Christopher Lum
lum@uw.edu

Lecture 04k
Cylindrical and Spherical Coordinates

Lecture is on YouTube

The YouTube video entitled ‘Cartesian, Polar, Cylindrical, and Spherical Coordinates’ that covers this lecture is located at https://youtu.be/FLQXW6G9P8I.

Outline

-Cylindrical Coordinates
-Spherical Coordinates

Cylindrical Coordinates

Typical analysis uses Cartesian coordinates to describe positions.

$$\bar{p}^c = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

where ‘c’ superscript denotes that this vector is expressed in the Cartesian coordinate system.

In this Cartesian system, we understand that the first element of the 3-tuple vector is the distance along the x-axis. The second element is the distance along the y-axis, and the third element is the distance along the z-axis.

Alternatively, we can express this position using the unit vectors $\hat{i}$, $\hat{j}$, and $\hat{k}$ as

$$\bar{p}^c = x\,\hat{i} + y\,\hat{j} + z\,\hat{k}$$

where $\hat{i} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$ $\hat{j} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$ $\hat{k} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$

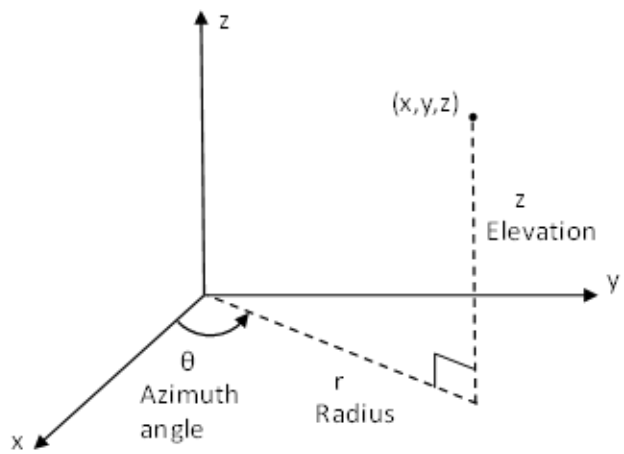
Recall that **cylindrical coordinates** instead describe positions using a radius (r), angle (θ), and altitude (z). This is a 3D extension of the 2D polar coordinate system.

$$\overline{p}^p = \begin{pmatrix} \text{radius} \\ \text{azimuth angle} \\ \text{elevation} \end{pmatrix} = \begin{pmatrix} r \\ \theta \\ z \end{pmatrix}$$

where ‘p’ superscript denotes that this vector is expressed in the cylindrical/polar coordinates

Cylindrical coordinates in 2 dimensions (only x, y and r, θ) are referred to as **polar coordinates**.

In this cylindrical system, we understand that the first element of the 3-tuple vector is the distance in the XY plane towards the point in question (this is a “radial distance”). The second element is the angle between the x-axis and this radial distance, and the third element is the distance along the z-axis (same definition as the Cartesian system).



The same position in space, $\overline{p}$, can be described by three coordinates (r, θ, z) with the understanding that these are expressed in cylindrical coordinates.

The unit vectors $\hat{e}_r$, $\hat{e}_\theta$, and $\hat{e}_z$ are defined as

$\hat{e}_r$ = unit vector in direction of increasing r, holding θ and z constant
$\hat{e}_\theta$ = unit vector in direction of increasing θ, holding r and z constant
$\hat{e}_z$ = unit vector in direction of increasing z, holding r and θ constant

From this geometry, we see that the transformation between Cartesian and cylindrical coordinates is given by

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

In vector notation

$$\overline{p}^c = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = f(\overline{p}^p) = \begin{pmatrix} r\cos(\theta) \\ r\sin(\theta) \\ z \end{pmatrix} \qquad \textbf{\textit{(Eq.1.1)}}$$

Note that any Cartesian coordinate specifies a single point of 3D space. On the other hand, every point has infinitely many equivalent cylindrical coordinates due to the fact that any multiple of 2π for the azimuth angle will describe the same point. If it is necessary to define a unique set of cylindrical coordinates for each point, one may restrict the range of θ. A common choice is

