Christopher Lum
lum@uw.edu

Lecture05d
Parameterizing Surfaces and Computing Surface Normal Vectors

Lecture is on YouTube

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

---

Simple Surface Descriptions

Often, simple surfaces take the form of

$$z = g(x, y) \qquad \textbf{(Eq.1)}$$

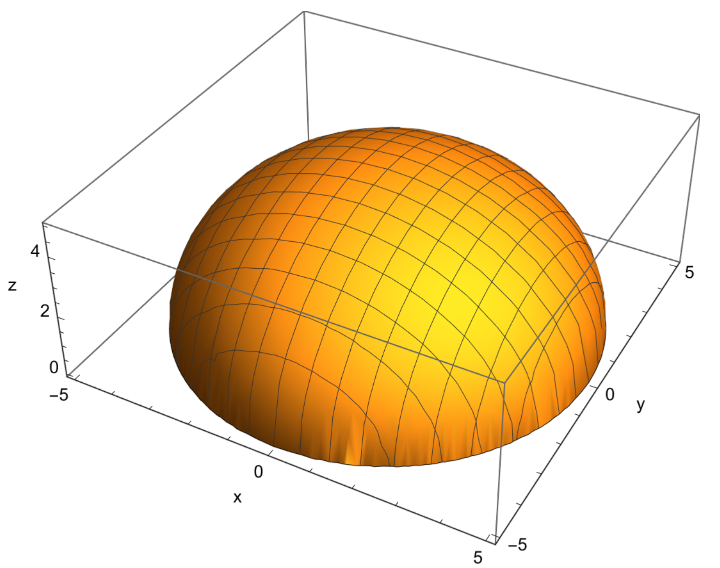
Example: Functional Representation of a Hemisphere

$$z = \sqrt{a^2 - x^2 - y^2}$$

These both represent the upper hemisphere of radius a centered at the origin

In[1]:=
```
a = 5;
Plot3D[Sqrt[a^2 - x^2 - y^2], {x, -a, a}, {y, -a, a},
 AxesLabel → {"x", "y", "z"}]
Clear[a]
```

Out[2]=



Note that we can alternatively write Eq.1 as

$$x^2 + y^2 + z^2 = a^2 \qquad z \geq 0$$

This form is a perhaps more general way of defining a surface. Consider a surface, S, which can be defined by

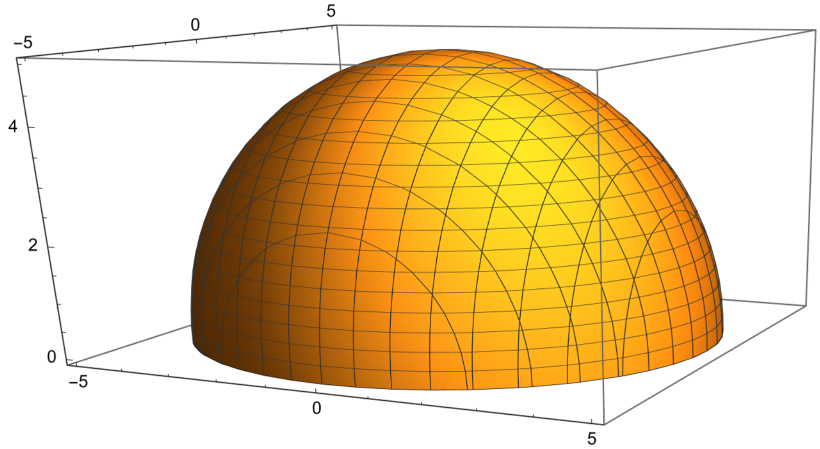
$$S = \{x, y, z \mid f(x, y, z) = \alpha = \text{constant}\} \qquad \textbf{\textit{(Eq.2)}}$$

In other words, the surface is comprised of all points which satisfy the relationship $f(x, y, z) = \alpha$. Such a surface is called a **level surface** of f and for different α we obtain different level surfaces.

