Deliverable 2: Step 1

Selected Issues:
1) https://github.com/matplotlib/matplotlib/issues/12911
2) https://github.com/matplotlib/matplotlib/issues/11759
3) https://github.com/matplotlib/matplotlib/issues/11746
4) https://github.com/matplotlib/matplotlib/issues/2341
5) https://github.com/matplotlib/matplotlib/issues/8532

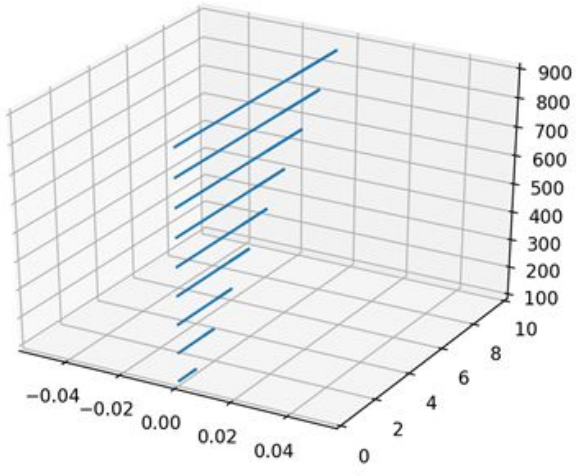
Issues selected for implementation: #11746, #12911, #11759

Issue: #11746

The size of a 3D arrow head plotted with Axes3D.quiver (is disproportional)

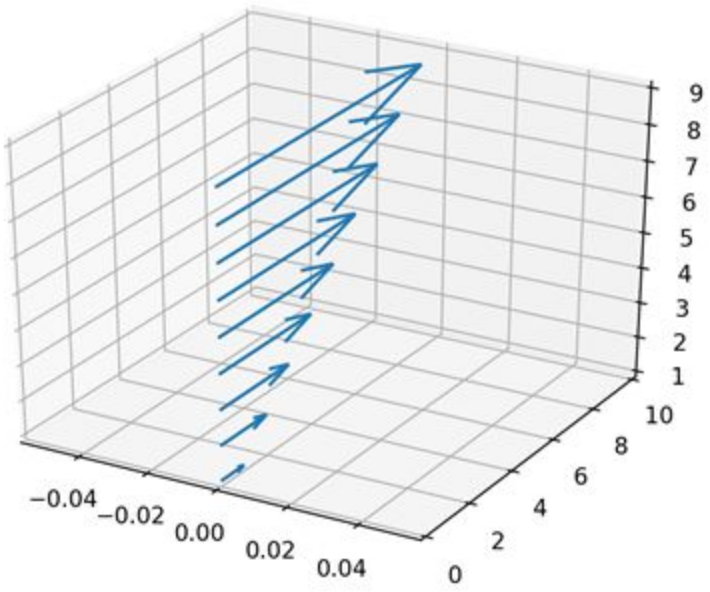
Code for reproduction:

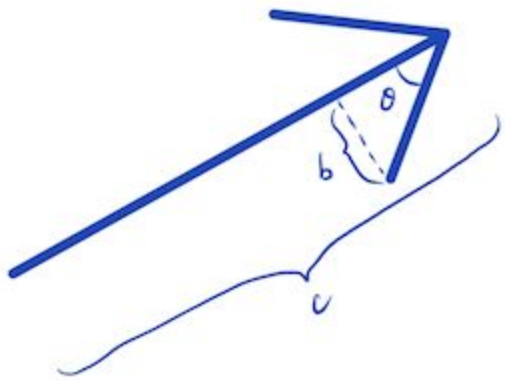
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = np.zeros(10)
y = np.zeros(10)
z = np.arange(10)*100 # remove *100 and the arrow heads will reappear.
dx = np.zeros(10)
dy = np.arange(10)
dz = np.zeros(10)

fig = plt.figure()
ax =
fig.gca(projection='3d')
ax.quiver(x, y, z, dx, dy,
dz)
ax.set_ylim(0,10)

plt.show()
```



Actual outcome:
Expected outcome:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = np.zeros(10)
y = np.zeros(10)
z = np.arange(10) # change in scale on the z axis.
dx = np.zeros(10)
dy = np.arange(10)
dz = np.zeros(10)

fig = plt.figure()
ax = fig.gca(projection='3d')
ax.quiver(x, y, z, dx, dy, dz)
ax.set_ylim(0,10)

plt.show()
```





This bug is likely caused by the construction logic of the arrow itself. As per picture, b, the distance between the diagonal and the line itself, is dependant on c, the length of the arrow. However, between the diagram for actual outcome, c does not change as it is only dependant on the y direction. As a result, the value for b (in the z direction) stays the same. The underlying problem lies in the scale of the z increasing such that the value of b looks very tiny. For example, The value of b might be 1 in the diagram for expected outcome and span over 1 grid line since the scale is divided in intervals of 1, but will span over 1/100 grid lines for the actual outcome as the scale is divided in intervals of 100.

