Scalar and Vector Fields

This tutorial will introduce you to the Five Minute Physics interactive simulations. Open the webpage
https://teaching.smp.uq.edu.au/fiveminutephysics/index.html#course=phys2055&lecture=Fields (https://teaching.smp.uq.edu.au/fiveminutephysics/index.html#course=phys2055&lecture=Fields)
It will take a moment to load, so make sure you have a pen or pencil handy, fill your water bottle, get comfortable. … Once the page loads read the sections on Field Concepts and Representing Fields.

Learning Outcome: Identify scalar and vector fields and be able to draw simple field diagrams.

In [8]:
```
import numpy as np
from mpl_toolkits.mplot3d import Axes3D     # For surface plot
import matplotlib.pyplot as plt
import matplotlib
plt.rcParams["figure.figsize"] = (10,8) #Make the plot bigger
plt.rcParams.update({'font.size': 14})#Make plot fong bigger
```

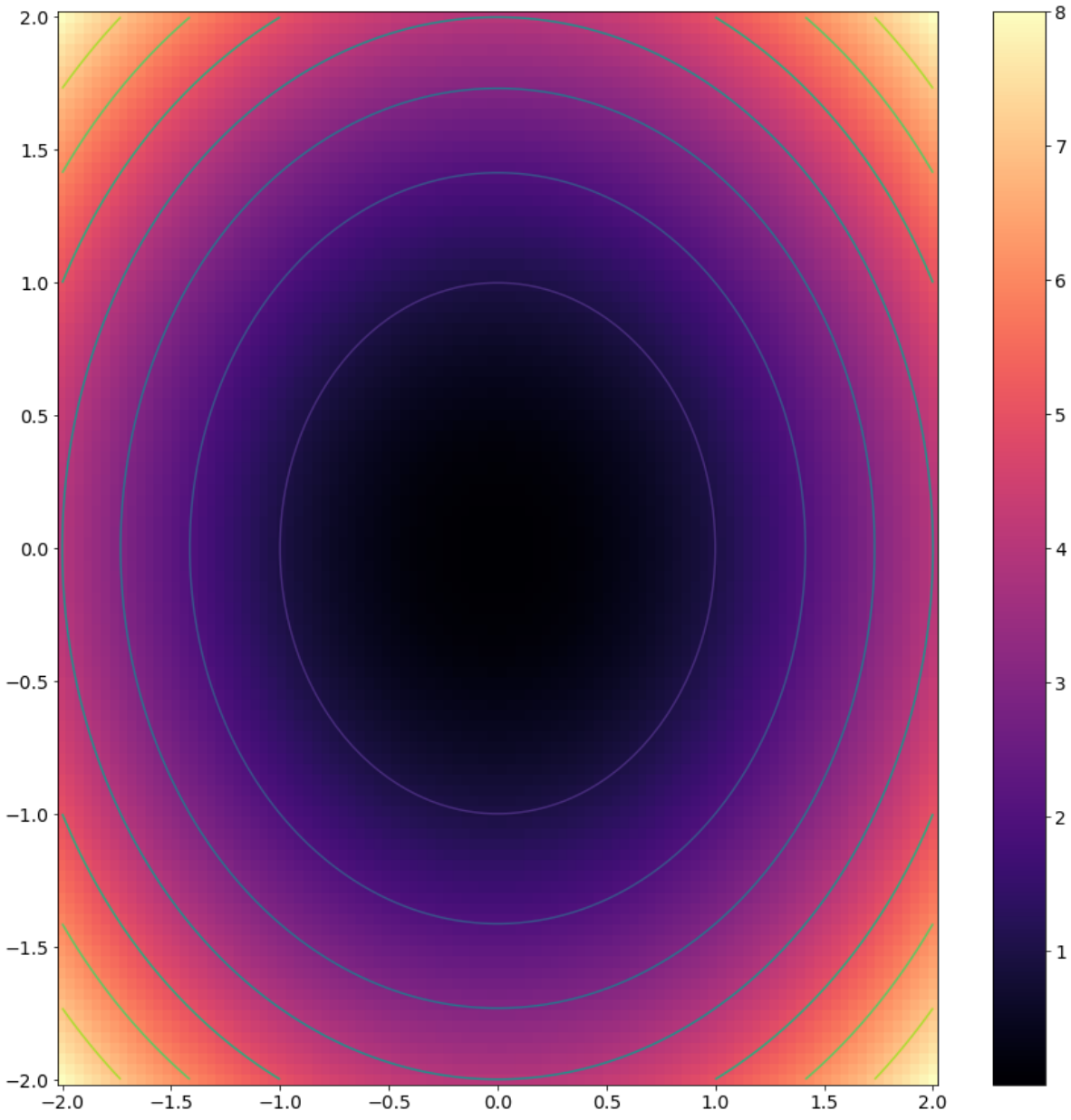
Example 1. A Scalar Field

The code in the next cell is pre-programmed to plot a bowl.

