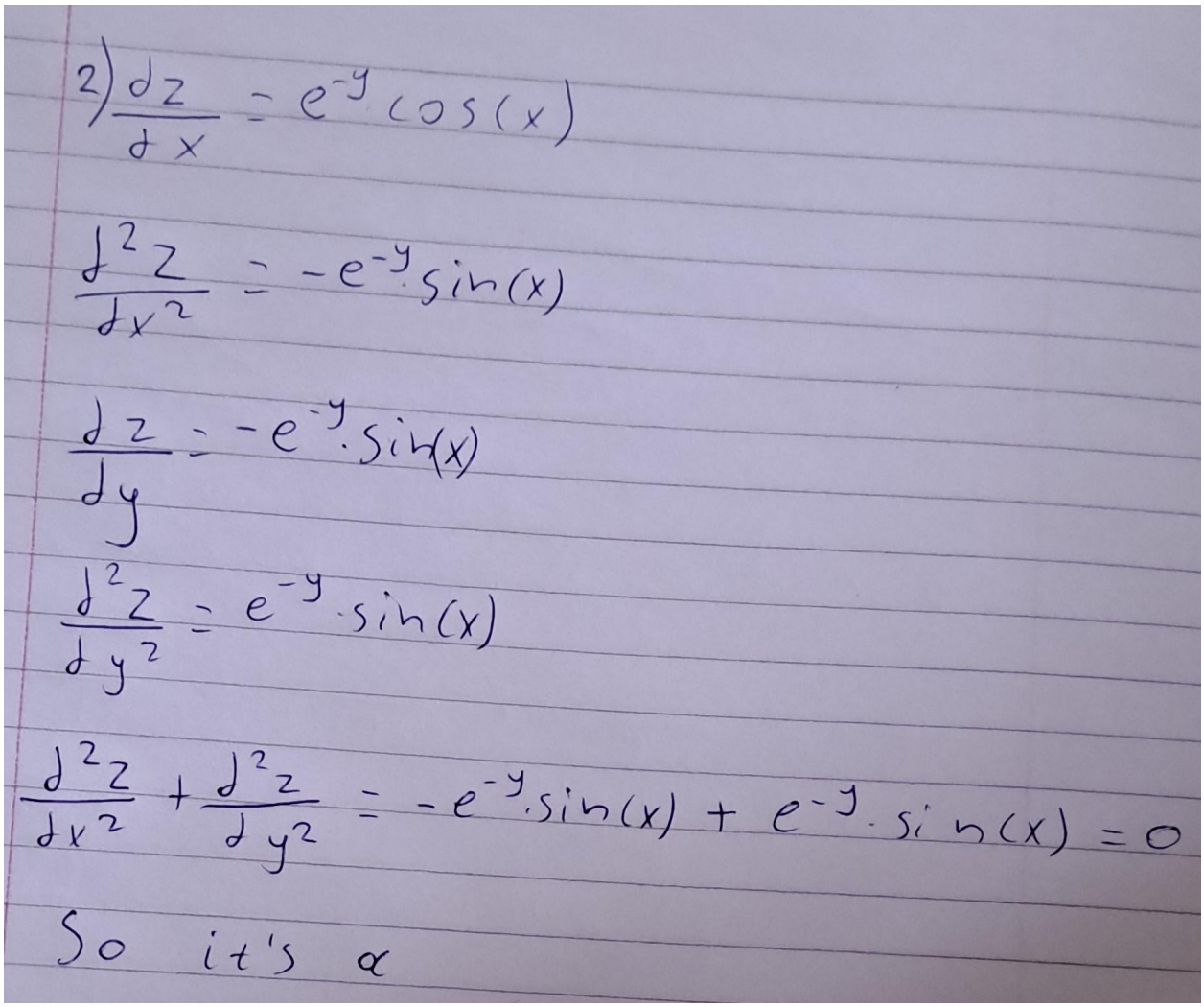
DAMA50 - 4th assignment

Quiz 1

Known math type, so it's (c)