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

a = 5;
ContourPlot3D[f[x, y, z] == a^2, {x, -a, a}, {y, -a, a}, {z, 0, a},
 AspectRatio → 1 / 2]
Clear[a]
```

Out[6]=



Gradient as a Surface Normal

Recall that we can parameterize a curve C_1 with a single independent variable t.

$\bar{r}(t) = x(t)\,\hat{i} + y(t)\,\hat{j} + z(t)\,\hat{k}$ (describes a curve parameterized by t)

If we would like C_1 to lie on S (see figure below), we require that the $x(t)$, $y(t)$, and $z(t)$ value on the curve also satisfy $f(x, y, z) = \alpha$.

$$f\,(x(t),\ y(t),\ z(t)) = \alpha \qquad \textbf{\textit{(Eq.3)}}$$

If we differentiate Eq.3 (using the chain rule) we obtain

$$\frac{\partial}{\partial t}[f(x(t),\ y(t),\ z(t))] = \frac{\partial}{\partial t}[\alpha]$$

$$\frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial t} = 0$$

$$\begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} & \frac{\partial f}{\partial z} \end{pmatrix} \begin{pmatrix} \frac{\partial x}{\partial t} \\ \frac{\partial y}{\partial t} \\ \frac{\partial z}{\partial t} \end{pmatrix} = 0 \quad \text{recall: grad } f = \nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}$$

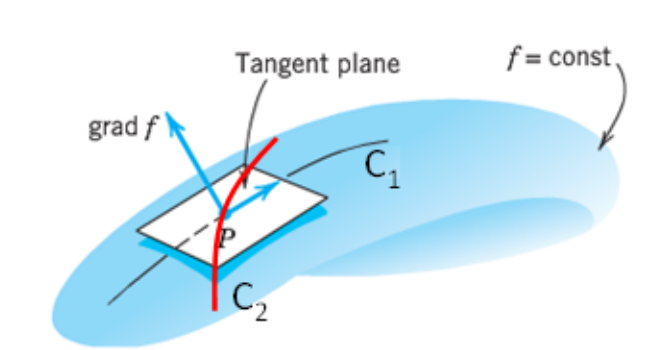
recall: $\bar{r}'(t) = \frac{\partial x}{\partial t}\hat{i} + \frac{\partial y}{\partial t}\hat{j} + \frac{\partial z}{\partial t}\hat{k}$ (tangent vector to C_1 at $\bar{r}(t)$)

$< \nabla f, \bar{r}'(t) \geq 0$ ***(Eq.4)***

Eq.4 states that the inner product between the gradient vector and the tangent vector to C_1 is always zero. The only way for two vectors to have an inner product of zero is if they are perpendicular to one another. Therefore ∇f is perpendicular to the tangent vector along C_1 at the point P

We can generate another another arbitrary curve, C_2, on the surface of S that also passes through P and repeat the same analysis. We will come to the same conclusion that ∇f must be perpendicular to the tangent vector along C_2 at the point P.

The only way for ∇f to be simultaneously perpendicular to both tangent vectors is that it must be orthogonal from the surface S.



This leads to the following theorem

Theorem: Gradient as a Surface Normal Vector

Let f be a differentiable scalar function that represents a surface $S : f(x, y, z) = \alpha =$ constant. Then if the gradient of f at a point P of S is not the zero vector, it is a normal vector of S at P.

Example: Gradient as Surface Normal Vector - Hemisphere

Consider the same hemisphere as above

We can compute the surface normal at the point

In[8]:=
$$P = \begin{pmatrix} -a \operatorname{Cos}\left[\frac{\pi}{9}\right] \operatorname{Sin}\left[\frac{7\pi}{36}\right] \\ -a \operatorname{Sin}\left[\frac{\pi}{9}\right] \operatorname{Sin}\left[\frac{7\pi}{36}\right] \\ a \operatorname{Cos}\left[\frac{7\pi}{36}\right] \end{pmatrix};$$

```
P // MatrixForm // N
```

Out[9]//MatrixForm=
$$\begin{pmatrix} -0.538986\, a \\ -0.196175\, a \\ 0.819152\, a \end{pmatrix}$$

We can first verify that this is on the surface

In[10]:=
```
P[[1, 1]]^2 + P[[2, 1]]^2 + P[[3, 1]]^2 == a^2 // Simplify
```
Out[10]=
```
True
```

Note that we can rewrite the surface as $f(x, y, z) = \alpha$

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

In[11]:=
```
f[x_, y_, z_] = x^2 + y^2 + z^2;
```

We can now compute the gradient of f

$$\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}$$

In[12]:=
```
Delf[x_, y_, z_] = {{D[f[x, y, z], x]}, {D[f[x, y, z], y]}, {D[f[x, y, z], z]}};
Delf[x, y, z] // MatrixForm
```
Out[13]//MatrixForm=

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

So the gradient at the point P is given as

In[14]:=
```
DelfP = Delf[P[[1, 1]], P[[2, 1]], P[[3, 1]]];
DelfP // MatrixForm
DelfP // MatrixForm // N
```
Out[15]//MatrixForm=

$$\begin{pmatrix} -2\,a\,\mathrm{Cos}\left[\frac{\pi}{9}\right]\,\mathrm{Sin}\left[\frac{7\pi}{36}\right] \\ -2\,a\,\mathrm{Sin}\left[\frac{\pi}{9}\right]\,\mathrm{Sin}\left[\frac{7\pi}{36}\right] \\ 2\,a\,\mathrm{Cos}\left[\frac{7\pi}{36}\right] \end{pmatrix}$$

Out[16]//MatrixForm=

$$\begin{pmatrix} -1.07797\,a \\ -0.392349\,a \\ 1.6383\,a \end{pmatrix}$$

As discussed previously, this gradient is normal to the surface at this point. We can plot the surface and this gradient to graphically verify this.

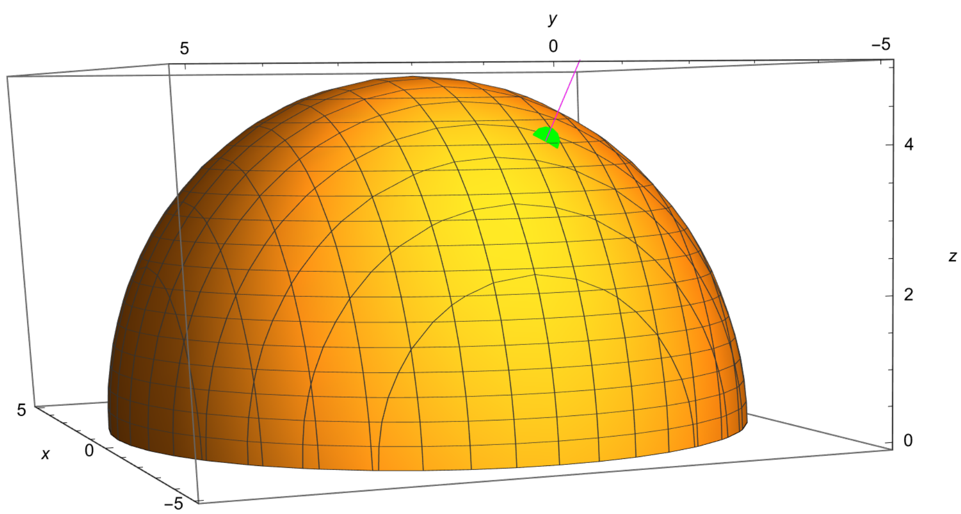
In[17]:=
```
a = 5;
Show[
 (*Plot the surface*)
 ContourPlot3D[f[x, y, z] == a^2, {x, -a, a}, {y, -a, a}, {z, 0, a},
  AxesLabel → {x, y, z}, AspectRatio → 1 / 2],

