \end{pmatrix}$$

In[∘]:=
```
xo = -3;
yo = 2;
```

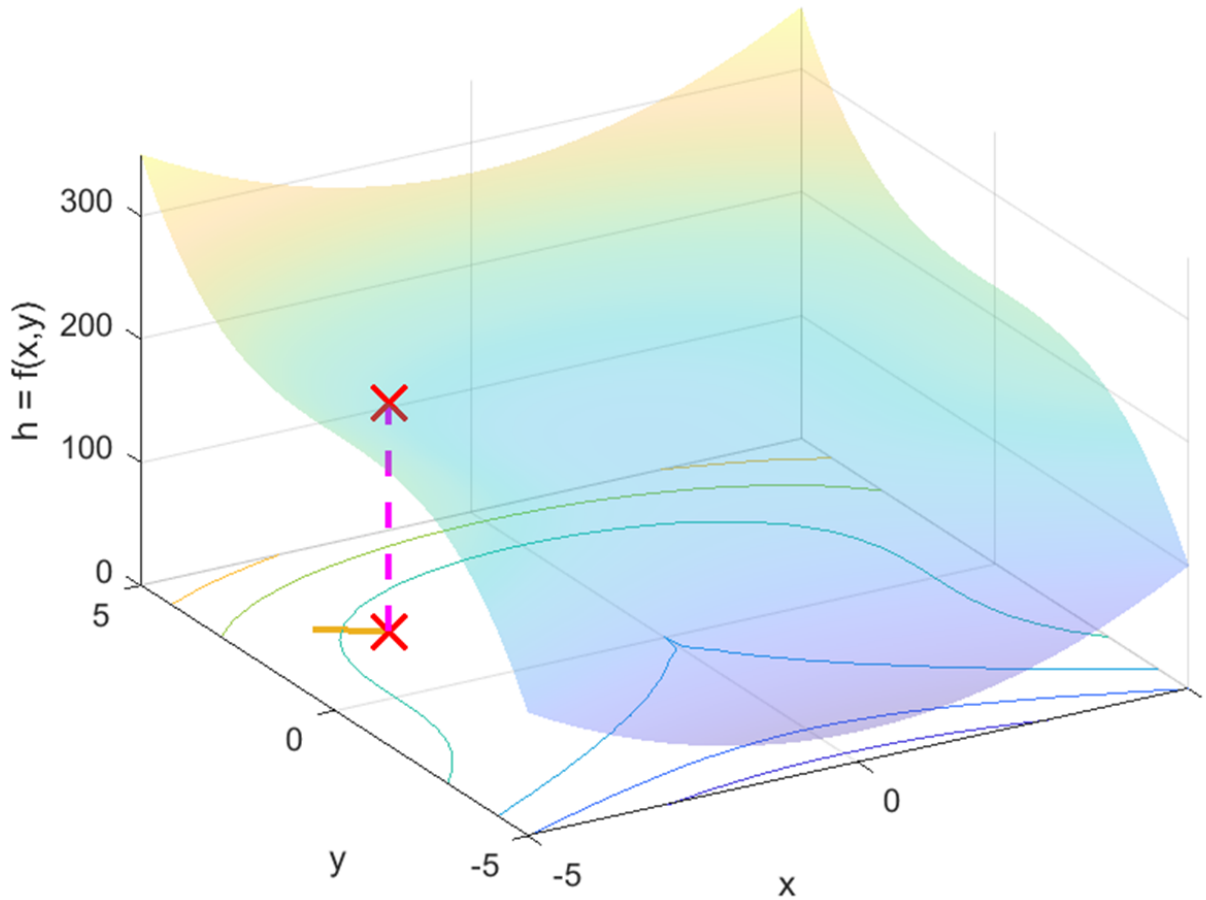
Then the gradient at this location is given as