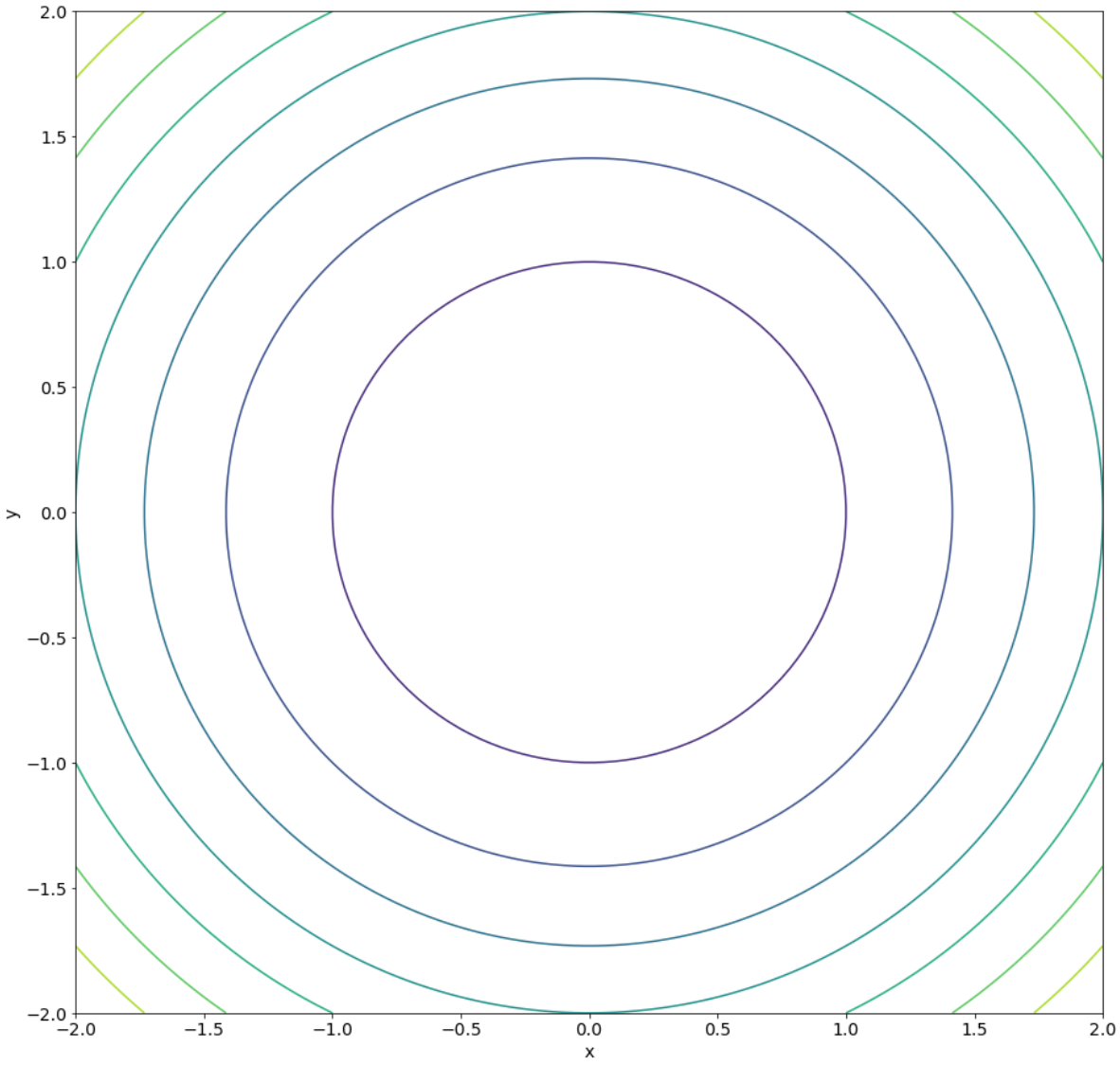
$$f(x, y) = x^2 + y^2$$

1. Use the code below to investigate how a scalar field can be pictorially represented. For each representation in the simulation, give a brief explanation of how it works. How does the simulation represent the properties of the field? What are the advantages and disadvantages of each method? Try different shapes such as a saddle $f(x, y) = x^2 - y^2$, a plane $f(x, y) = x$, and a function of your own creation.
    - Color map
    - Contours
    - Surface
2. Which representation do you think is the best at helping you understand the geometry of the given field. Input your own “User Field” and show your instructor. What are the main advantages of the representation you have chosen?

