EGR 103L – Fall 2019

Laboratory 8 - Surfaces

Marcus Deansd (md374)
Lab Section 4, Tuesdays 11:45-2:35
17 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard in completing every part of this assignment. I understand that a violation of any part of the Standard on any part of this assignment can result in failure of this assignment, failure of this course, and/or suspension from Duke University.

Contents

List of Figures

1 Chapra 2.22

The graph is smoother at the beginning of both curves with comparison to the end. This is due to the spacing of the data points (originating in the initial domain function). At the start, these data points are more dense and thus when the 'dots' of the data points are connected, they are so close together so as to appear as a smooth line. Near the end of both curves, the size of the curve is larger so the distance is increasing while the number of data points remains constant, so the density goes down for that interval. This lower density means that the 'dots' of the data points, when connected, create a perceptibly non-smooth line as they are too far apart. This problem would be rectified by increasing the number of points in the domain, either in general or over that specific area of the domain.

2 Chapra 3.9

This graph demonstrates that the velocity increases when either heigh of width increases, or both. There is direct relationship between them. An increase in width for a fixed height seems to more significantly increase the velocity compared to an increase in height for a fixed width; this was determined due to the increased number of striated colour bands on the right side of the graph (corresponding to changes in height) compared to the left side of the graph (corresponding to change in width). Each comparison was made while holding the other type constant. Regardless, both were found to have significant effect on velocity with the velocity being maximized when both height and width were at their maximum values. Furthermore, it was observed that **both** affected velocity to a large extent, as opposed to one factor having a large effect and the other not corresponding to significant change in velocity. This overall makes sense considering the equation used to obtain this result (specifically the roots).

3 Chapra 15.5

Strong relationships were observed between the temperature and chlorie concentration and the oxygen concentration. Namely, for any given temperature, the oxygen concentration decreased as chloride concentration went up. Therefore, the chloride concentration has an inverse relationship with oxygen concentration. Temperature was also found to have an inverse relationship with chloride concentration. As the temperature decreased for a given chloride concentration, the oxygen concentration rapidly increased. Therefore, the maximum oxygen concentration would be found at a temperature and chloride concentration of 0, while the lowest oxygen concentration would result from the greatest chloride concentration and temperature ($20g/L$ and 30 Celsius, respectively).

4 Chapra 15.6

Using a planar model:

$$\hat{OC}(c,T) = \sum_m a_m \phi_m(c,T) = a_0(c^0T^0) + a_1(c^1) + a_2(T^1)$$

the model coefficients and other required values are:

a_0	a_1	a_2	r^2	$OC(15,12)$ g/ml	ϵ_t
13.52	-0.1049	-0.2012	9.659e-01	9.533	4.88%

The graph of the residuals could most likely be modeled effectively using equations for parabolas. With the graph roughly forming the shape of a 'U' with its peaks, parabolas could be used to approximate this to various extents depending on the region. In terms of trends, the residuals were highest when the chloride concentration and temperature were lowest (around 0 in particular) and when the chloride concentration and temperature were highest (specifically near 30). This two regions serve as key areas of examination in order to produce equations that could accurately model this.

5 Chapra 15.7

Using a first-order model for c and a third-order model for T:

$$\hat{OC}(c,T) = \sum_m a_m \phi_m(c,T) = a_0(c^0T^0) + a_1(c^1) + a_2(T^3) + a_3(T^2) + a_4(T^1)$$

the model coefficients and other required values are:

s

a_0	a_1	a_2	a_3	a_4	r^2	$OC(15, 12)$ g/ml	ϵ_t
1.403e01	-1.049e-1	-4.370	5.744e-03	-3.364e-01	9.877e-01	9.168	0.856%

At first glance, the residuals seemed difficult to model using simple, common equations. With the directionality of the peaks and lows of the residuals, it seemed much more complex than the prior situation. The trend of higher residuals with maximums for chloride concentration and temperature, and minimums, held true. These areas still had high residuals. However, the residuals were dramatically reduced in the areas where the two factors were moving inversely of one another - when chloride concentration was high but temperature was low, and vice versa, the residuals were minimized in this model. Again, developing an accurate model would be difficult. Based on online research, it seemes that a hyperbolic paraboloid would be the srtructure created by the equations to model this graph, which would allow both the minima and maxima of the residuals to be incorporated into the model.