In[∘]:=
```
gradF[xo, yo] // MatrixForm
```

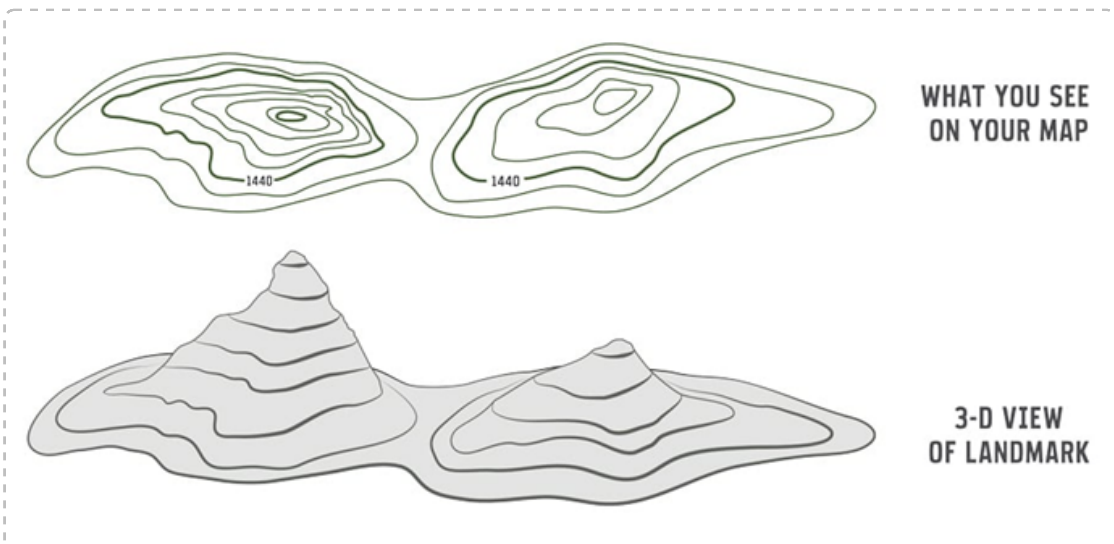
Out[∘]//MatrixForm=

$$\begin{pmatrix} -18 \\ 12 \end{pmatrix}$$

We can visualize the artifacts as shown below (note that the gradient was normalized to a unit vector).



The physical description/intuition with the gradient is that it is a vector that shows the direction of maximum function value increase at the point *P*.



We will revisit this when we discuss optimization, especially numerical optimization via gradient descent.

```
In[]:= Clear[gradF, xo, yo, f]
```

Directional Derivative

The rate of change of f at any point P in any fixed direction given by a vector $\overline{b}$ is defined as the directional derivative of f at point P in the direction of $\overline{b}$. This is given by

$$D_{\overline{b}} f = \frac{df}{ds} = \lim_{s \to 0} \frac{f(Q) - f(P)}{s} \qquad \textbf{\textit{(Eq.3)}}$$

where $D_{\overline{b}} f$ = directional derivative of f at point P in the direction of $\overline{b}$

Q = a point on the ray from P in the direction of $\overline{b}$

s = distance between P and Q

Recall from our discussion of vector dot products and scalar projections, the dot product effectively measures how aligned two vectors are with each other. Furthermore, recall from calculus in 1D that the derivative of a function measures how fast the function is changing in a direction. Therefore, we consider the vector $\overline{b}$ dotted with the vector ∇f to be a measure of how much the function is changing in the direction of $\overline{b}$.

If $\overline{b}$ is a unit vector, we can express the directional derivative of f in the direction of $\overline{b}$ as

$$D_{\overline{b}} f = \overline{b} \cdot \nabla f \qquad (\text{if } |\overline{b}| = 1) \qquad \textbf{\textit{(Eq.5)}}$$

If $|\overline{b}| \neq 1$, we need to simply normalize it to obtain the directional derivative

$$D_{\overline{b}} f = \frac{\overline{b}}{|\overline{b}|} \cdot \nabla f \qquad \textbf{\textit{(Eq.5*)}}$$

Note that the gradient of f is a function of the point $P = (\, x_o \quad y_o \quad z_o \,)$. Therefore, the directional derivative is also a function of the point P, so often you will see the notation of

$$D_{\overline{b}} f(x_o, y_o, z_o) = \frac{\overline{b}}{|\overline{b}|} \cdot \nabla f(x_o, y_o, z_o)$$

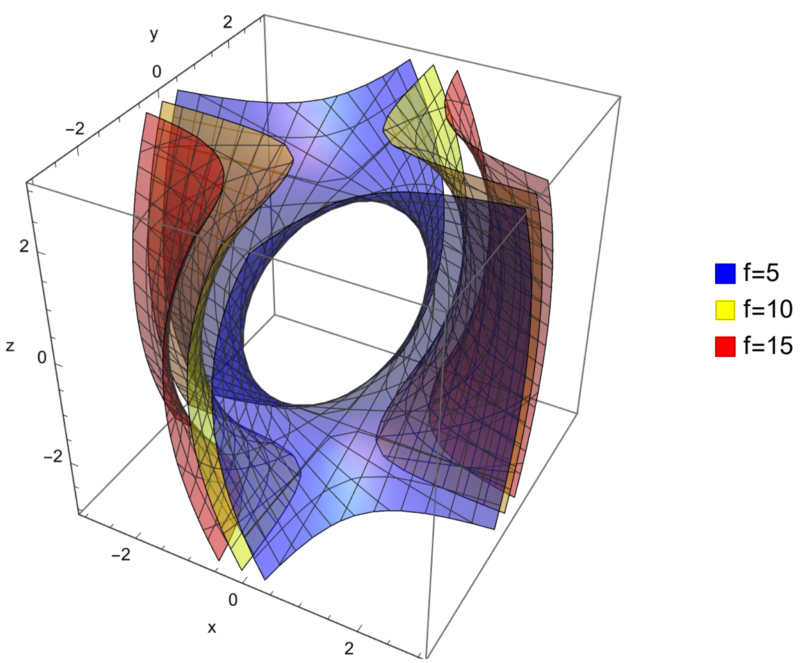
Example 1: Gradient. Directional derivative