 (*Plot the point of interest*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{P[[1, 1]], P[[2, 1]], P[[3, 1]]}]
  }
 ],

 (*Plot the surface normal*)
 Graphics3D[
  {
   Magenta, Arrow[{{P[[1, 1]], P[[2, 1]], P[[3, 1]]},
      {P[[1, 1]] + DelfP[[1, 1]], P[[2, 1]] + DelfP[[2, 1]], P[[3, 1]] + DelfP[[3, 1]]}}]
  }
 ]

]
```

Out[18]=



In[19]:=
```
Clear[a, DelfP, f, P]
```

Example 2: Gradient as a Surface Normal Vector - Cone

Consider a surface defined by

$$z^2 = 4\left(x^2 + y^2\right)$$

Compute the surface normal at the point $P = (1 \quad 0 \quad 2)$. Note that this is a point on the surface

In[20]:=
```
P = (1
     0
     2);
```

We can first verify that this is on the surface

In[21]:=
```
4 (P[[1, 1]]^2 + P[[2, 1]]^2) == P[[3, 1]]^2 // Simplify
```

Out[21]=
```
True
```

Note that we can rewrite the surface as $f(x, y, z) = \alpha$

$$f(x, y, z) = 4\left(x^2 + y^2\right) - z^2 = 0$$

In[22]:=
```
f[x_, y_, z_] = 4 (x^2 + y^2) - z^2;
```

We can now compute the gradient of f

$$\nabla f = \frac{\partial f}{\partial x}\hat{i} + \frac{\partial f}{\partial y}\hat{j} + \frac{\partial f}{\partial z}\hat{k}$$

In[23]:=
```
Delf[x_, y_, z_] = (D[f[x, y, z], x]
                    D[f[x, y, z], y]
                    D[f[x, y, z], z]);

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

Out[24]//MatrixForm=

$$\begin{pmatrix} 8\,x \\ 8\,y \\ -2\,z \end{pmatrix}$$

So the gradient at the point P is given as

In[25]:=
```
DelfP = Delf[P[[1, 1]], P[[2, 1]], P[[3, 1]]]
```

Out[25]=
```
{{8}, {0}, {-4}}
```

As discussed previously, this gradient is normal to the surface at this point. We can plot the surface and this gradient to graphically verify this.

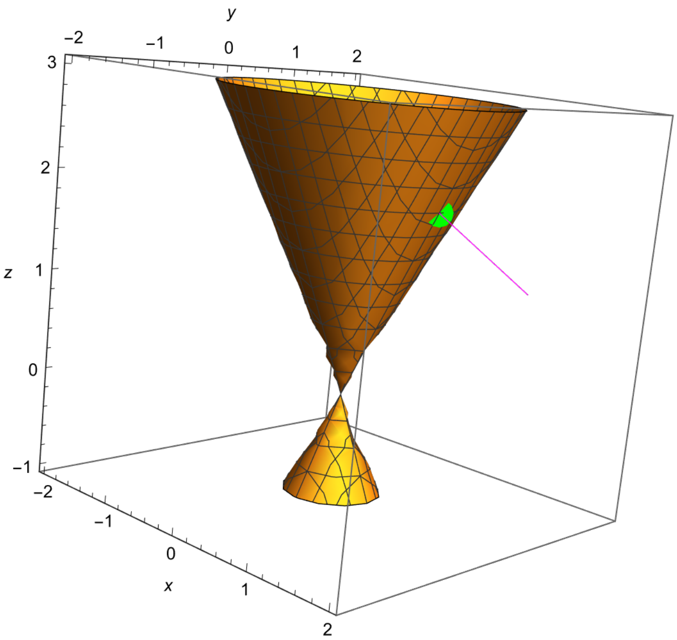
In[26]:=
```
Show[
 (*Plot the surface*)
 ContourPlot3D[f[x, y, z] == 0, {x, -2, 2}, {y, -2, 2}, {z, -1, 3}, AxesLabel → {x, y, z}],

 (*Plot the point of interest*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{P[[1, 1]], P[[2, 1]], P[[3, 1]]}]
  }
 ],

 (*Plot the surface normal*)
 Graphics3D[
  {
   Magenta, Arrow[{{P[[1, 1]], P[[2, 1]], P[[3, 1]]},
     {P[[1, 1]] + DelfP[[1, 1]], P[[2, 1]] + DelfP[[2, 1]], P[[3, 1]] + DelfP[[3, 1]]}}]
  }
 ]

]
```

Out[26]=



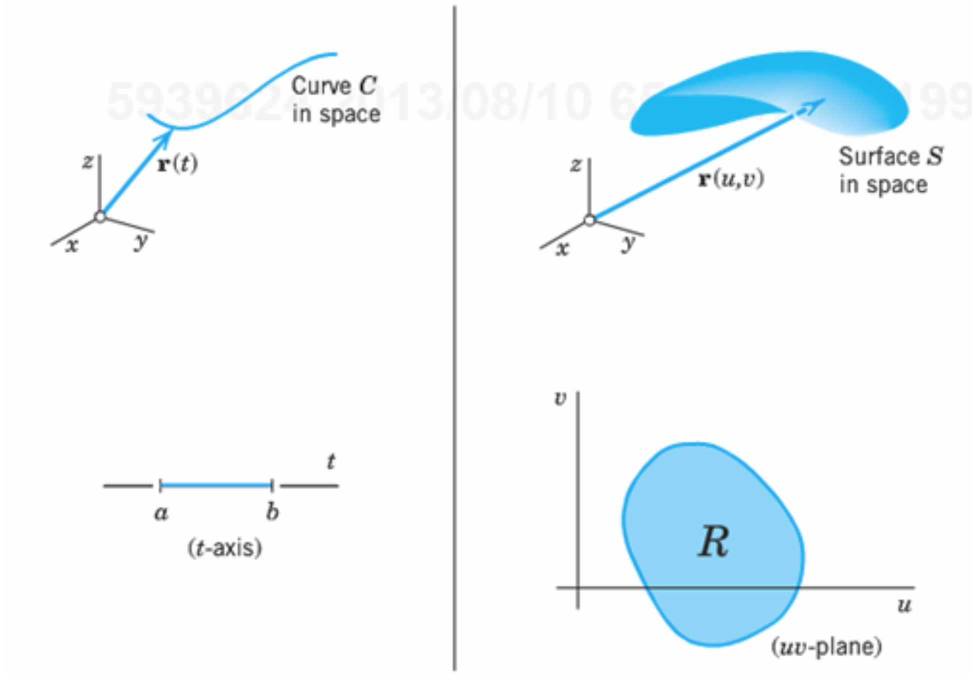
In[27]:=
```
Clear[DelfP, Delf, f, P]
```