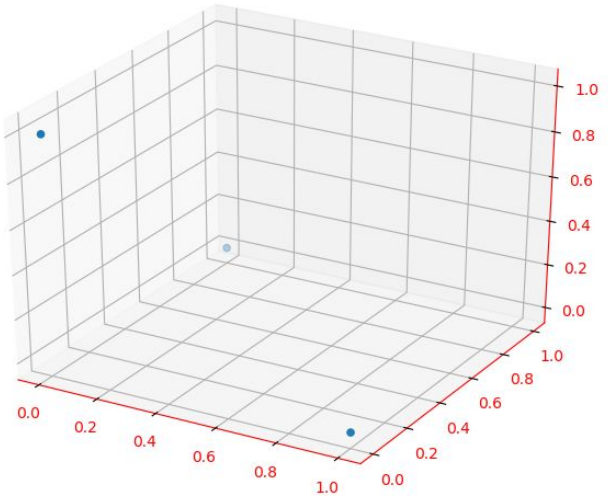
Issue: #12911

Tick mark color cannot be set on Axes3D

Code for reproduction:

```
>>>
from mpl_toolkits.mplot3d import Axes3D
from matplotlib import pyplot as plt

fig = plt.figure()
ax = Axes3D(fig)

ax.scatter((0, 0, 1), (0, 1, 0), (1, 0, 0))
ax.w_xaxis.line.set_color('red')
ax.w_yaxis.line.set_color('red')
ax.w_zaxis.line.set_color('red')
ax.xaxis.label.set_color('red')
ax.yaxis.label.set_color('red')
ax.zaxis.label.set_color('red')
ax.tick_params(axis='x', colors='red')  # only affects
ax.tick_params(axis='y', colors='red')  # tick labels
ax.tick_params(axis='z', colors='red')  # not tick marks

plt.show()
>>>
```

Actual outcome:



Expected Result:

The tick marks on the axis should be red, instead of black.

Possible solution for this problem in matplotlib/lib/mpl_toolkits/mplot3d/axis3d.py
>>> tick.tick1line.set_color(info['tick']['color'])

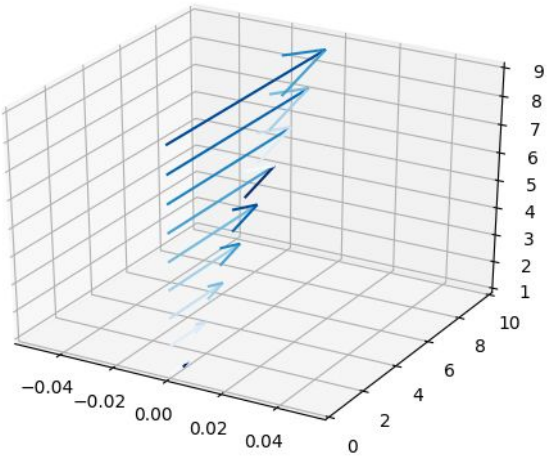
Issue: #11759

The colour of the 3D arrow head does not match that of the arrow body.

Code for reproduction:

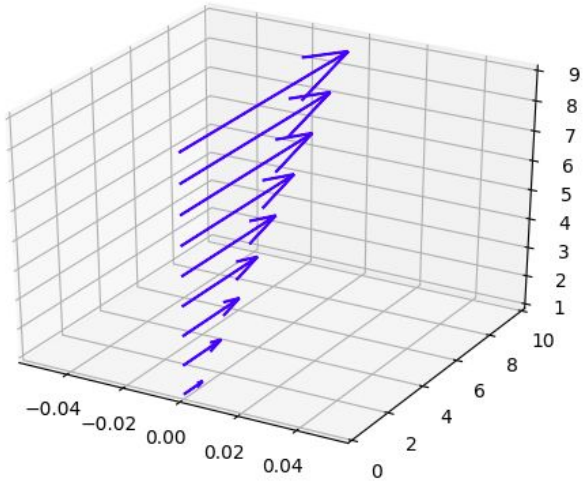
```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = np.zeros(10)
y = np.zeros(10)
z = np.arange(10.)
dx = np.zeros(10)
dy = np.arange(10.)
dz = np.zeros(10)

fig = plt.figure()
ax = fig.gca(projection='3d')

arrow_color = plt.cm.Blues(dy/dy.max())

ax.quiver(x, y, z, dx, dy, dz, colors=arrow_color)
ax.set_ylim(0,10)
plt.show()
```

Actual Outcome:



It appears as though the line “arrow_color = plt.cm.Blues(dy/dy.max())” might be the cause of this bug. The reason for this suspicion is because when simply assigning “arrow_color” to be “Blue”, we get the follow outcome:

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

x = np.zeros(10)
y = np.zeros(10)
z = np.arange(10.)
dx = np.zeros(10)
dy = np.arange(10.)
dz = np.zeros(10)

fig = plt.figure()
ax = fig.gca(projection='3d')

arrow_color = “Blue”

ax.quiver(x, y, z, dx, dy, dz, colors=arrow_color)
ax.set_ylim(0,10)
plt.show()
```



Notice that in this instance we get an arrow head and body with matching colours. Through further investigation we will explore the impact “plt.cm.Blues(dy/dy.max())” has on the code responsible for assigning colour to the arrow head and body. Understanding the impact that this line has on the code base will be a good place to start in terms of understanding where fixes could be made. Victor will be tasked with further investigating this bug. Since the exact location of this bug has not been found, allocating group members to the implementation and testing of this issue has yet to be determined. The size and perceived complexity to fix this bug will be the main factor when allocating group members.