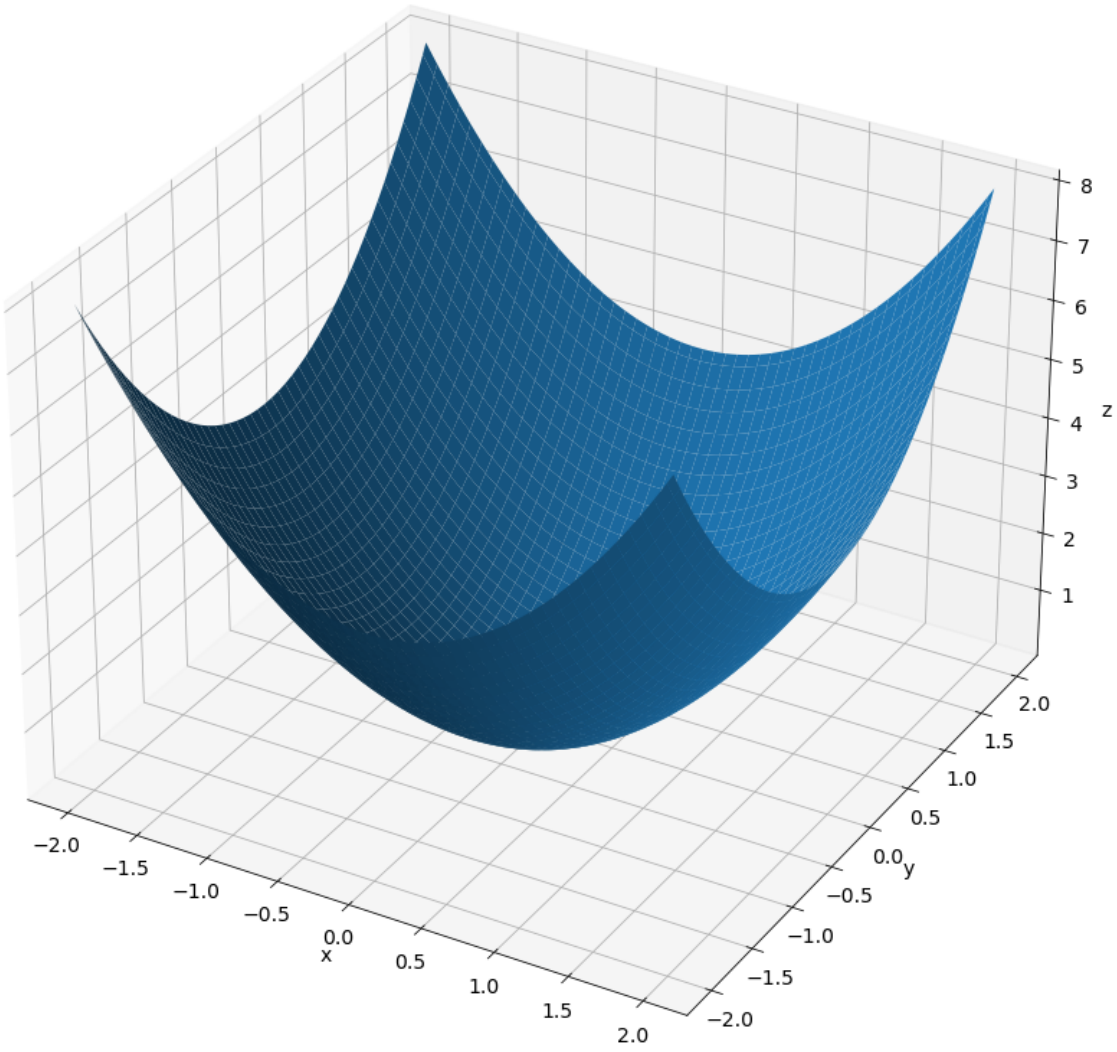
In [53]:
```
# Calculate a grid of points from -2-2 with 100 points in each direction
x,y = np.meshgrid(np.linspace(-2,2,100),np.linspace(-2,2,100));

# Calculate the scalar field
f = x**2 + y**2

# False colour plot
#cmap options are 'magma', 'plasma', 'flag', 'Spectral', 'autumn', 'winter'
plt.pcolor(x,y,f, shading='auto', cmap='magma')#default cmap is viridis
plt.colorbar()
plt.contour(x,y,f)
plt.show()
```



In [45]:
```
# Contour plot
#limits = np.linspace(0, np.amax(f), 10)#create contour values
plt.contour(x,y,f)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



In [38]:
```
# Surface plot
ax = plt.figure().add_subplot(projection='3d')
ax.plot_surface(x,y,f)
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```