The temperature in the atmosphere is given by

$$T = f(x, y, z) = 4x + z^2$$

In[]:=
```
f[x_, y_, z_] = 4 x y + z^2;

(*Visualize*)
lim = 3;
alpha = 0.5;
plot1 = Legended[
  Show[
   ContourPlot3D[f[x, y, z] == 5, {x, -lim, lim}, {y, -lim, lim}, {z, -lim, lim},
    ContourStyle → Directive[Blue, Opacity[alpha], Specularity[White, 30]]],

   ContourPlot3D[f[x, y, z] == 10, {x, -lim, lim}, {y, -lim, lim}, {z, -lim, lim},
    ContourStyle → Directive[Yellow, Opacity[alpha], Specularity[White, 30]]],

   ContourPlot3D[f[x, y, z] == 15, {x, -lim, lim}, {y, -lim, lim}, {z, -lim, lim},
    ContourStyle → Directive[Red, Opacity[alpha], Specularity[White, 30]]],

   (*Plot Options*)
   AxesLabel → {"x", "y", "z"}
  ],

  (*Add legend information*)
  SwatchLegend[{Blue, Yellow, Red}, {"f=5", "f=10", "f=15"}]
 ]
```

Out[]=



Suppose that we have an autonomous rotor craft vehicle that is capable in moving in all three directions. The aircraft is currently located at the point

$$P = \begin{pmatrix} 2 \\ 1 \\ -1 \end{pmatrix}$$

In[ ]:=
```
xo = 2;
yo = 1;
zo = -1;
```

If the aircraft moves in the direction of

$$\overline{b} = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$$

In[ ]:=
```
b = (1
     -2
      1);
```

What is the rate of change of temperature in this direction?

We can first compute the gradient of the function

In[ ]:=
```
gradF[x_, y_, z_] = (D[f[x, y, z], x]
                     D[f[x, y, z], y]
                     D[f[x, y, z], z]);

gradF[x, y, z] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 4\,y \\ 4\,x \\ 2\,z \end{pmatrix}$$

Mathematica also provides the function ‘Grad’ to compute the gradient

In[ ]:=
```
Grad[f[x, y, z], {x, y, z}]
```

Out[ ]= {4 y, 4 x, 2 z}

Note that this returns the gradient as a 1D array/list rather than a 2D matrix.

So the gradient at the point *P* is obtained by evaluating our previous expression at the *P*.

In[ ]:=
```
gradF[xo, yo, zo] // MatrixForm
```

Out[ ]//MatrixForm=

$$\begin{pmatrix} 4 \\ 8 \\ -2 \end{pmatrix}$$

We can now use the directional derivative to determine the rate of change of the function in the specified direction

$$D_{\overline{b}} f(x_o, y_o, z_o) = \frac{\overline{b}}{|\overline{b}|} \cdot \nabla f(x_o, y_o, z_o)$$

In[ ]:=
```
  1
------- Transpose[b].gradF[xo, yo, zo] // N
Norm[b]
```

Out[ ]=
```
{{-5.71548}}
```

So we see that

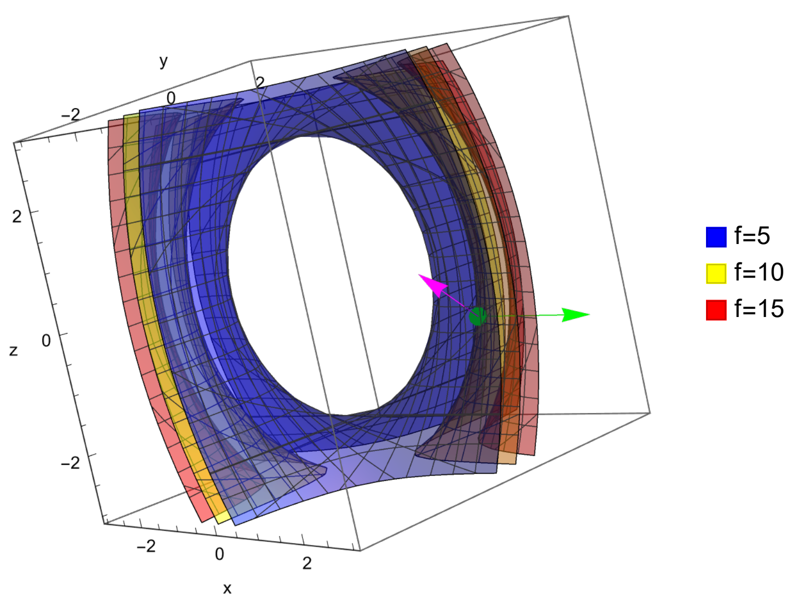
$$D_{\overline{b}}f(2, 1, -1) = -5.715$$

The negative number means that if the aircraft moves in the direction of $\overline{b}$, the temperature actually decreases at a rate of -5.715 units temperature per unit movement.

In[ ]:=