Parameterizing Surfaces

In general, a more robust way to define a surface is to use a parameterization similar to how we parameterized curves in space.

Parameterizing in uv-plane

Recall that for line integrals, it was often easier to parameterize the curve using a single parameter, t. The same is true for a surface. However, a surface is two dimensional and therefore, we will require 2 parameters, u and v, to parameterize it appropriately.



(A) Curve (B) Surface

A parametric representation of surface S is then of the form

$$\bar{r}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} = x(u, v)\,\hat{i} + y(u, v)\,\hat{j} + z(u, v)\,\hat{k} \qquad \textbf{(Eq.5)}$$

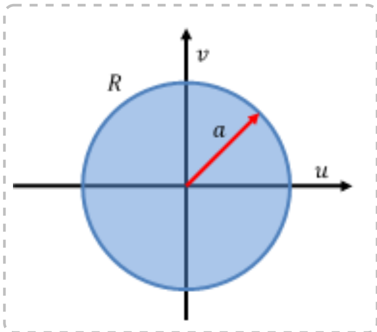
And u and v can vary in the region R in the uv – plane.

Example: Parametric Representation of a Hemisphere

Again we consider the hemisphere of radius a centered at the origin that we investigated in the previous example. In this case, a simple parameterization of this is

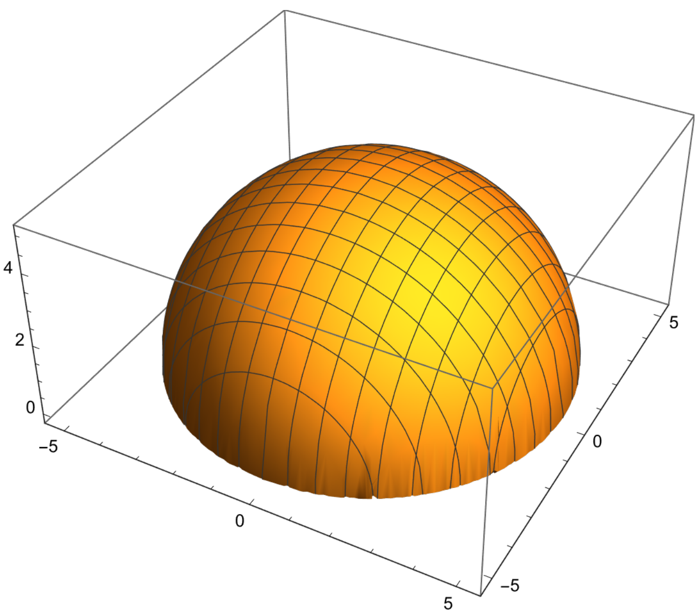
$$= \bar{r}(u, v) = \begin{pmatrix} u \\ v \\ \sqrt{a^2 - u^2 - v^2} \end{pmatrix}$$

And the region R in the uv plane is shown below



In[28]:=
```
a = 5;
(*Note that this is not an exactly correct representation of the region R
 but Mathematica does not plot imaginary terms so this appears correct*)
ParametricPlot3D[{u, v, (a^2 - u^2 - v^2)^(1/2)}, {u, -a, a}, {v, -a, a}]
Clear[a]
```

Out[29]=



We can formulate an alternate parameterization of this surface by using spherical coordinates. Recall that a coordinate in spherical coordinates is given as (see YouTube video entitled 'Cartesian, Polar, Cylindrical, and Spherical Coordinates' at https://youtu.be/FLQXW6G9P8I)

$$\overline{p}^s = \begin{pmatrix} \text{radial distance} \\ \text{azimuth angle} \\ \text{inclination/polar angle} \end{pmatrix} = \begin{pmatrix} r \\ \theta \\ \phi \end{pmatrix}$$

Recall that the conversion between spherical coordinates and Cartesian coordinates is

$$x = r\sin(\phi)\cos(\theta)$$
$$y = r\sin(\phi)\sin(\theta)$$
$$z = r\cos(\phi)$$

We note that in spherical coordinates, this surface is simple to describe as

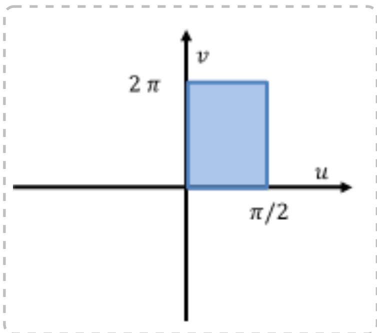
$r = a$
$v = \theta \in [0, 2\pi]$ (longitude)
$u = \phi \in [0, \pi/2]$ (latitude)

The Cartesian parameterization of the surface is therefore

$$\bar{r}(u, v) = \begin{pmatrix} a\sin(u)\cos(v) \\ a\sin(u)\sin(v) \\ a\cos(u) \end{pmatrix}$$

where $u = \phi$ (latitude)
$v = \theta$ (longitude)

