Oppgave 1d

import numpy as np

import matplotlib.pyplot as plt

fig = plt.figure(figsize =(10, 10))

ax = fig.add\_subplot()

t = np.linspace(0, 2\*np.pi, 200)

x = np.cos(t)

y = t\*np.sin(t)

ax.plot(x,y, label="A parametric curve")

ax.legend()

plt.show()

ax = plt.figure(figsize =(10, 10))\

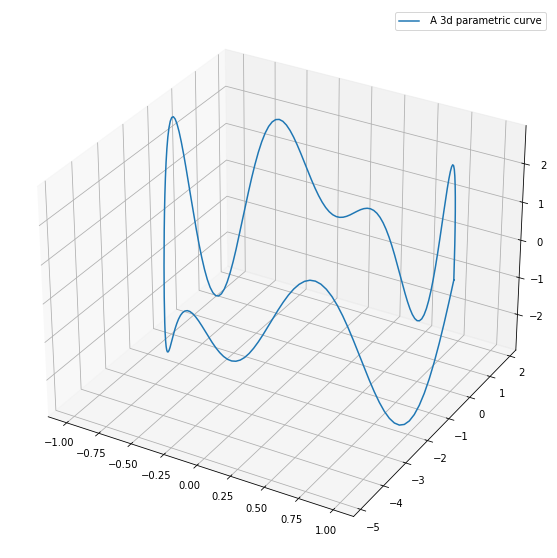
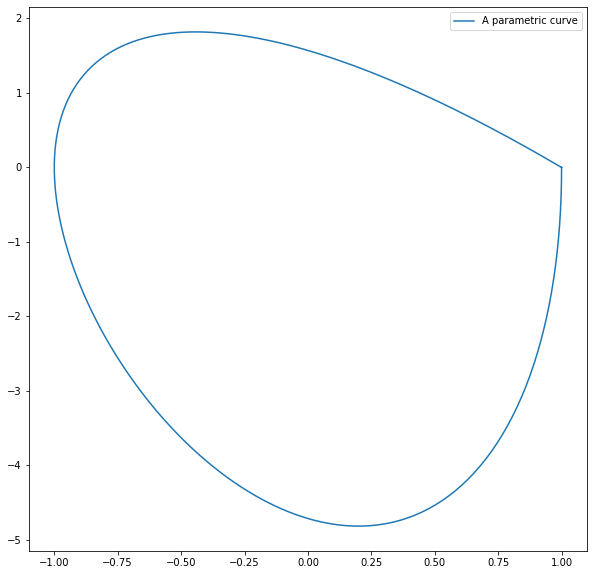
.add\_subplot(projection="3d")

z = np.sin(5\*t)\*3\*np.cos(t)

ax.plot(x, y, z, label=" A 3d parametric curve")

ax.legend()

plt.show()



Oppgave 2b

import numpy as np

import matplotlib.pyplot as plt

fig = plt.figure(figsize =(10, 10))

ax = fig.add\_subplot()

t = np.linspace(-1, 1, 100)

x = t

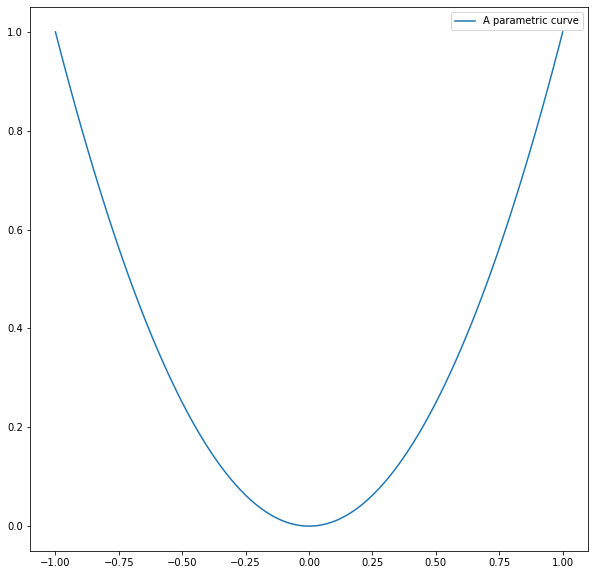
y = t\*\*2 #Since (x,y) go goes thru the point (-1,1), (0,0) and (1,1)

#The formula for y will by y\*\*2.

ax.plot(x, y, label="A parametric curve")

ax.legend()

plt.show()



Oppgave 3b

import numpy as np

import matplotlib.pyplot as plt

def f(x, y):

return (x\*\*3 - 3\*x\*y\*\*2) # Monkeysaddle function

# U = [-1,1] X [-1,1]

x = np.linspace(-1, 1, 40)

y = np.linspace(-1, 1, 40)

x, y = np.meshgrid(x, y)

X = x

Y = y

Z = f(x, y)

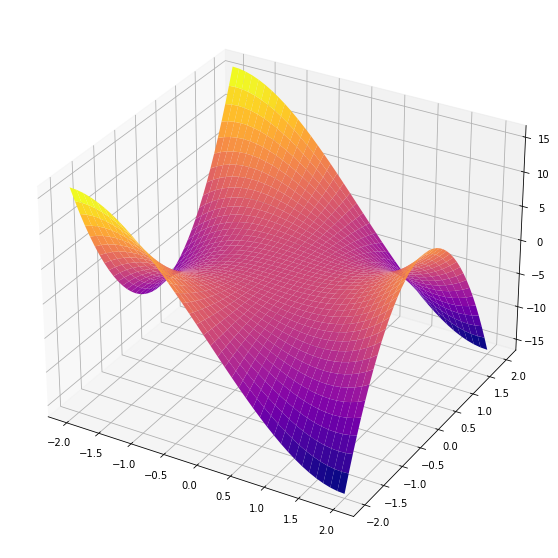
fig = plt.figure(figsize = (10, 10))

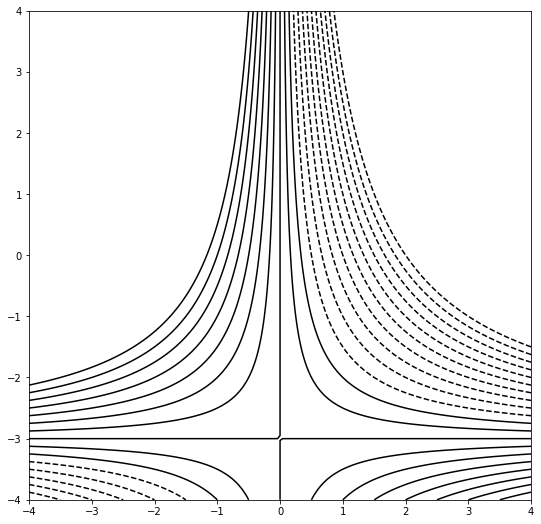
ax = plt.axes(projection = '3d')

ax.plot\_surface(X, Y, Z, cmap = 'plasma', \

edgecolor = 'none')

plt.show()



Oppgave 7c

import numpy as np

import matplotlib.pyplot as plt

x = np.linspace(-4.0, 4.0, 100)

y = np.linspace(-4.0, 4.0, 100)

u, v = np.meshgrid(x, y)

w = 1 -u\*2 -v\*3 -u + 3\*v -u\*v #Calculates "Z" values ploting

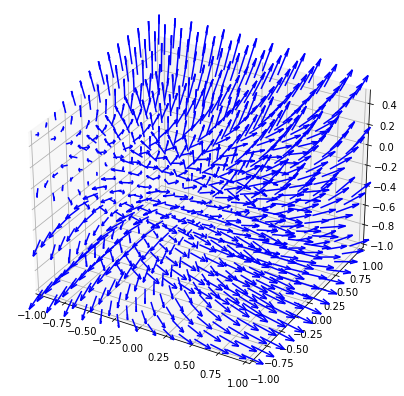
fig = plt.figure(figsize =(9, 9))

ax = fig.add\_subplot()

levels = np.arange(-5, 5, 0.5) # Decides where to draw level lines 0.5 is spacing -5,5 is the range

cont = ax.contour(u, v, w, levels, colors = "black")

plt.show()



import numpy as np

import matplotlib.pyplot as plt

delta\_z = 0.5 # step length in z-direction

x = np.arange(-1, 1, 0.15) #Decides how many points and the range to distribute them on.

y = np.arange(-1, 1, 0.15)

z = np.arange(-1, 1, delta\_z)

# 3d point mesh

x, y, z = np.meshgrid(x, y, z)

# the vector field

u = np.sin(x)\*np.cos(y) # array with x-coord.

v = np.cos(x+z) # array with y-coord.

w = np.sin(y+z) # array with z-coord.

# set up coordinate system

fig = plt.figure(figsize =(7, 7))

ax = fig.add\_subplot(projection="3d")

# plot vector field

ax.quiver(x, y, z, u, v, w, color = "blue", length = 0.2)

# determine interval for plot

ax.axis([-1, 1, -1, 1])

plt.show()