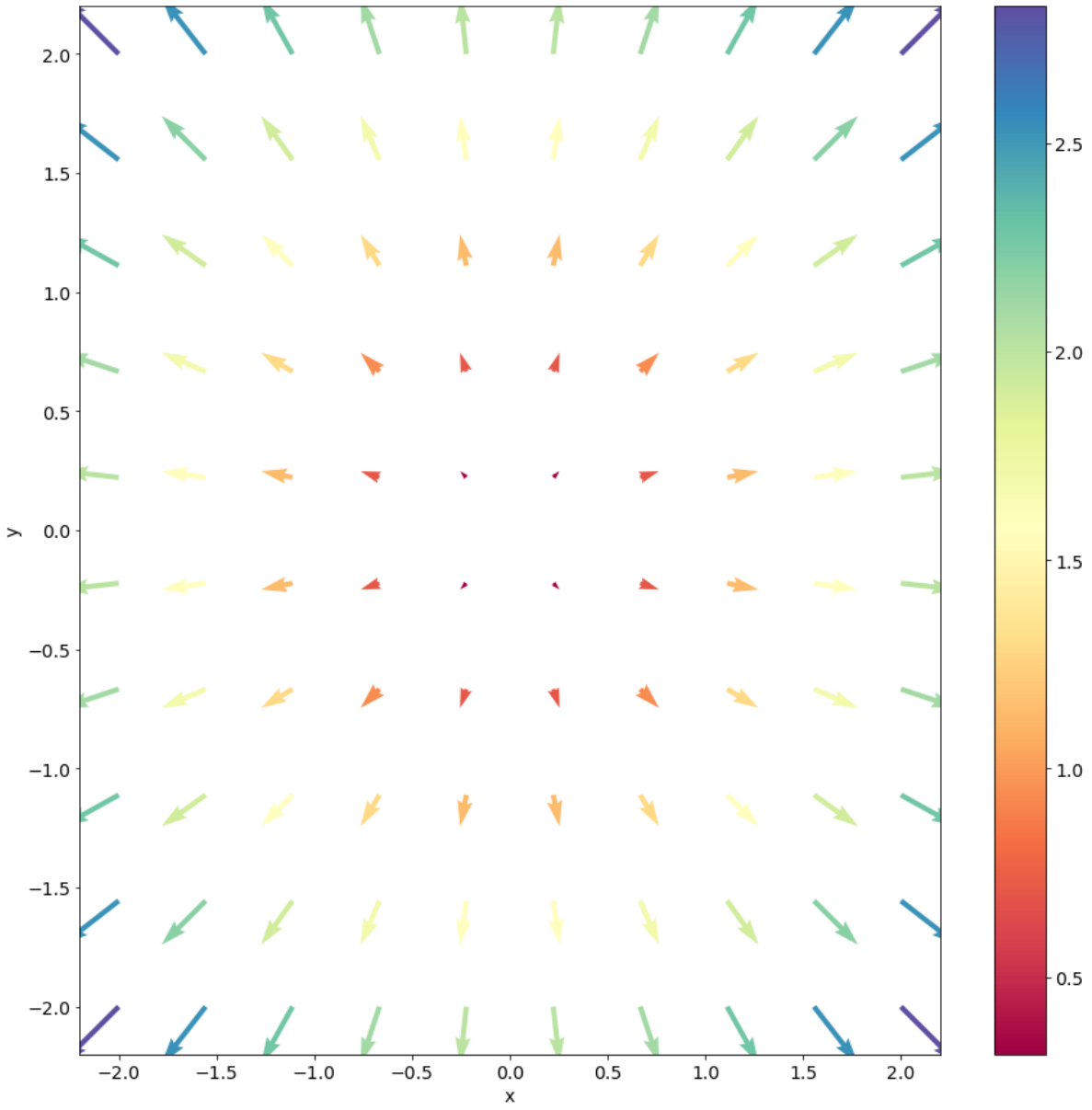


Vector Fields

1. Use the second simulation to explore the properties of a vector field. How does a vector field differ from a scalar field?
2. Compare and contrast the different representations of the vector field.
    - Field Arrows 1
    - Field Arrows 2
    - Color: Some options shown. More found at https://matplotlib.org/stable/gallery/color/colormap_reference.html (https://matplotlib.org/stable/gallery/color/colormap_reference.html)

3. Try creating your own vector field in the simulation using the representation that you think is most useful. Show your instructor.
4. Why must the representation of a vector field be different than that of a scalar field?