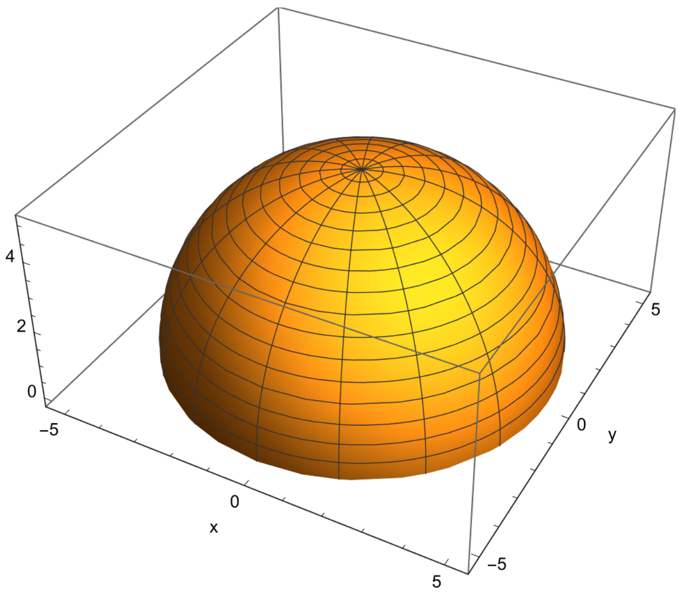
The advantage of this parameterization is that the region *R* in the uv plane is a simple rectangle



In[31]:=
```
r[u_, v_] = {a Sin[u] Cos[v], a Sin[u] Sin[v], a Cos[u]};

a = 5;
ParametricPlot3D[r[u, v], {u, 0, π / 2}, {v, 0, 2 π},
 AxesLabel → {"x", "y", "z"}]
```

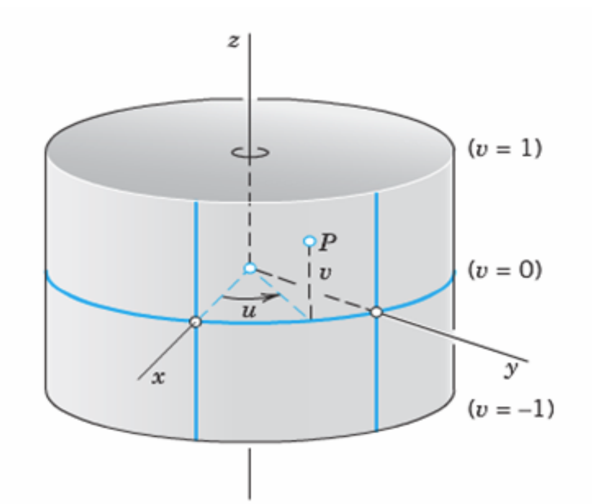
Out[33]=



In[34]:=

Example: Parametric Representation of a Cylinder

Consider a circular cylinder $x^2 + y^2 = a^2$ with $z \in [-1, 1]$ that has a radius a and height 2. Note that this surface has no top nor bottom "lids" to the cylinder.



One possible parametric representation of the cylinder is

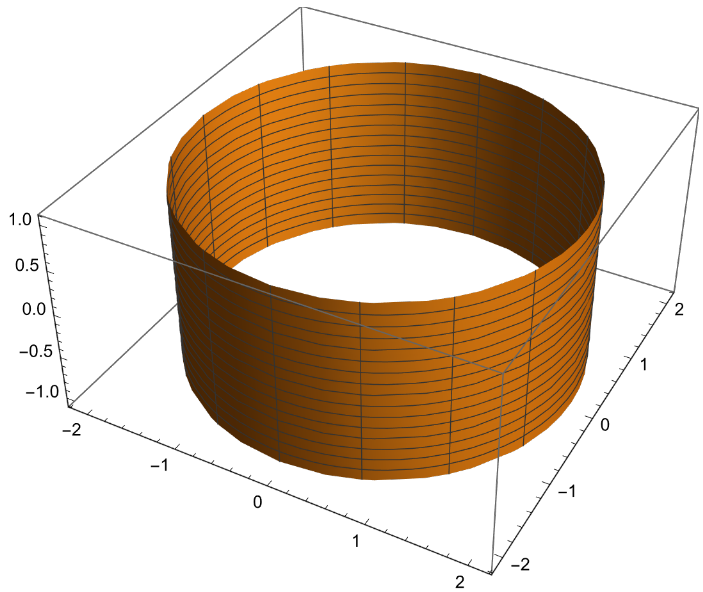
$$\bar{r}(u, v) = [a\cos(u),\ a\sin(u),\ v] = a\cos(u)\,\hat{i} + a\sin(u)\,\hat{j} + v\,\hat{k}$$

$u \in [0, 2\pi]$

$v \in [-1, 1]$

In[35]:=
```
a = 2;
ParametricPlot3D[{a Cos[u], a Sin[u], v}, {u, 0, 2 π}, {v, -1, 1}]
Clear[a]
```