```
P = (xo
     yo
     zo);
pt = Point[Transpose[P]];

Show[plot1,
 (*Draw P*)
 Graphics3D[{AbsolutePointSize[10], Green, pt}],

 (*Plot b*)
 Graphics3D[
  {
   Magenta, Arrow[{{P[[1, 1]], P[[2, 1]], P[[3, 1]]},
     {P[[1, 1]] + b[[1, 1]], P[[2, 1]] + b[[2, 1]], P[[3, 1]] + b[[3, 1]]}}]
  }],

 (*plot gradF*)
 Graphics3D[
  {
   Green, Arrow[{{P[[1, 1]], P[[2, 1]], P[[3, 1]]},
     {P[[1, 1]] + gradF[xo, yo, zo][[1, 1]] / 5,
      P[[2, 1]] + gradF[xo, yo, zo][[2, 1]] / 5, P[[3, 1]] + gradF[xo, yo, zo][[3, 1]] / 5}}]
  }
 ]
]
```

Out[ ]=



```
Clear[gradF, b, xo, yo, zo, f]
```

Gradient is the Vector of Maximum Increase

Recall that the directional derivative characterizes how the function changes in the direction of the vector $\overline{b}$. The natural question to ask is then, which direction yields the largest increase?

$$D_{\overline{b}} f(x_o, y_o, z_o) = \frac{\overline{b}}{|\overline{b}|} \cdot \nabla f(x_o, y_o, z_o)$$

We can view this as the dot product of two vectors

$D_{\overline{b}} f(x_o, y_o, z_o) = \frac{1}{|\overline{b}|} < \overline{b}, \nabla f(x_o, y_o, z_o) >$ recall: $< \overline{u}, \overline{v} \geq | \overline{u} | \; | \overline{v} | \cos(\theta)$ where $\theta =$ angle between $\overline{u}$ and $\overline{v}$

From this, we see that in order to maximize $D_{\overline{b}} f(x_o, y_o, z_o)$, we should choose $\overline{b}$ to be in the same direction as $\nabla f(x_o, y_o, z_o)$, so that $\theta = 0$ and $\cos(\theta) = 1$.

By similar analysis, if we want to minimize $D_{\overline{b}} f(x_o, y_o, z_o)$, we should choose $\overline{b}$ to be in the opposite direction of $\nabla f(x_o, y_o, z_o)$, so that $\theta = \pi$ and $\cos(\theta) = -1$

To summarize

$$\left\{\overline{b} = \alpha \nabla f(x_o, y_o, z_o),\ \alpha > 0\right\} \Longleftrightarrow \left\{D_{\overline{b}} f(x_o, y_o, z_o) \text{ is maximum}\right\} \qquad \text{(maximum increase)}$$
$$\left\{\overline{b} = \alpha \nabla f(x_o, y_o, z_o),\ \alpha < 0\right\} \Longleftrightarrow \left\{D_{\overline{b}} f(x_o, y_o, z_o) \text{ is minimum}\right\} \qquad \text{(steepest descent)}$$

This realization has many important applications such as optimization, gradient climbing algorithms, etc. (we will study these in greater detail in later lectures)

Gradient as a Surface Normal Vector

We will investigate this in the YouTube video entitled ‘Parameterizing Surfaces and Computing Surface Normal Vectors’ that covers this lecture is located at https://youtu.be/a3_c4c9PYNg