In [49]:
```
# Calculate a grid of points from -2-2 with 11 points in each direction
[x,y] = np.meshgrid(np.linspace(-2,2,10),np.linspace(-2,2,10));

# Calculate the vector field (source field)
fx = x#/np.sqrt(x**2+y**2)
fy = y#/np.sqrt(x**2+y**2)

colors = np.abs(np.sqrt(fx**2+fy**2))#create colors based on magnitude of vecte

# Plot arrows
plt.quiver(x, y, fx, fy, colors, cmap='Spectral') #cmap options are 'magma', '
#plt.quiver(x, y, fx, fy)
plt.colorbar()
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Try for yourself

Plot in the xy-plane of the following vector fields. Plot enough different vectors to get a feel for what the field looks like.

$$\vec{f}(x, y) = y\hat{i}$$

$$\vec{g}(x, y) = x\hat{i} - y\hat{j}$$

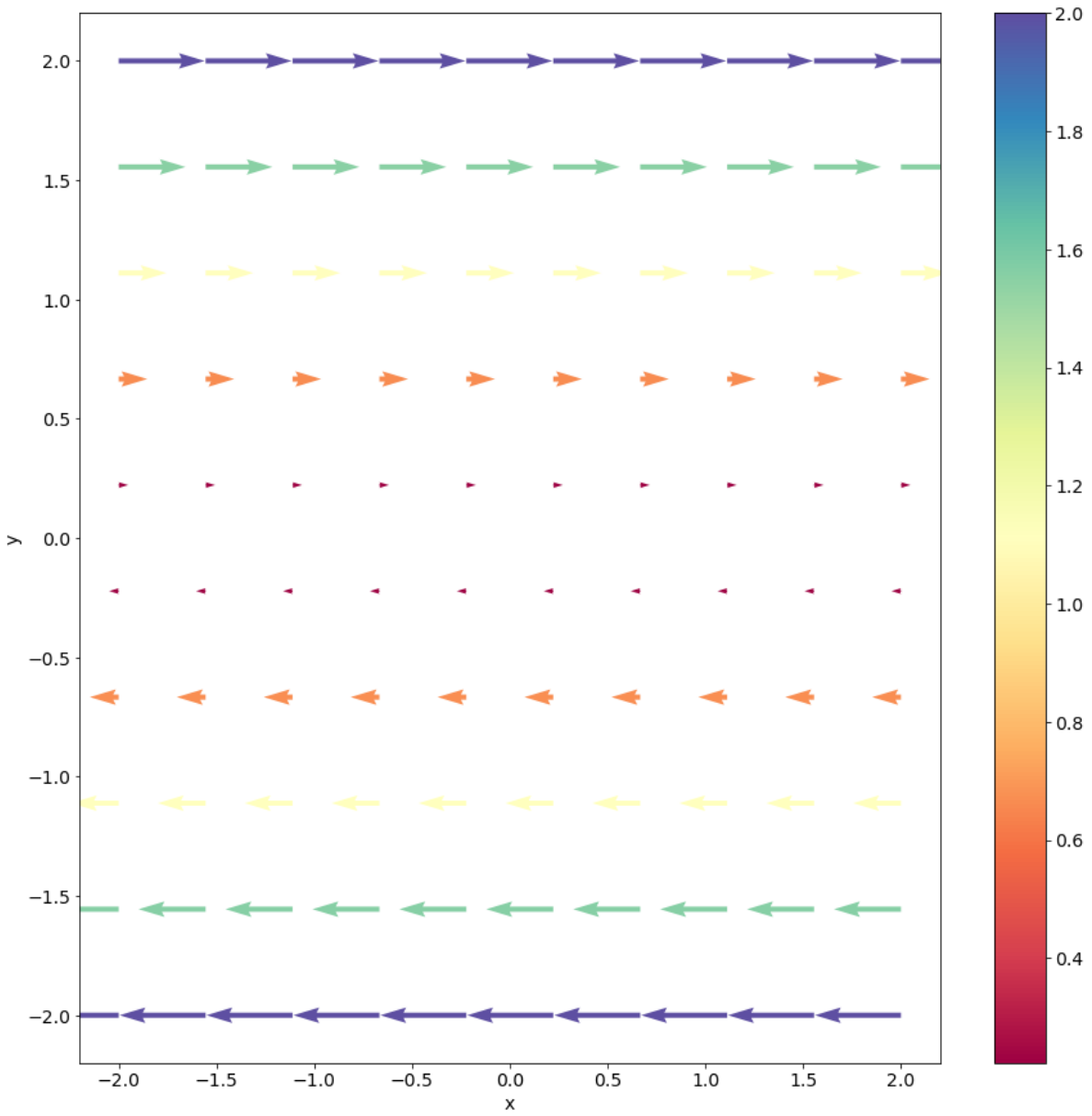
$$\vec{h}(x, y) = \frac{x}{\sqrt{x^2 + y^2}}\hat{i} \frac{y}{\sqrt{x^2 + y^2}}\hat{j}$$

Interpret the physical meaning of the last field.

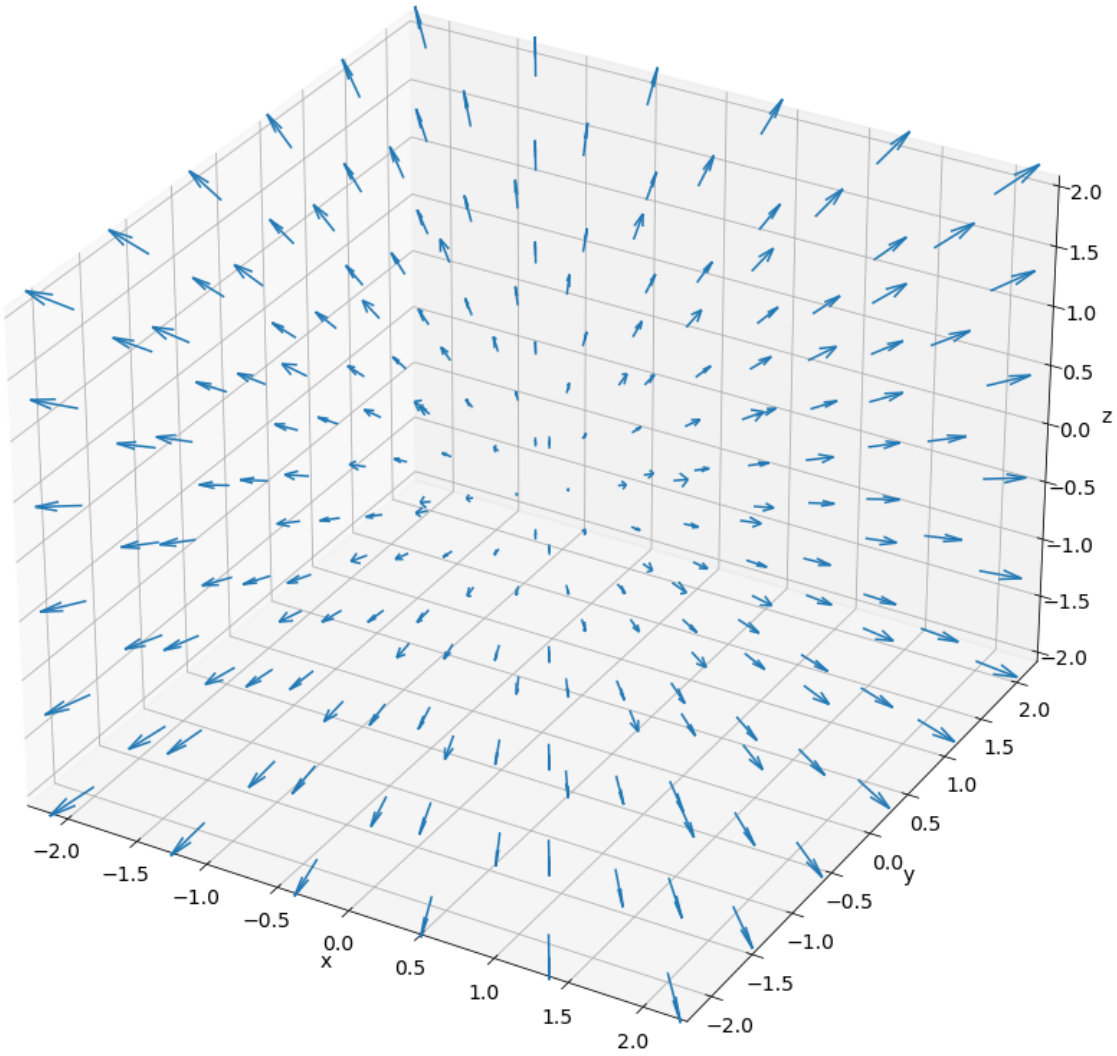
3D Vector Fields

Suppose we wanted to look at a three-dimensional vector field. Here is an example.

In [51]:
```
# Calculate the vector field (source field)
gx = y
gy = 0