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

Again, from geometry, we see that the reverse transformation is given by

$$\overline{p}^p = \begin{pmatrix} r \\ \theta \\ z \end{pmatrix} = f^{-1}(\overline{p}^c) = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \text{atan2}(y, x) \\ z \end{pmatrix} \qquad \textbf{\textit{(Eq.1.2)}}$$

Note: here we assume that atan2 requires the y component as the first argument and the x component as the second argument. <REFERENCE 4-QUADRANT INVERSE TANGENT VIDEO HERE>

Example

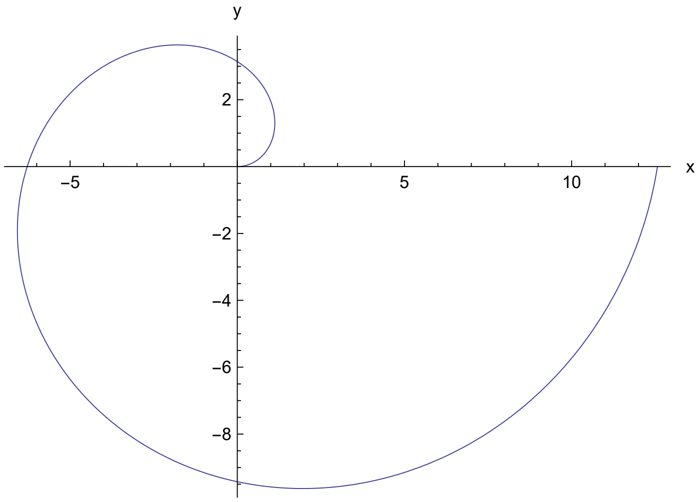
```
(*Define point in cartesian coordinates*)
Print["Cartesian Coordinates"]
xCartesian = -3
yCartesian = -1
zCartesian = -4

(*Compute point in cylindrical coordinates*)
rCylindrical = (xCartesian^2 + yCartesian^2)^(1/2);
θCylindrical = ArcTan[xCartesian, yCartesian];
zCylindrical = zCartesian;

Print["Cylindrical Coordinates"]
rCylindrical // N
θCylindrical * 180 / π // N
zCylindrical // N

(*Verify reverse transformation holds*)
Print["Verify reverse transformation"]
xCartesian == rCylindrical Cos[θCylindrical]
yCartesian == rCylindrical Sin[θCylindrical]
zCartesian == zCylindrical
```

```
Cartesian Coordinates

-3

-1

-4

Cylindrical Coordinates

3.16228

-161.565

-4.

Verify reverse transformation

True

True

True
```

Show Matlab animations

Mathematica supplies the ‘PolarPlot’ function to plot 2D functions in polar coordinates. If you have a function that describes the radius as a function of the angle θ, this function can plot it.

```
r[θ_] = 2 θ;
PolarPlot[r[θ], {θ, 0, 360 π/180},
 AxesLabel → {"x", "y"}]
Clear[r]
```

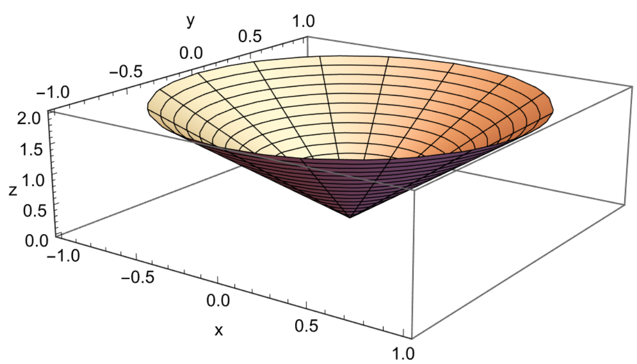


Mathematica supplies the 'RevolutionPlot3D' function to visualize a surface in polar coordinates. This generates a plot of a surface of revolution with height f_z at radius r.