6 Sphere

Transformation equations:

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

A Codes

A.1 Chapra 2.22

```python
# -*- coding: utf-8 -*-
"""
Chapra 2.22
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d

tmodel = np.arange(0.0,(np.pi)*6, (np.pi)/64)
x = np.array(tmodel*(np.cos(6*tmodel)))
y = np.array(tmodel*(np.sin(6*tmodel)))
z = np.array(tmodel)

fig = plt.figure()
fig.set_size_inches(6,8)

# fig, ax = plt.subplots(num=1,clear=True)
ax = fig.add_subplot(2,1,1)
ax.plot(x, y, 'r-')
ax.grid()
ax.set(xlabel = 'x',ylabel='y')
ax.axis('equal')

# fig = plt.figure()
ax = fig.add_subplot(2,1,2, projection='3d')
ax.plot3D(x, y, z, 'cyan')
ax.set(xlabel='x', xticks=[-20,0,20], ylabel='y', yticks=[-20,0,20], zlabel='z',
    zticks=[0,10,20])

fig.tight_layout()
fig.savefig("Chapra_2.22_Plot.png")
fig.savefig("Chapra_2.22_Plot.eps")
```

A.2 Chapra 3.9

```python
"""
Chapra 3.9
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d
from matplotlib import cm

n = 0.02
s = 0.001

(x, y) = np.meshgrid(np.linspace(0.01,20,40), np.linspace(0.01,5,41))
z = ((np.sqrt(s))/n)*(((x*y)/(x+(2*y)))**(2/3))

fig = plt.figure(num=1,clear=True)
ax = plt.axes(projection='3d')
makeit = ax.plot_surface(x,y,z, cmap=cm.viridis_r)
ax.set(title="Velocity Using Manning's Equation",
       xlabel='Width $B$, $m$', xticks=[0,5,10,15, 20],
       ylabel='Height $H$, $m$', yticks=[0,1,2,3,4,5],
       zlabel='Velocity $U$, $m/s$', zticks=[0.5,1,1.5,2,2.5,3,3.5])
ax.view_init(elev=35,azim=-135)
fig.colorbar(makeit)
fig.tight_layout()
fig.savefig("Chapra_3.9_Plot.png")
fig.savefig("Chapra_3.9_Plot.eps")
```

A.3 Chapra 15.5

```python
"""
Chapra 15.5
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d
from matplotlib import cm

valu = np.loadtxt("chapra_p15_5.dat")
ix = valu[:,0].copy()
iy = valu[:,1].copy()
iz = valu[:,2].copy()

x = np.reshape(ix,(7,3))
y = np.reshape(iy,(7,3))
z = np.reshape(iz,(7,3))

fig = plt.figure(num=1,clear=True)
ax = plt.axes(projection='3d')
ax.plot_surface(x,y,z, cmap=cm.winter)
ax.set(xlabel='Chloride Concentration $c$, $g/L$', xticks=[0,10,20],
       ylabel='Temperature $T$, $C$', yticks=[0,5,10,15,20,25,30],
       zlabel='Oxygen Concentration $OC$, $mg/L$')
ax.view_init(elev=6,azim=-152)
fig.tight_layout()
fig.savefig("Chapra_15.5_Plot.png")
fig.savefig("Chapra_15.5_Plot.eps")
```

A.4 Chapra 15.6

```python
"""
Chapra 15.6
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d
from matplotlib import cm
from fitting_common import *

"""
Use multiple linear regression to derive a predictive
equation for dissolved oxygen concentration as a function of temperature and chloride based on the data from
Table P15.5. Use the equation to estimate the concentration
of dissolved oxygen for a chloride concentration of 15 g/L at
T = 12 °C. Note that the true value is 9.09 mg/L. Compute the percent relative error for your prediction.
Explain possible causes for the discrepancy.
"""

valu = np.loadtxt("chapra_p15_5.dat")
x = valu[:,0].copy()
y = valu[:,1].copy()
z = valu[:,2].copy()

xmat = np.reshape(x,(7,3))
ymat = np.reshape(y,(7,3))
zmat = np.reshape(z,(7,3))

# %% Perform calculations
def zfun(xe, ye, coefs):
    return (coefs[0]*(xe**0)*ye**0 + coefs[1]*xe**1 + coefs[2]*ye**1)

# %% Reshape data for block matrices
xv = np.reshape(x, (-1, 1))
yv = np.reshape(y, (-1, 1))
zv = np.reshape(z, (-1, 1))
phi_mat = np.block([[(xv**0)*(yv**0), xv**1, yv**1]])
pvec = np.linalg.lstsq(phi_mat, zv, rcond=None)[0]

# %% First Graph Values
(xi, yi) = np.meshgrid(np.linspace(0,20,7), np.linspace(0,30,19))
zi = zfun(xi, yi, pvec)
zest = zfun(xv, yv, pvec)
#%% First Graph
fig = plt.figure(num=1,clear=True)
fig.set_size_inches(6,4)
```

```python
ax = plt.axes(projection='3d')
ax.plot_surface(xi,yi,zi, cmap=cm.twilight)
ax.set(xlabel='Chloride Concentration $c$, $g/L$', xticks=[0,10,20],
       ylabel='Temperature $T$, $C$', yticks=[0,5,10,15,20,25,30],
       zlabel='Oxygen Concentration $OC$, $mg/L$')
ax.view_init(elev=6,azim=-152)
fig.tight_layout()
fig.savefig("Chapra_15.6_Plot_Alpha.png")
fig.savefig("Chapra_15.6_Plot_Alpha.eps")

#%% Second Graph
fig = plt.figure(num=2, clear=True)
fig.set_size_inches(6,4)
ax = fig.add_subplot(1, 1, 1, projection='3d')

ax.plot_surface(xmat, ymat, zmat-(np.reshape(zest,(7,3))), cmap=cm.autumn)
ax.set(xlabel='Chloride Concentration $c$, $g/L$', xticks=[0,10,20],
       ylabel='Temperature $T$, $C$', yticks=[0,5,10,15,20,25,30],
       zlabel='Residual, $mg/L$')
fig.tight_layout()
fig.savefig("Chapra_15.6_Plot_Bravo.png")
fig.savefig("Chapra_15.6_Plot_Bravo.eps")
#%% Get and Print Statistics
st, sr, r2 = calc_stats(zv, zest,False)
print("The coefficients are: ")
print(pvec)
print("The OC Estimate when c = 15 g/L and T is 12 degrees is {:.3e}".format(float(
    zfun(15,12,pvec))))
print(abs((((float(zfun(15,12,pvec))))-9.09)/9.09))
print("St - Sum of Residual Squares: {:.3e}".format(st))
print("Sr - Sum of Squares of Estimate Residuals: {:.3e}".format(sr))
print("R Squared Value: {:.3e}".format(r2))
```

A.5 Chapra 15.7

```python
"""
Chapra 15.6
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d
from matplotlib import cm
from fitting_common import *

"""
Use multiple linear regression to derive a predictive
equation for dissolved oxygen concentration as a function of temperature and chloride based on the data from
Table P15.5. Use the equation to estimate the concentration
of dissolved oxygen for a chloride concentration of 15 g/L at
T = 12 °C. Note that the true value is 9.09 mg/L. Compute the percent relative error for your prediction.
Explain possible causes for the discrepancy.
"""

valu = np.loadtxt("chapra_p15_5.dat")
x = valu[:,0].copy()
y = valu[:,1].copy()
z = valu[:,2].copy()

xmat = np.reshape(x,(7,3))
ymat = np.reshape(y,(7,3))
zmat = np.reshape(z,(7,3))

# %% Perform calculations
def zfun(xe, ye, coefs):
    return (coefs[0]*(xe**0)*ye**0 + coefs[1]*xe**1 + coefs[2]*ye**3 +
              coefs[3]*ye**2 + coefs[4]*ye**1)

# %% Reshape data for block matrices
xv = np.reshape(x, (-1, 1))
yv = np.reshape(y, (-1, 1))
zv = np.reshape(z, (-1, 1))
phi_mat = np.block([[(xv**0)*(yv**0), xv**1, yv**3, yv**2, yv**1]])
pvec = np.linalg.lstsq(phi_mat, zv, rcond=None)[0]

# %% First Graph Values
(xi, yi) = np.meshgrid(np.linspace(0,20,7), np.linspace(0,30,19))
zi = zfun(xi, yi, pvec)
zest = zfun(xv, yv, pvec)
#%% First Graph
fig = plt.figure(num=1,clear=True)
```

```python
fig.set_size_inches(6,4)
ax = plt.axes(projection='3d')
ax.plot_surface(xi,yi,zi, cmap=cm.twilight)
ax.set(xlabel='Chloride Concentration $c$, $g/L$', xticks=[0,10,20],
       ylabel='Temperature $T$, $C$', yticks=[0,5,10,15,20,25,30],
       zlabel='Oxygen Concentration $OC$, $mg/L$')
ax.view_init(elev=6,azim=-152)
fig.tight_layout()
fig.savefig("Chapra_15.7_Plot_Alpha.png")
fig.savefig("Chapra_15.7_Plot_Alpha.eps")

#%% Second Graph
fig = plt.figure(num=2, clear=True)
fig.set_size_inches(6,4)
ax = fig.add_subplot(1, 1, 1, projection='3d')

ax.plot_surface(xmat, ymat, zmat-(np.reshape(zest,(7,3))), cmap=cm.autumn)
ax.set(xlabel='Chloride Concentration $c$, $g/L$', xticks=[0,10,20],
       ylabel='Temperature $T$, $C$', yticks=[0,5,10,15,20,25,30],
       zlabel='Residual, $mg/L$')
fig.tight_layout()
fig.savefig("Chapra_15.7_Plot_Bravo.png")
fig.savefig("Chapra_15.7_Plot_Bravo.eps")
#%% Get and Print Statistics
st, sr, r2 = calc_stats(zv, zest,False)
print("The coefficients are: ")
print(pvec)
print("The OC Estimate when c = 15 g/L and T is 12 degrees is {:.3e}".format(float(zfun(15,12,pvec))))
print(abs((((float(zfun(15,12,pvec))))-9.09)/9.09))
print("St - Sum of Residual Squares: {:.3e}".format(st))
print("Sr - Sum of Squares of Estimate Residuals: {:.3e}".format(sr))
print("R Squared Value: {:.3e}".format(r2))
```

A.6 Sphere

```python
"""
Sphere
Marcus Deans
11 November 2019

I understand and have adhered to all the tenets of the Duke Community Standard
in creating this code.
Signed: [md374]
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import axes3d
from mpl_toolkits import mplot3d

(phi, theta) = np.meshgrid((np.linspace(0,np.pi,20)), (np.linspace(0,2*np.pi,40)))
x = ((np.sin(phi))*(np.cos(theta)))
y = ((np.sin(phi))*(np.sin(theta)))
z = np.cos(phi)

fig = plt.figure(num=1,clear=True)
fig.set_size_inches(6,6)
ax = plt.axes(projection='3d')
ax.plot_wireframe(x,y,z, color='coral')
ax.set(xlabel='x', ylabel='y', zlabel='z')
fig.tight_layout()
fig.savefig("Sphere.png")
fig.savefig("Sphere.eps")
```

B Figures

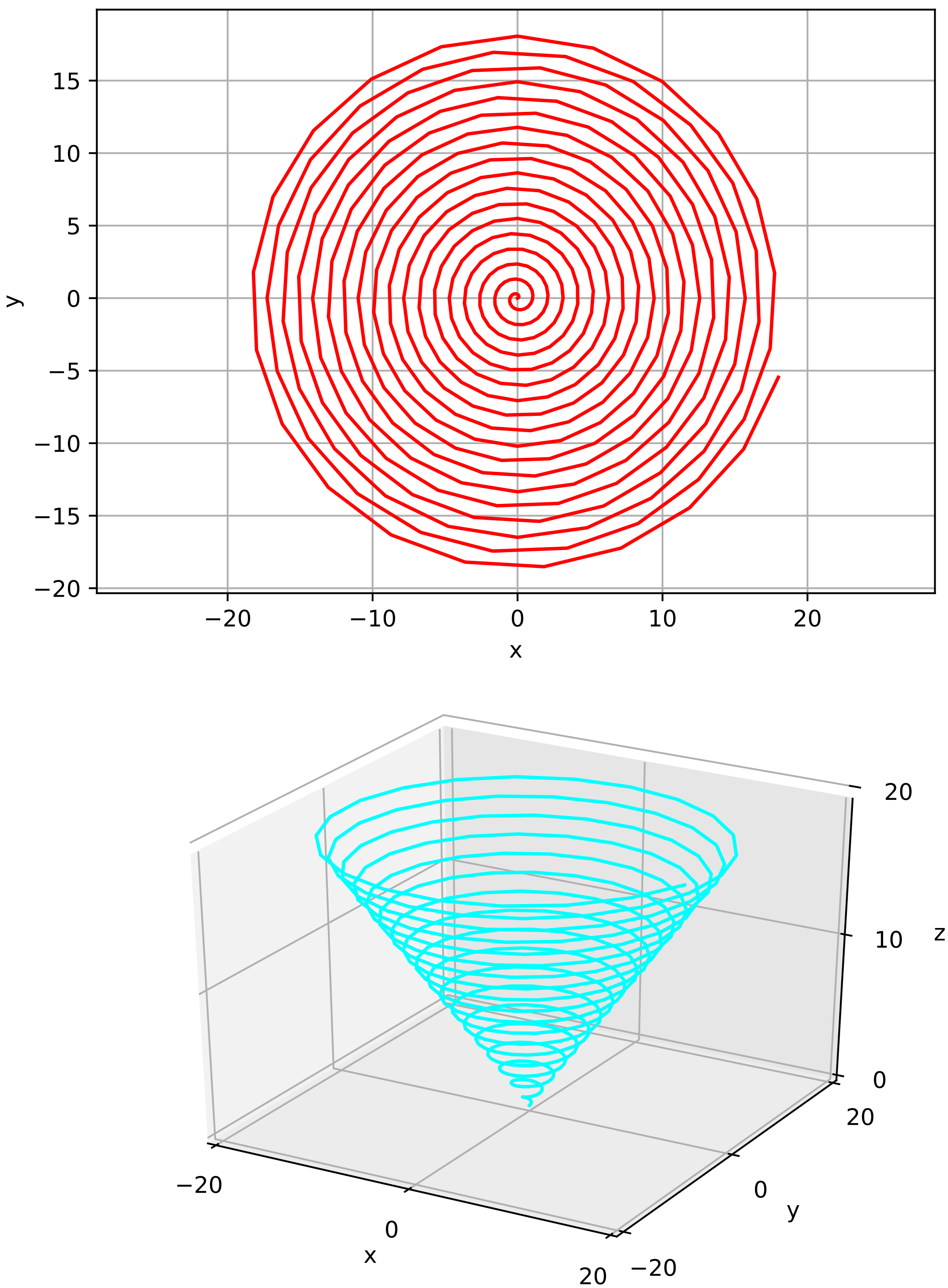

Figure 1: Figure for Chapra 2.22

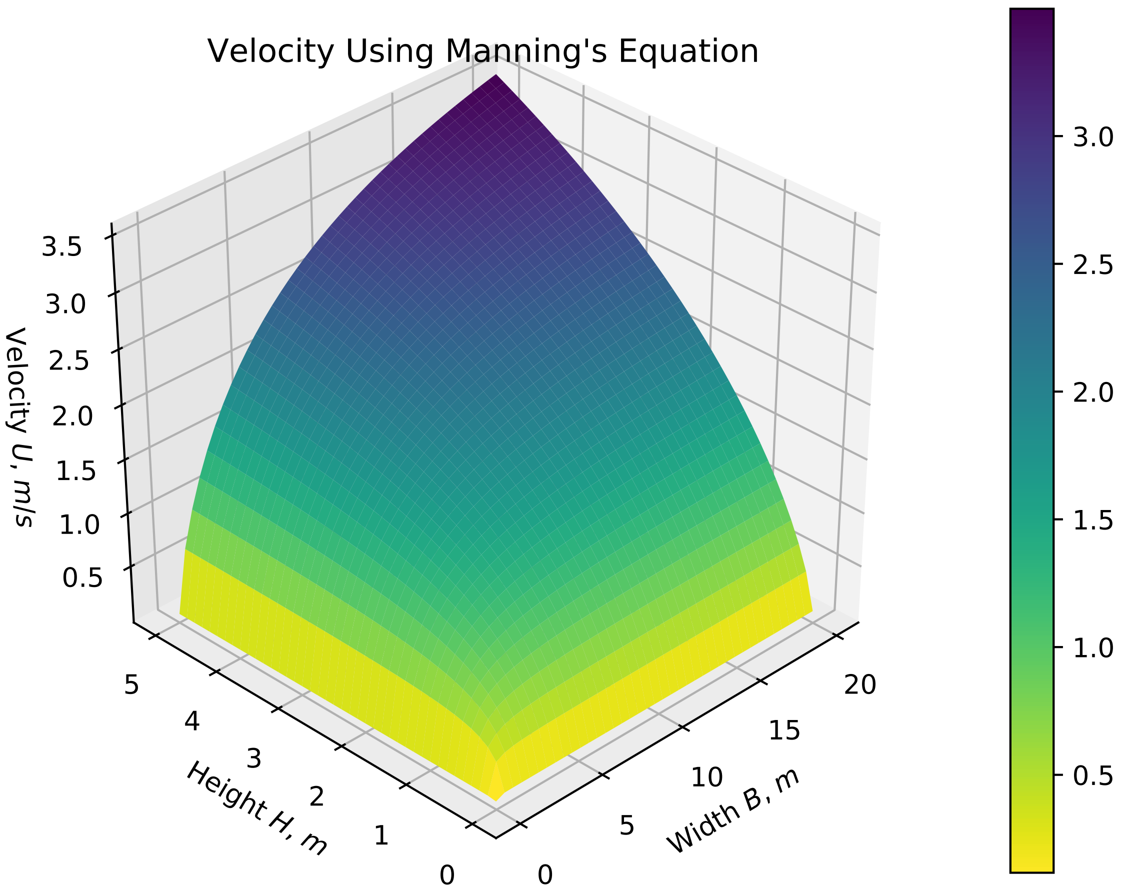

Figure 2: Figure for Chapra 3.9

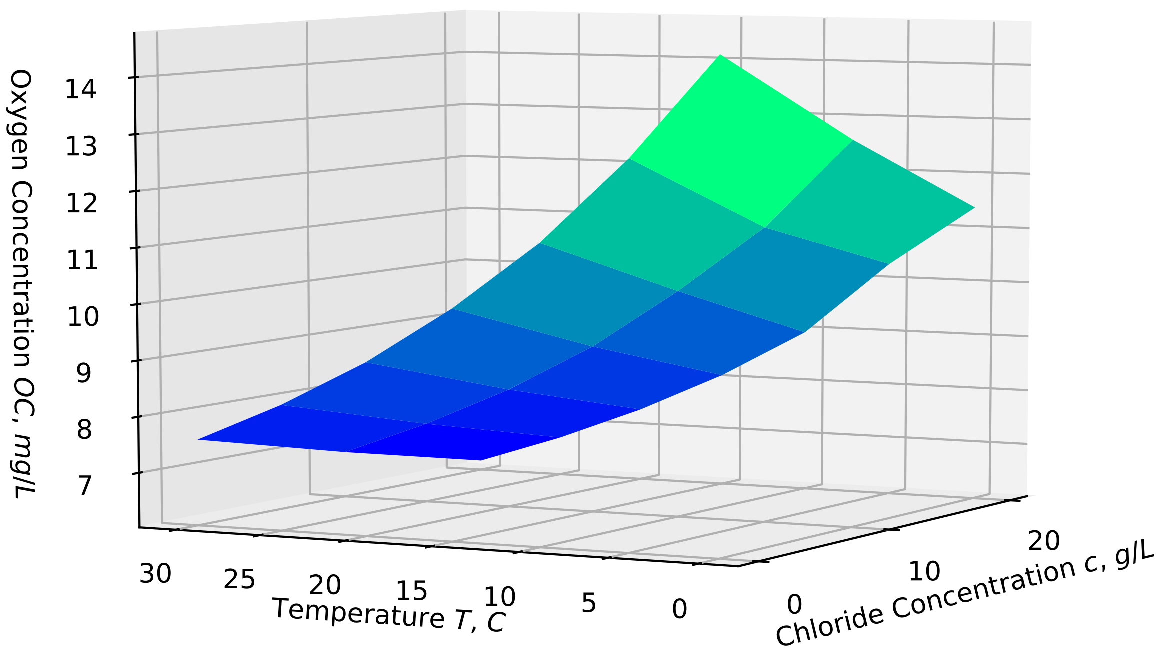

Figure 3: Figure for Chapra 15.5

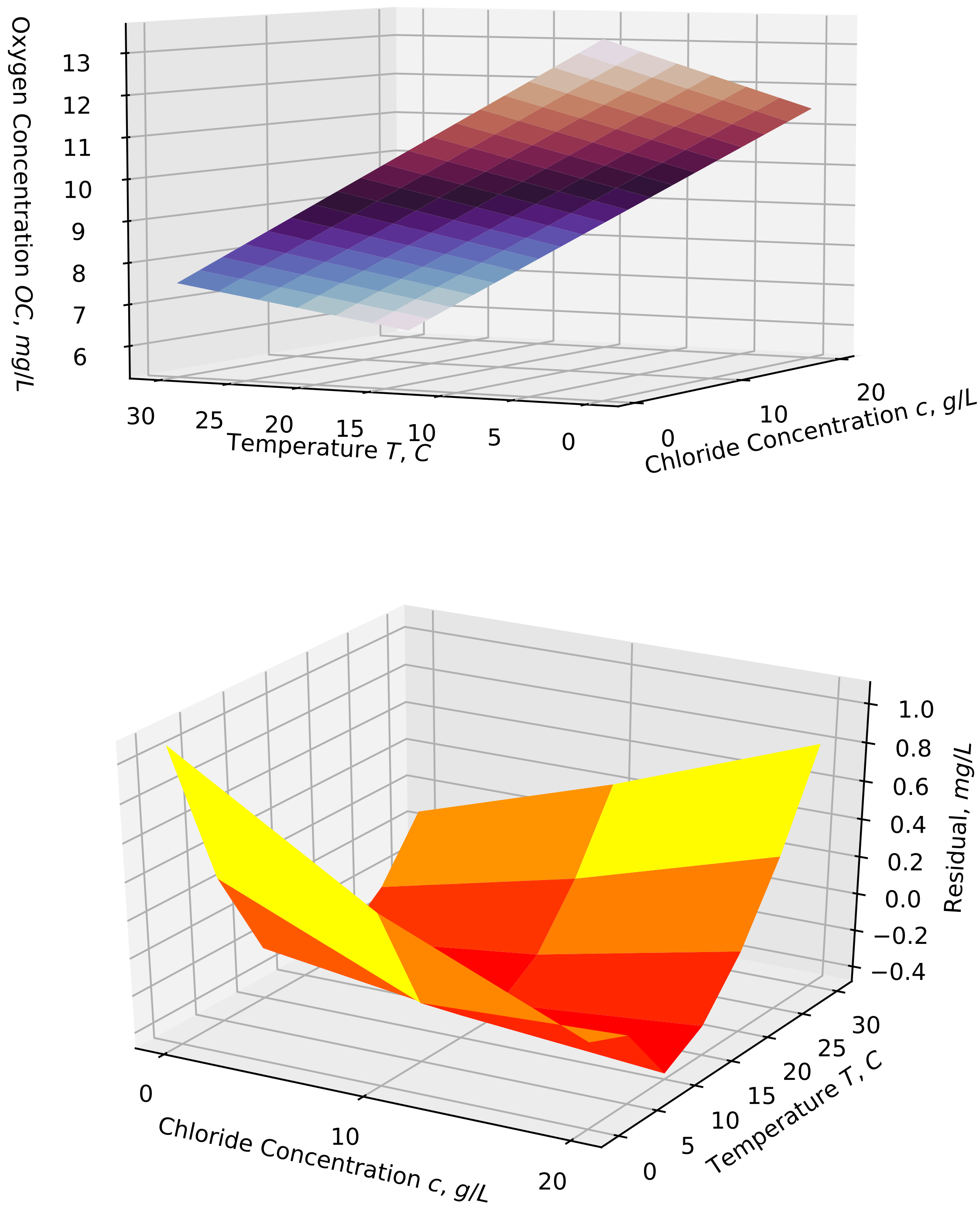

Figure 4: Figures for Chapra 15.6

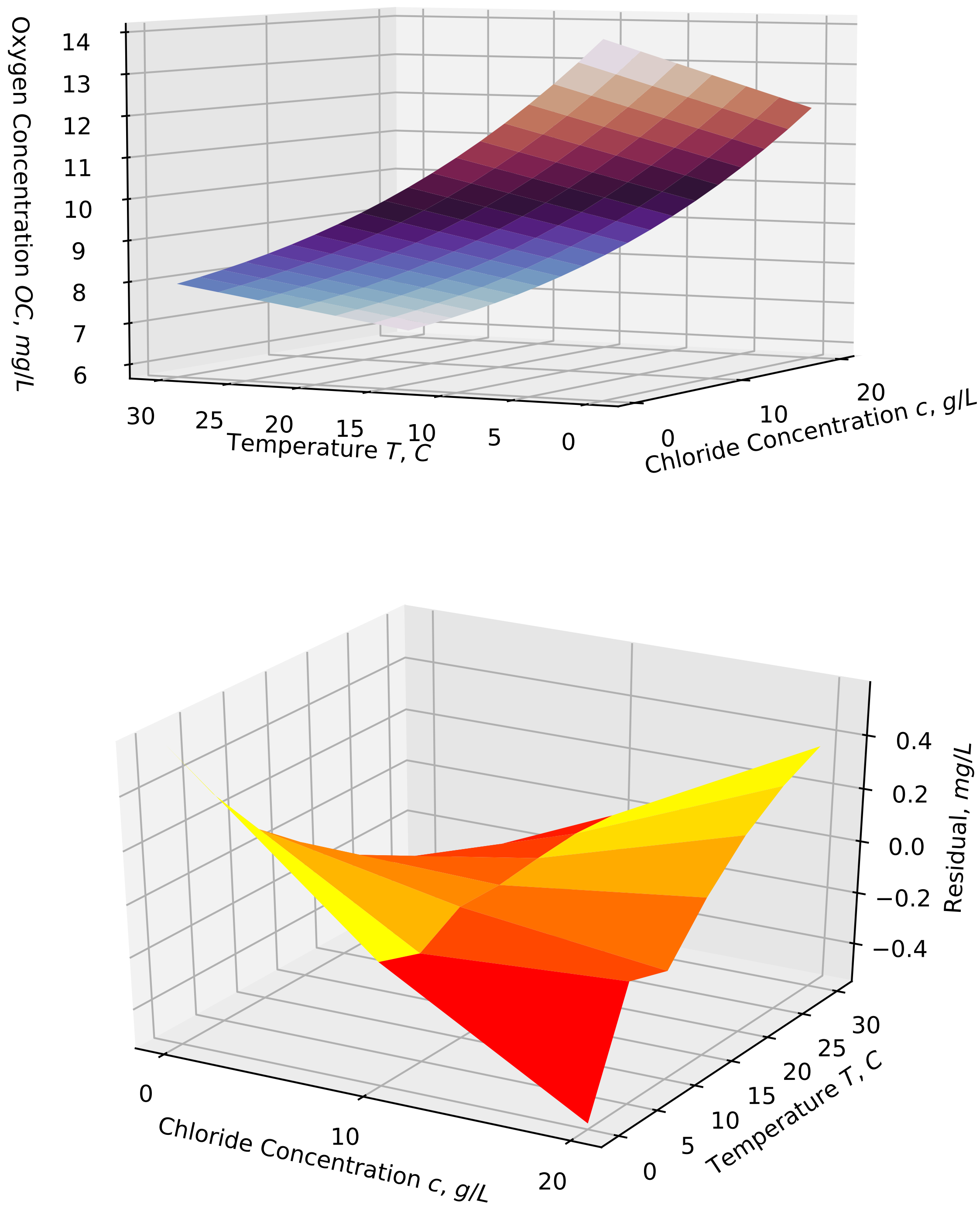

Figure 5: Figures for Chapra 15.7

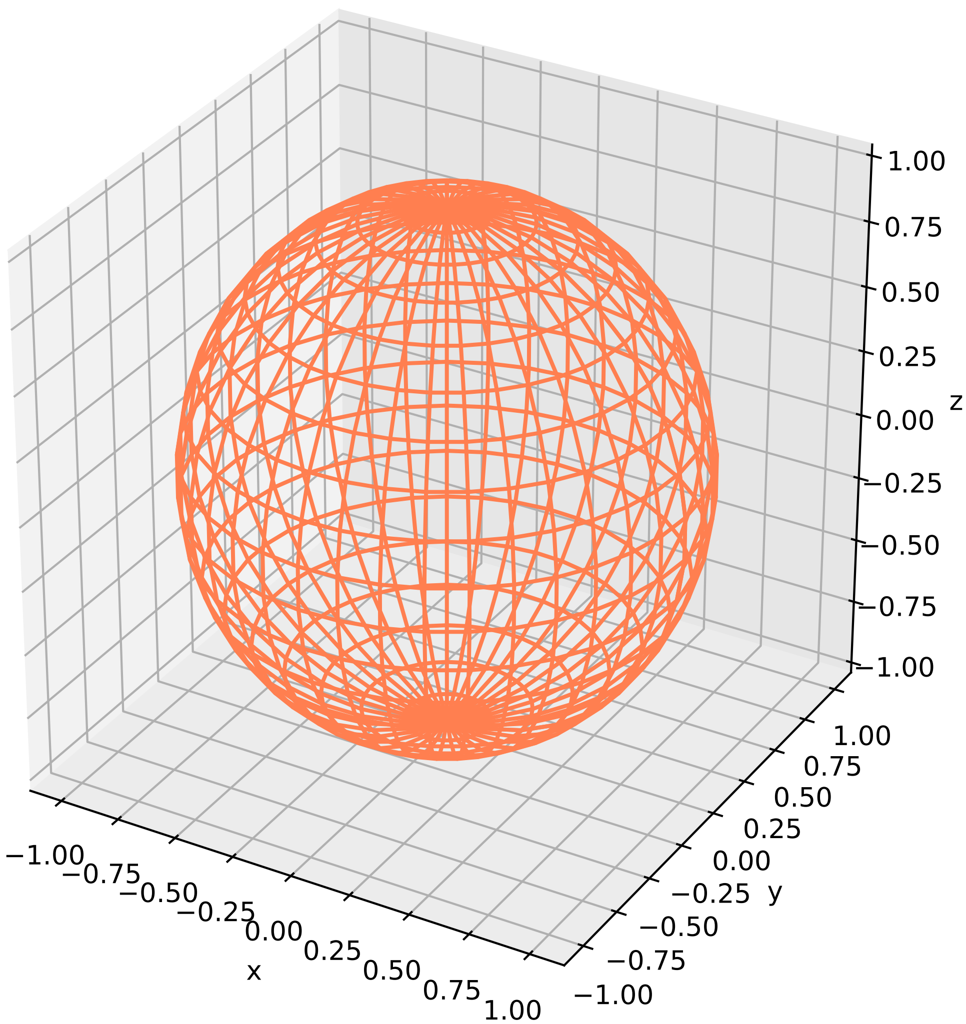

Figure 6: Figure for Sphere