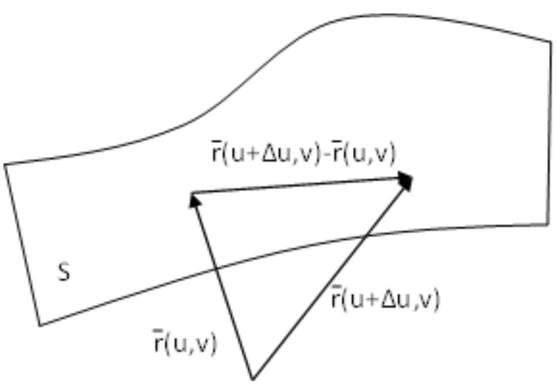
Out[36]=



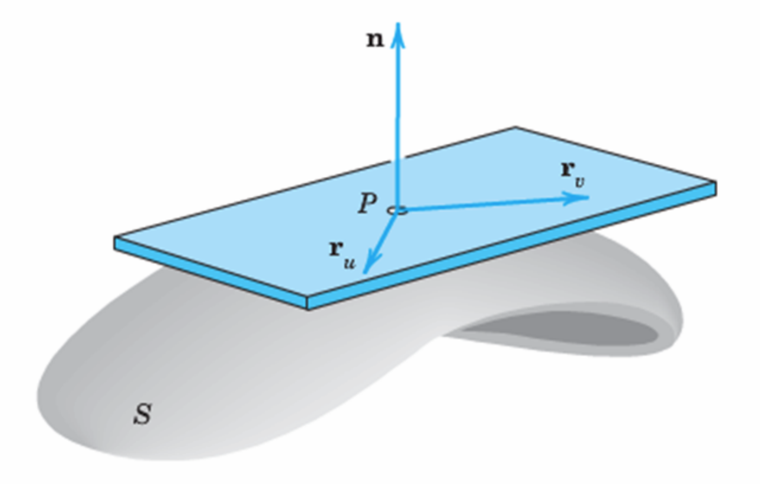
Tangent Plane and Surface Normal

In the previous section we saw how the gradient could be used as the surface normal. We now investigate a method to compute the gradient for a parameterized surface.

We can compute the tangent plane to the surface by looking at how the position vector $\bar{r}(u, v)$ changes with respect to u and v.



The partial derivatives $\bar{r}_u = \partial \bar{r} / \partial u$ and $\bar{r}_v = \partial \bar{r} / \partial v$ should both be tangent to the surface at point P.



So their cross product gives the surface normal

$$\overline{N} = \overline{r}_u \times \overline{r}_v \qquad \textbf{(Eq.6)}$$

Interestingly, note that cross product actually gives the area of the parallelogram enclosed by $\overline{r}_u$ and $\overline{r}_v$. As such, note that it is likely not a unit vector. This will become important later when we consider surface integrals.

This can be normalized to obtain the unit normal vector of S at a point P

$$\overline{n} = \frac{1}{|\overline{N}|}\overline{N} \qquad \textbf{(Eq.7)}$$

Recall from the previous lecture that if S is represented by $g(x, y, z) = 0$ instead of the parametrization with u, and v, we could write the surface normal vector using the gradient of g

$$\overline{n} = \frac{1}{|\text{grad}\, g|}\text{grad}\, g \qquad \textbf{(Eq.7*)}$$

Example: Unit Normal Vector of a Hemisphere (Revisited)

We now re-examine the hemisphere example using an alternative approach.

In[38]:=
```
r[u_, v_] = {a Sin[u] Cos[v], a Sin[u] Sin[v], a Cos[u]};
```

We can now compute the surface normal vector using Eq.5. We first compute $\overline{r}_u$ and $\overline{r}_v$

$$\overline{r}_u = \partial\overline{r}/\partial u = [\cos(v), \sin(v), 1]$$

$$\overline{r}_v = \partial\overline{r}/\partial v = [-u\sin(v), u\cos(v), 0]$$

In[39]:=
```
ru[u_, v_] = D[r[u, v], u]
rv[u_, v_] = D[r[u, v], v]
```

Out[39]= {a Cos[u] Cos[v], a Cos[u] Sin[v], -a Sin[u]}

Out[40]= {-a Sin[u] Sin[v], a Cos[v] Sin[u], 0}

We now compute the normal vector using

$$\overline{N} = \overline{r}_u \times \overline{r}_v$$

In[41]:=
```
NVector[u_, v_] = Cross[ru[u, v], rv[u, v]] // Simplify
```

Out[41]= $\{a^2 \text{Cos}[v] \, \text{Sin}[u]^2, \, a^2 \text{Sin}[u]^2 \, \text{Sin}[v], \, a^2 \text{Cos}[u] \, \text{Sin}[u]\}$

We can now choose a point, (u_o, v_o), to evaluate the normal vector.

In[42]:=
```
(*Choose a point of interested in the uv space*)
uGiven = 30 π/180;
vGiven = 235 * π/180;
```

Recall that this point is in the uv space, not xyz space (although its equivalent point in xyz space is easy to obtain by simply applying the parameterization)

In[44]:=
```
(*What is the equivalent point in xyz space?*)
rGiven = r[uGiven, vGiven];
rGiven /. {a → 5} // MatrixForm // N
```

Out[45]//MatrixForm=

$$\begin{pmatrix} -1.43394 \\ -2.04788 \\ 4.33013 \end{pmatrix}$$

We can now find the surface normal vector at this point

In[46]:=
```
(*What is the surface normal at this point?*)
NGiven = NVector[uGiven, vGiven];
NGiven // MatrixForm

(*Show that this is not a normal vector*)
Norm[NGiven] /. {a → 5} // N
```

Out[47]//MatrixForm=

$$\begin{pmatrix} -\frac{1}{4} a^2 \text{Sin}\left[\frac{7\pi}{36}\right] \\ -\frac{1}{4} a^2 \text{Cos}\left[\frac{7\pi}{36}\right] \\ \frac{\sqrt{3}\, a^2}{4} \end{pmatrix}$$