colors = np.abs(np.sqrt(gx**2+gy**2))#create colors based on magnitude of vecte

# Plot arrows
plt.quiver(x, y, gx, gy, colors, cmap='Spectral') #cmap options are 'magma', '
plt.colorbar()
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

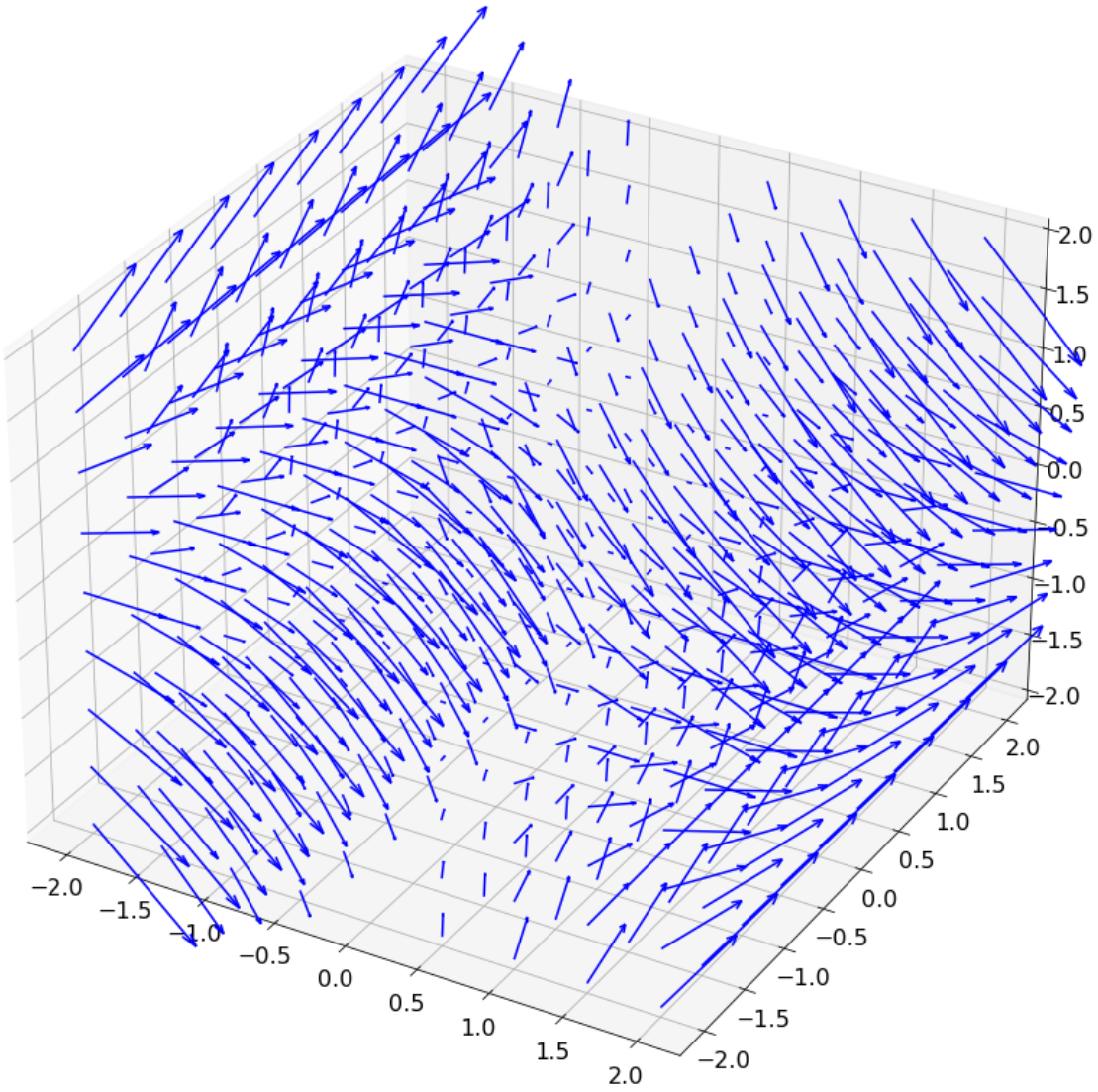


In [29]:
```
# Calculate a grid of points from -2-2 with 5 points in each direction
[x,y,z] = np.meshgrid(np.linspace(-2,2,6),np.linspace(-2,2,6),np.linspace(-2,2

# Calculate the vector field (source field)

fx = x#/(x**2+y**2+z**2)**(3/2)
fy = y#/(x**2+y**2+z**2)**(3/2)
fz = z#/(x**2+y**2+z**2)**(3/2)

# Plot arrows
ax = plt.figure().add_subplot(projection='3d')
ax.quiver(x, y, z, fx, fy, fz, length=0.1, arrow_length_ratio=0.5, cmap='Spect
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_zlabel('z')
plt.show()
```