For example, a cone is a surface whose height increases with radius.

$$r \in [r_{\text{min}}, r_{\text{max}}]$$
$$\theta \in [0, 2\pi]$$
$$z = 2r$$

```
fz[r_] = 2 r;
RevolutionPlot3D[fz[r], {r, 0, 1},
 AxesLabel → {"x", "y", "z"}]
Clear[fz]
```



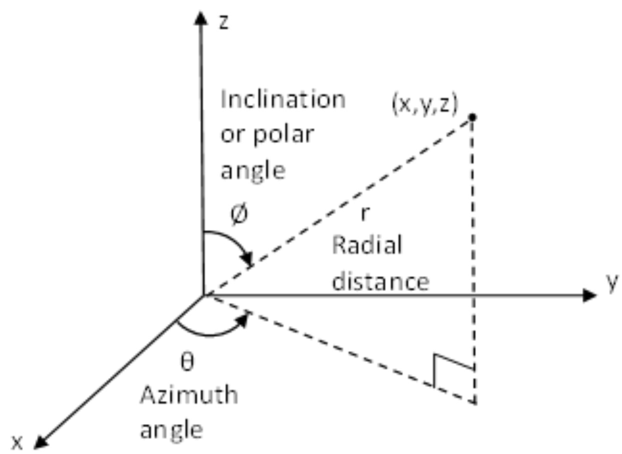
Spherical Coordinates

In a similar fashion, **spherical coordinates** describe positions using a radial distance (r), azimuth angle (θ), polar angle (ϕ). Note that the radial distance in spherical coordinates is defined differently from the radius in cylindrical coordinates. Sometimes $\rho = r$ is used for spherical coordinates to avoid confusion but this usage is not standardized and therefore care must be exercised as r can mean two different distances in the two systems.

In spherical coordinates, we describe a position in space using a radial distance (r), inclination/polar angle (ϕ), and azimuth angle (θ) as shown below

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

where ‘s’ superscript denotes that this vector is expressed in the spherical coordinates



WARNING: To further complicate matters, the notation that is commonly used in physics has the definition of θ and ϕ switched. Therefore, be careful how these two angles are defined in your application.

The same position in space, $\overline{p}$, can be described by three coordinates (r, θ, ϕ) with the understanding that these are expressed in spherical coordinates.

The unit vectors $\hat{e}_r$, $\hat{e}_\theta$, and $\hat{e}_\phi$ are defined as

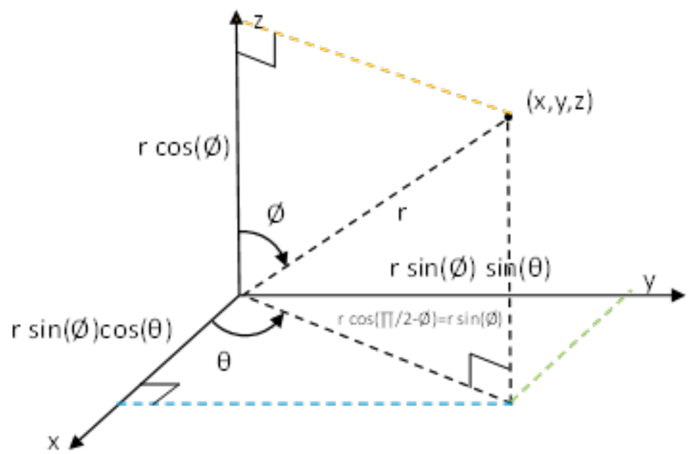
$\hat{e}_r$ = unit vector in direction of increasing r, holding θ and ϕ constant
$\hat{e}_\theta$ = unit vector in direction of increasing θ, holding r and ϕ constant
$\hat{e}_\phi$ = unit vector in direction of increasing ϕ, holding r and θ constant

From this geometry, we see that the transformation between Cartesian and spherical coordinates is given by

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

$y = r \sin(\phi) \sin(\theta)$

$z = r \cos(\phi)$



Cos [π / 2 - ϕ]

Sin[ϕ]

In vector notation

$$\overline{p}^c = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = f(\overline{p}^s) = \begin{pmatrix} r \sin(\phi) \cos(\theta) \\ r \sin(\phi) \sin(\theta) \\ r \cos(\phi) \end{pmatrix} \qquad \textbf{\textit{(Eq.1.3)}}$$

Note that any Cartesian coordinate specifies a single point of 3D space. On the other hand, every point has infinitely many equivalent spherical coordinates due to the fact that any multiple of 2π for the angles will describe the same point. If it is necessary to define a unique set of spherical coordinates for each point, one may restrict their ranges. A common choice is

$r \geq 0$

$\theta \in (-\pi, \pi]$ (longitude)

$\phi \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ (latitude)

This is similar to how the earth’s longitude/latitude system works.

Another common choice is

$r \geq 0$

$\theta \in [0, \pi]$

$\phi \in [0, 2\pi)$

Solving Eq.1.3 simultaneously for r, θ, and ϕ gives the reverse transformation

```
temp = Solve[{x == r Cos[θ] Sin[ϕ], y == r Sin[θ] Sin[ϕ], z == r Cos[ϕ]}, {r, θ, ϕ}];
Simplify[temp[[4]], {r > 0}]
```

$$\left\{r \to \sqrt{x^2+y^2+z^2},\ \theta \to \text{ConditionalExpression}\left[\text{ArcTan}\left[\frac{x}{\sqrt{x^2+y^2}},\ \frac{y}{\sqrt{x^2+y^2}}\right] + 2\pi C[1],\ C[1] \in \mathbb{Z}\right],\right.$$
$$\left.\phi \to \text{ConditionalExpression}\left[\text{ArcTan}\left[\frac{z}{\sqrt{x^2+y^2+z^2}},\ \frac{\sqrt{x^2+y^2}}{\sqrt{x^2+y^2+z^2}}\right] + 2\pi C[2],\ C[2] \in \mathbb{Z}\right]\right\}$$

So we have

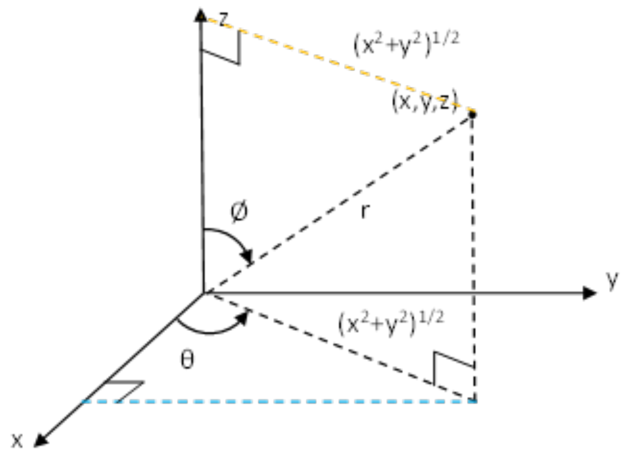
$$\overline{p}^s = \begin{pmatrix} r \\ \theta \\ \phi \end{pmatrix} = f^{-1}(\overline{p}^c) = \begin{pmatrix} \sqrt{x^2+y^2+z^2} \\ \text{atan2}\left(\frac{y}{\sqrt{x^2+y^2}}, \frac{x}{\sqrt{x^2+y^2}}\right) \\ \text{atan2}\left(\frac{\sqrt{x^2+y^2}}{\sqrt{x^2+y^2+z^2}}, \frac{z}{\sqrt{x^2+y^2+z^2}}\right) \end{pmatrix} \qquad \textbf{(Eq.1.4)}$$

```
(*Compute point in spherical coordinates*)
rSpherical = (xCartesian^2 + yCartesian^2 + zCartesian^2)^(1/2);
θSpherical = ArcTan[xCartesian / (xCartesian^2 + yCartesian^2)^(1/2), yCartesian / (xCartesian^2 + yCartesian^2)^(1/2)];
ϕSpherical = ArcTan[zCartesian / (xCartesian^2 + yCartesian^2 + zCartesian^2)^(1/2),
    (xCartesian^2 + yCartesian^2)^(1/2) / (xCartesian^2 + yCartesian^2 + zCartesian^2)^(1/2)];

Print["Spherical Coordinates"]
rSpherical // N
θSpherical * 180 / π // N
ϕSpherical * 180 / π // N

(*Verify reverse transformation holds*)
Print["Verify reverse transformation"]
xCartesian == rSpherical Sin[ϕSpherical] Cos[θSpherical]
yCartesian == rSpherical Sin[ϕSpherical] Sin[θSpherical]
zCartesian == rSpherical Cos[ϕSpherical]
```

```
Spherical Coordinates

5.09902

-161.565

141.671

Verify reverse transformation

True

True

True
```

Alternatively, some formulations attempt to derive the inverse transformation from geometry. Some authors will attempt to write

$$\overline{p}^s = \begin{pmatrix} r \\ \theta \\ \phi \end{pmatrix} = f^{-1}(\overline{p}^c) = \begin{pmatrix} \sqrt{x^2+y^2+z^2} \\ \arccos\left(\frac{x}{\sqrt{x^2+y^2}}\right) \\ \arccos\left(\frac{z}{r}\right) = \arccos\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) \end{pmatrix} \qquad \textbf{\textit{(Eq.1.5)}}$$



Again, we need to be cognizant of the limits of arccos as Eq.1.5 will return the incorrect result in some situations. For example, if we consider the same point and apply Eq.1.5 we obtain

```
(*Compute point in spherical coordinates*)
rSpherical2 = (xCartesian^2 + yCartesian^2 + zCartesian^2)^(1/2);
θSpherical2 = ArcCos[xCartesian / (xCartesian^2 + yCartesian^2)^(1/2)];
ϕSpherical2 = ArcCos[zCartesian / (xCartesian^2 + yCartesian^2 + zCartesian^2)^(1/2)];

Print["Spherical Coordinates (set 2)"]
rSpherical2 // N
θSpherical2 * 180 / π // N
ϕSpherical2 * 180 / π // N

(*Verify reverse transformation holds*)
Print["Verify reverse transformation"]
xCartesian == rSpherical2 Sin[ϕSpherical2] Cos[θSpherical2]
yCartesian == rSpherical2 Sin[ϕSpherical2] Sin[θSpherical2]
zCartesian == rSpherical2 Cos[ϕSpherical2]
```

```
Spherical Coordinates (set 2)

5.09902

161.565

141.671

Verify reverse transformation

True

False

True
```

So we see in this case Eq.1.5 incorrectly computed θ and we instead should have used the solution of -161.565 degrees. As such, Eq.1.4 is a superior formulation of the inverse transformation.

Show Matlab animations

Mathematica supplies the 'SphericalPlot3D' to visualize a surface in spherical coordinates. This can be used in the situation where the radius is prescribed as a function of θ and ϕ. We need to be careful as Mathematica's documentation uses θ to describe polar angle and ϕ to denote the azimuth angle, which is backwards from the notation we are adopting.

Example: Part of a Unit Sphere

We can use SphericalPlot3D to generate a portion of a unit sphere. If we hold the radius constant

$$r(\theta, \phi) = 1$$

We can then let the azimuth angle θ vary as

$$\theta \in \left[0, 45 \frac{\pi}{180}\right]$$

And if we let ϕ vary from

$$\phi \in \left[0, 180 \frac{\pi}{180}\right]$$

We obtain

```
polarAngleMax = 180 π/180;  (*We call this ϕ*)
azimuthAngleMax = 45 π/180;  (*We call this θ*)
R = 1;

SphericalPlot3D[R, {polarAngle, 0, polarAngleMax}, {azimuthAngle, 0, azimuthAngleMax},
 AxesLabel → {"x", "y", "z"}]

Clear[polarAngleMax, azimuthAngleMax, R]
```