Out[48]= 12.5

We can now visualize this using ParametricPlot3D

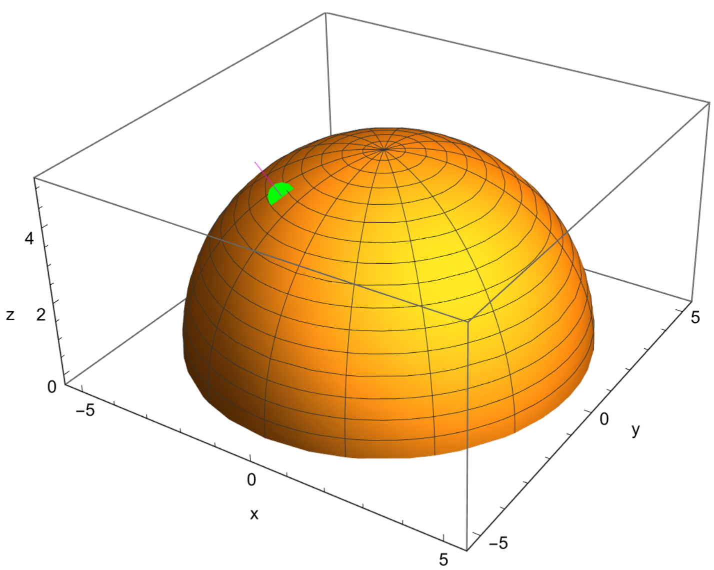
In[49]:=
```
a = 5;
(*Plot the scenario*)
Show[
 (*Plot the surface*)
 ParametricPlot3D[
  r[u, v], {u, 0, π / 2}, {v, 0, 2 π},
  AxesLabel → {"x", "y", "z"}
 ],

 (*Plot the point of interest*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{rGiven[[1]], rGiven[[2]], rGiven[[3]]}]
  }
 ],

 (*Plot the surface normal*)
 Graphics3D[
  {
   Magenta, Arrow[{{rGiven[[1]], rGiven[[2]], rGiven[[3]]},
     {rGiven[[1]] + NGiven[[1]], rGiven[[2]] + NGiven[[2]], rGiven[[3]] + NGiven[[3]]}}]
  }
 ]

]
```

Out[50]=



In[51]:=
```
Clear[a, NGiven, rGiven, vGiven, uGiven, NVector, rv, ru, r]
```

Example: Unit Normal Vector of a Cone (Revisited)

We now re-examine the code example using an alternative approach.

One parametric representation of the cone is

$$\bar{r}(u, v) = [u \cos(v),\ u \sin(v),\ u]$$

where $u \in [0, H]$ (H = height of cone)
$v \in [0, 2\pi]$

```
In[52]:= r[u_, v_] = {u Cos[v], u Sin[v], u};
```

We can now compute the surface normal vector using Eq.5. We first compute $\bar{r}_u$ and $\bar{r}_v$

$$\bar{r}_u = \partial \bar{r} / \partial u = [\cos(v),\ \sin(v),\ 1]$$

$$\bar{r}_v = \partial \bar{r} / \partial v = [-u \sin(v),\ u \cos(v),\ 0]$$

```
In[53]:= ru[u_, v_] = D[r[u, v], u]
         rv[u_, v_] = D[r[u, v], v]
Out[53]= {Cos[v], Sin[v], 1}
Out[54]= {-u Sin[v], u Cos[v], 0}
```

We now compute the normal vector using

$$\bar{N} = \bar{r}_u \times \bar{r}_v$$

```
In[55]:= NVector[u_, v_] = Cross[ru[u, v], rv[u, v]] // Simplify
Out[55]= {-u Cos[v], -u Sin[v], u}
```

We can now choose a point, (u_o, v_o), to evaluate the normal vector.

```
In[56]:= (*Choose a point of interested in the uv space*)
         uGiven = 1;
         vGiven = 234.34 * π/180;
```

Recall that this point is in the uv space, not xyz space (although its equivalent point in xyz space is easy to obtain by simply applying the parameterization)

```
In[58]:= (*What is the equivalent point in xyz space?*)
         rGiven = r[uGiven, vGiven];
         rGiven // MatrixForm
Out[59]//MatrixForm=
```

$$\begin{pmatrix} -0.582974 \\ -0.812491 \\ 1 \end{pmatrix}$$

We can now find the surface normal vector at this point

In[60]:=
```
(*What is the surface normal at this point?*)
NGiven = NVector[uGiven, vGiven];
NGiven // MatrixForm
```

Out[61]//MatrixForm=

$$\begin{pmatrix} 0.582974 \\ 0.812491 \\ 1 \end{pmatrix}$$

We can now visualize this using ParametricPlot3D

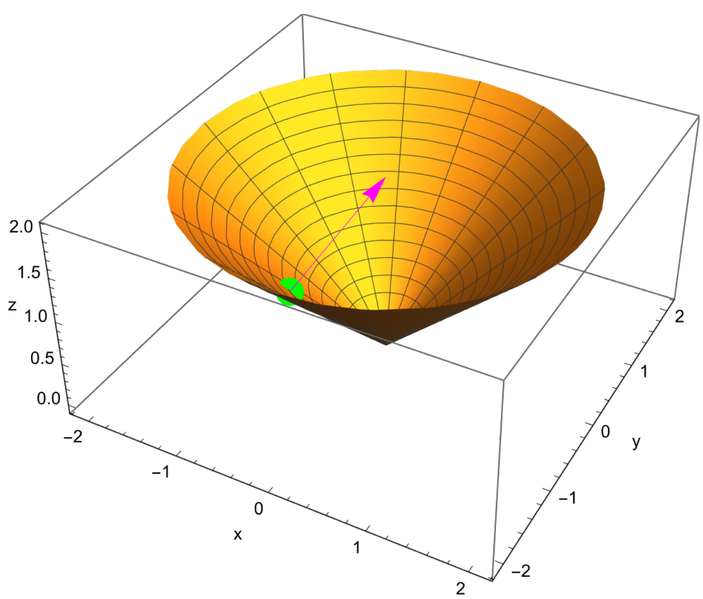
In[62]:=
```
(*Plot the scenario*)
Show[
 (*Plot the surface*)
 ParametricPlot3D[
  {u Cos[v], u Sin[v], u}, {u, 0, 2}, {v, 0, 2 π},
  AxesLabel → {"x", "y", "z"}
 ],

 (*Plot the point of interest*)
 Graphics3D[
  {
   AbsolutePointSize[15], Green, Point[{rGiven[[1]], rGiven[[2]], rGiven[[3]]}]
  }
 ],

 (*Plot the surface normal*)
 Graphics3D[
  {
   Magenta, Arrow[{{rGiven[[1]], rGiven[[2]], rGiven[[3]]},
     {rGiven[[1]] + NGiven[[1]], rGiven[[2]] + NGiven[[2]], rGiven[[3]] + NGiven[[3]]}}]
  }
 ]

]
```

Out[62]=



In[63]:=
```
Clear[NGiven, rGiven, vGiven, uGiven, NVector, rv, ru, r]
```

Note that the direction of the normal vector may not be what is expected as it points “inside” the surface, but it is normal to the surface as expected. This will become relevant later when we discuss orientated surfaces.