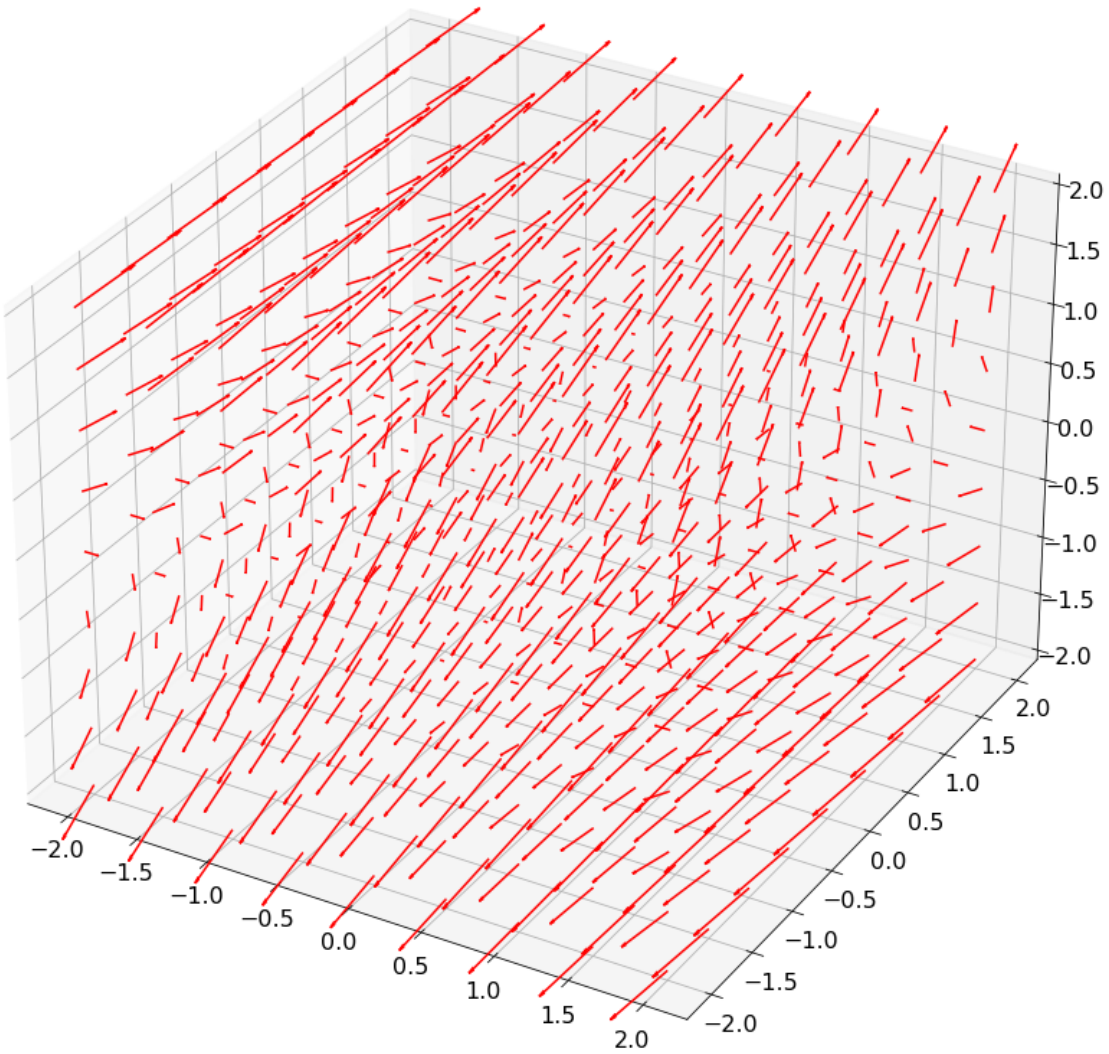


I did this example in class in 2022 while explaining divergence and curl.

In [13]:
```
# Calculate a grid of points from -2-2 with 11 points in each direction
[x,y,z] = np.meshgrid(np.linspace(-2,2,9),np.linspace(-2,2,9),np.linspace(-2,2

# Calculate the vector field (source field)
fx = 3*x**2
fy = x*z
fz = -5*x*z

curlx = -x
curly = 5*z
curlz = z

col = np.abs(np.sqrt(fx**2+fy**2+fz**2))#create colors based on magnitude of v

# Plot arrows
plt.rcParams["figure.figsize"] = (15,15) #Make the plot bigger
ax = plt.figure().add_subplot(projection='3d')
ax.quiver(x, y, z, fx, fy, fz, length=0.05, arrow_length_ratio=0.1, color='b')
#ax.quiver(x, y, z, curlx, curly, curlz, length=0.05, arrow_length_ratio=0.1, 
plt.show()
```



In [14]:
```
# Plot curl arrows
plt.rcParams["figure.figsize"] = (15,15) #Make the plot bigger
ax = plt.figure().add_subplot(projection='3d')
#ax.quiver(x, y, z, fx, fy, fz, length=0.05, arrow_length_ratio=0.1, color='b'
ax.quiver(x, y, z, curlx, curly, curlz, length=0.05, arrow_length_ratio=0.1, c
plt.show()
```



In [ ]: