

3D Polar Coordinate Conversion

Kaylin Shanahan 2022

This is a python programme to convert spherical polar coordinates of (r, θ, ϕ) to cartesian coordinates (x, y, z) using a function.

In 3 dimensions a point can be defined by cartesian coordinates (x, y, z) or spherical polar coordinates (r, θ, ϕ) .

These cartesian coordinates and sperical polar coordinates are related by the following equations:

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

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

$$z = r \cos \theta$$

- Define function to convert polar coordinates to cartesian coordinates
- Input spherical polar coordinate values of (r, θ, ϕ)
- Calculate cartesian coordinates (x, y, z) with function using equations outlined above
- Output cartesian coordinates (x, y, z)

```
In [8]: # Numpy is needed for our calculations
import numpy as np

# Define the function to convert polar coordinates to cartesian coordinates
def cart(r, t, p):
    x = (r)*(np.sin(t*np.pi/180))*(np.cos(p*np.pi/180))
    y = (r)*(np.sin(t*np.pi/180))*(np.sin(p*np.pi/180))
    z = (r)*(np.cos(t*np.pi/180))
    return x, y, z

# Main programme
def main():
    radius = eval(input("Enter a value for radius in metres: "))
    theta = eval(input("Enter a value for theta in degrees: "))
    phi = eval(input("Enter a value for phi in degrees: "))
    [xv, yv, zv] = cart(radius, theta, phi)
```

```

# Display cartesian coordinates
print("The cartesian coordinates are:")
print("x = {0:10.2e}".format(xv), "metres")
print("y = {0:10.2e}".format(yv), "metres")
print("z = {0:10.2e}".format(zv), "metres")

```

```
main()
```

Enter a value for radius in metres: 2.5

Enter a value for theta in degrees: 25

Enter a value for phi in degrees: 50

The cartesian coordinates are:

x = 6.79e-01 metres

y = 8.09e-01 metres

z = 2.27e+00 metres

In [9]: *# Numpy is needed for our calculations*

```
import numpy as np
```

```
# Define the function to convert polar coordinates to cartesian coordinates
```

```
def cart(r, t, p):
    x = (r)*(np.sin(t*np.pi/180))*(np.cos(p*np.pi/180))
    y = (r)*(np.sin(t*np.pi/180))*(np.sin(p*np.pi/180))
    z = (r)*(np.cos(t*np.pi/180))
    return x, y, z

```

```
# Main programme
```

```
def main():
    radius = eval(input("Enter a value for radius in metres: "))
    theta = eval(input("Enter a value for theta in degrees: "))
    phi = eval(input("Enter a value for phi in degrees: "))
    [xv, yv, zv] = cart(radius, theta, phi)

```

```

# Display cartesian coordinates
print("The cartesian coordinates are:")
print("x = {0:10.2e}".format(xv), "metres")
print("y = {0:10.2e}".format(yv), "metres")
print("z = {0:10.2e}".format(zv), "metres")

```

```
main()
```

Enter a value for radius in metres: 6360000

Enter a value for theta in degrees: 36.7

Enter a value for phi in degrees: 9.1

The cartesian coordinates are:

x = 3.75e+06 metres

y = 6.01e+05 metres

z = 5.10e+06 metres