Example: More Complicated Parameterization

Consider the parameterization of

$$\bar{r}(u, v) = \begin{pmatrix} (2 + \sin(v))\cos(u) \\ (2 + \sin(v)\sin(u) \\ u + \cos(v) \end{pmatrix}$$

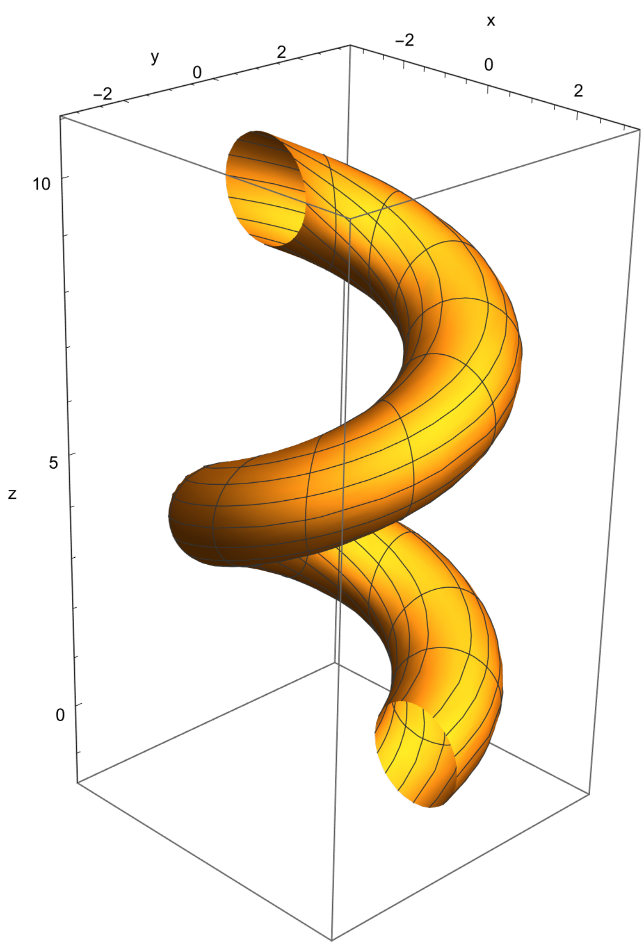
$u \in [0, 3\pi]$

$v \in [0, 2\pi]$

In[64]:=
```
r[u_, v_] = {(2 + Sin[v]) Cos[u], (2 + Sin[v]) Sin[u], u + Cos[v]};

(*Visualize the surface*)
plot1 = ParametricPlot3D[
  r[u, v], {u, 0, 3 π}, {v, 0, 2 π},
  AxesLabel → {"x", "y", "z"}
 ]
```

Out[65]=



We again compute the surface normal vector. Note that this time, we simply add a -1 to the surface normal equation to make it point in the opposite direction

In[66]:=
```
ru[u_, v_] = D[r[u, v], u];
rv[u_, v_] = D[r[u, v], v];

NVector[u_, v_] = -Cross[ru[u, v], rv[u, v]] // Simplify;

ru[u, v] // MatrixForm
rv[u, v] // MatrixForm
NVector[u, v] // MatrixForm
```

Out[69]//MatrixForm=

$$\begin{pmatrix} -\mathrm{Sin}[u]\ (2+\mathrm{Sin}[v]) \\ \mathrm{Cos}[u]\ (2+\mathrm{Sin}[v]) \\ 1 \end{pmatrix}$$

Out[70]//MatrixForm=

$$\begin{pmatrix} \mathrm{Cos}[u]\ \mathrm{Cos}[v] \\ \mathrm{Cos}[v]\ \mathrm{Sin}[u] \\ -\mathrm{Sin}[v] \end{pmatrix}$$

Out[71]//MatrixForm=

$$\begin{pmatrix} \mathrm{Cos}[v]\ \mathrm{Sin}[u]+\mathrm{Cos}[u]\ \mathrm{Sin}[v]\ (2+\mathrm{Sin}[v]) \\ -\mathrm{Cos}[u]\ \mathrm{Cos}[v]+\mathrm{Sin}[u]\ \mathrm{Sin}[v]\ (2+\mathrm{Sin}[v]) \\ \mathrm{Cos}[v]\ (2+\mathrm{Sin}[v]) \end{pmatrix}$$

We can plot the surface normal at various *u* and *v* values

In[72]:=
```
Manipulate[
 (*Compute rGiven and NGiven*)
 rGiven = r[uGiven, vGiven];
 NGiven = NVector[uGiven, vGiven];

 (*Plot the scenario*)
 Show[
  (*Plot the surface*)
  plot1,

  (*Plot the point of interest*)
  Graphics3D[
   {
    AbsolutePointSize[15], Green, Point[{rGiven[[1]], rGiven[[2]], rGiven[[3]]}]
   }
  ],

  (*Plot the surface normal*)
  Graphics3D[
   {
    Magenta, Arrow[{{rGiven[[1]], rGiven[[2]], rGiven[[3]]},
       {rGiven[[1]] + NGiven[[1]], rGiven[[2]] + NGiven[[2]], rGiven[[3]] + NGiven[[3]]}}]
   }
  ]
 ],

 (*Manipulate variables*)
 {uGiven, 0, 3 π}, {vGiven, 0, 2 π}
]
```

Out[72]=