Quiz 2



It's (a)

DAMA50 - 4th assignment

Quiz 3

```
%display latex
f(x, y) = (x^2)*y
x(u, v) = u + v
y(u, v) = u - v

f_u = diff(f(x(u,v), y(u,v)), u)
f_v = diff(f(x(u,v), y(u,v)), v)
gradient_f = (f_u, f_v)

gradient_f
```

$$\left((u+v)^2 + 2\,(u+v)(u-v), -(u+v)^2 + 2\,(u+v)(u-v)\right)$$

It's (e)

Quiz 4

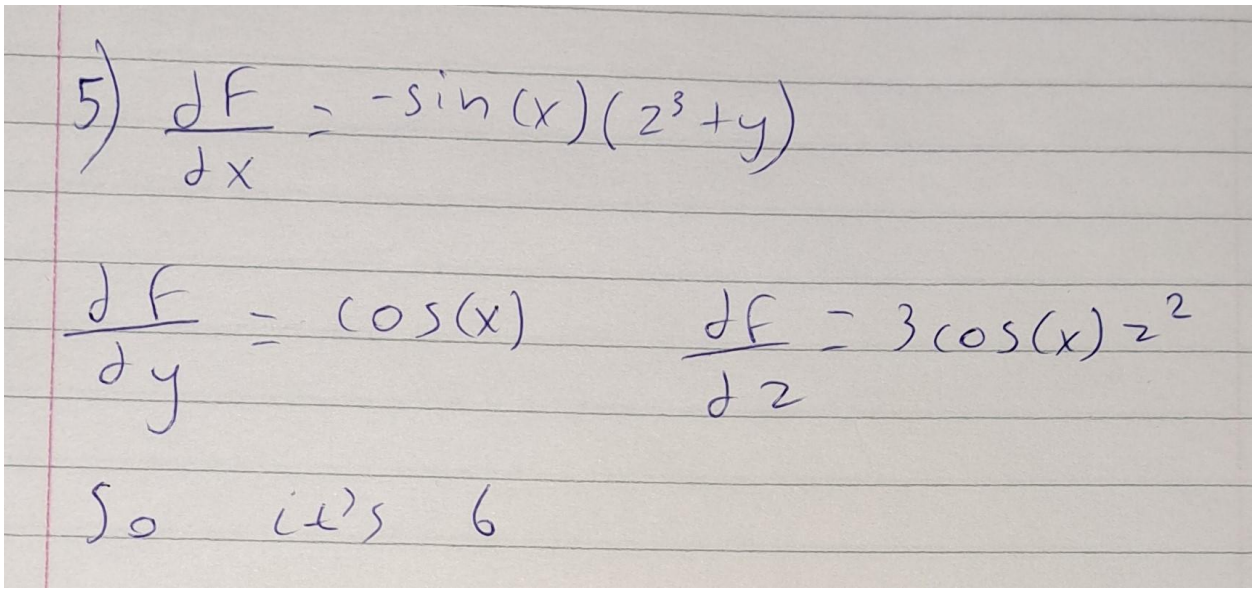
```
%display latex
var('x')
f(x) = (e^x - e^(-x))/(e^x + e^(-x))
diff(f(x), x)
```

$$-\frac{\left(e^{(-x)} - e^{x}\right)^2}{\left(e^{(-x)} + e^{x}\right)^2} + 1$$

It's (c)

DAMA50 - 4th assignment

Quiz 5



It's (b)

Quiz 6

0.12

```
%display latex
var('x y')
f1 = 4*(x^2)*cos(y)
f2 = log(x)*(y^2)
J = matrix([[diff(f1,x), diff(f1,y)], [diff(f2,x), diff(f2,y)]])
J.simplify_full()
```

$$\begin{pmatrix} 8\,x\cos\left(y\right) & -4\,x^2\sin\left(y\right) \\ \frac{y^2}{x} & 2\,y\log\left(x\right) \end{pmatrix}$$

It's (c)

DAMA50 - 4th assignment

Problem 7

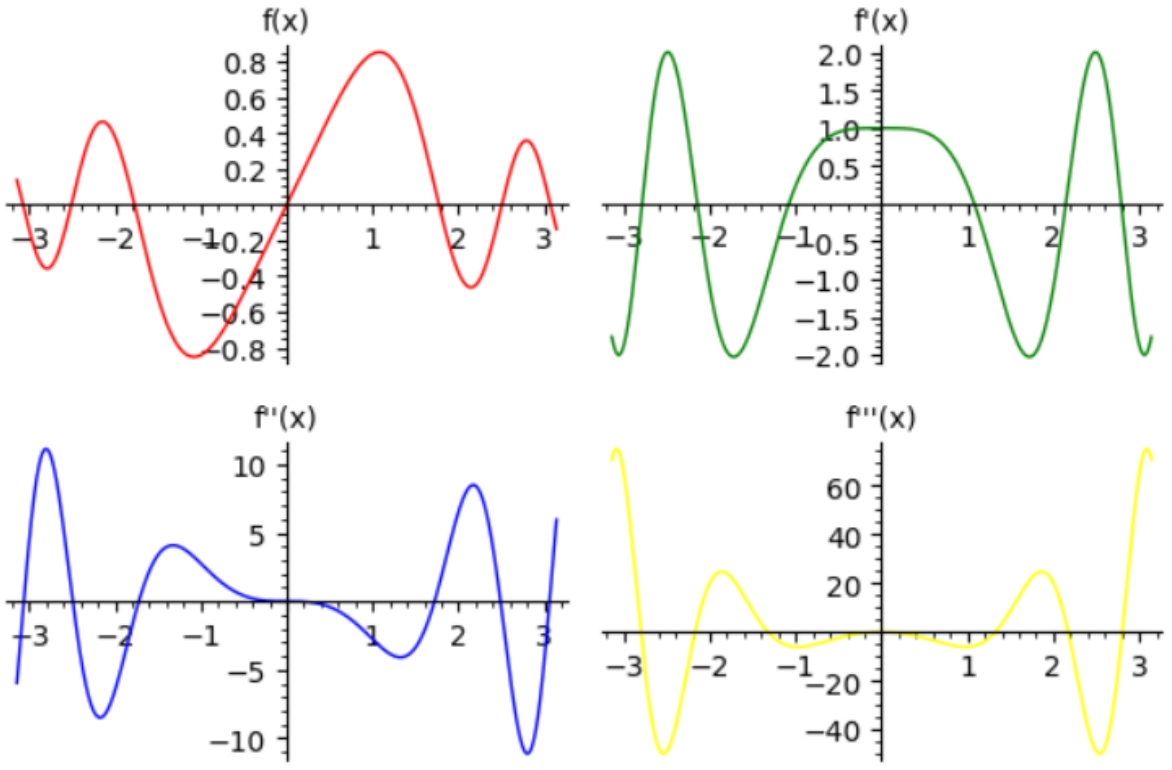
Solved in CoCalc

```
#7.a
```

```
def dplot(f, x0):
    x = var('x')
    p1 = plot(f, (x, -x0, x0), color='red', title="f(x)")
    p2 = plot(diff(f, x), (x, -x0, x0), color='green', title="f'(x)")
    p3 = plot(diff(f, x, 2), (x, -x0, x0), color='blue', title="f''(x)")
    p4 = plot(diff(f, x, 3), (x, -x0, x0), color='yellow', title="f'''(x)")
    return graphics_array([[p1, p2], [p3, p4]])
```

```
#7.b
```

```
f(x) = sin(x^2)/x
dplot(f, pi)
```



DAMA50 - 4th assignment

Problem 8

a,b,c,d solved in SageMath Notebook
e (plot) shown in IntelliJ
Note: This exercise was mostly thought of and solved in Python rather than SageMath since I run in some problems when trying to use some libraries and functionalities (e.g. the .unit_step() method you suggested in class)

8.a Installing autograd and importing all the necessary libraries for the exercise.

```
# 8.a
```

```
try :
    import autograd
    print ("Autograd is already installed!")
except :
    print ("Installing autograd.")
    !pip install autograd
import autograd.numpy as np
from autograd import grad
import numpy as np
import math
import matplotlib.pyplot as plt
```

```
Autograd is already installed!
```

8.b Defining the function that computes the numerical value of the gradient of f(x, y).

Note: As you'll see in 8.d's output, the print that I put in here shows the float we want to see but for some reason the return statement shows an error. I spent hours trying to find the reason of the error but I had no success in solving it. Also, the line from class auto_grad=grad(f) wasn't working so I made it into a function which worked.

DAMA50 - 4th assignment

```
# 8.b
```

```
def f(x, y):
    if x**2 + y**2 < 1:
        a = np.sin(x*y)
    else:
        a = np.sin(x*y) / np.sqrt(x**2 + y**2)
    print(a)
    return a

def auto_grad(x, y):
    return grad(f(x,y))
```

8.c Defining the function analytic_grad that computes analytically the gradient of f(x, y).

```
# 8.c
```

```
def analytic_grad(x, y):
    if x**2 + y**2 < 1:
        df_dx = y * math.cos(x*y)
        df_dy = x * math.cos(x*y)
    else:
        df_dx = (y * math.cos(x*y)) / math.sqrt(x**2 + y**2) - (x * y * math.sin(x*y)) / (x**2 + y**2)**(3/2)
        df_dy = (x * math.cos(x*y)) / math.sqrt(x**2 + y**2) - (x * y * math.sin(x*y)) / (x**2 + y**2)**(3/2)
    return np.array([df_dx, df_dy])
```

8.d Comparison the results of the functions auto_grad and analytic_grad.

Note: Here we see the error I mentioned in 8.b. Also, I used the np.random.seed(0) line so that the 10 points used were the same in many different tries to make sure I got the same results in each try thus the code worked correctly.

DAMA50 - 4th assignment

```
# 8.d
```

```
np.random.seed(0)
points = np.random.uniform(-np.pi, np.pi, size=(10, 2))
for x, y in points:
    auto_grad_final = auto_grad(x, y)
    analytic_grad_final = analytic_grad(x, y)
    print(f"x = {x}, y = {y}, auto_grad = {auto_grad_final}, analytic_grad = {analytic_grad_final}")
```

```
0.29060617180855036
x = 0.3067042906681201, y = 1.3520746650524709, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0
x6ffe856c8e60>, analytic_grad = [0.82987143 0.13977457]
0.18108362170117434
x = 0.6456813346495229, y = 0.2820093559455552, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0
x6ffe85651170>, analytic_grad = [0.27734709 0.63500674]
-0.41145107360095917
x = -0.47969104306747123, y = 0.9166797476244106, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at
0x6ffe85651560>, analytic_grad = [ 0.63270822 -0.58856443]
-0.32986401746551647
x = -0.3921511171743539, y = 2.46158236226362, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0x
6ffe856510e0>, analytic_grad = [ 0.51082287 -0.14079242]
-0.2815590653779432
x = 2.9132790442663943, y = -0.7323585363410419, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at
0x6ffe856515f0>, analytic_grad = [ 0.06350198 -0.58400367]
0.17735191094809816
x = 1.8329624730174032, y = 0.1815521352435825, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0
x6ffe85651830>, analytic_grad = [0.07576289 0.9231403 ]
0.3360322557895209
x = 0.42753658649887916, y = 2.6741025445054722, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at
0x6ffe85651440>, analytic_grad = [0.35702941 0.01307159]
0.17398448386451515
x = -2.695259936440876, y = -2.594143117880226, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0
x6ffe856513b0>, analytic_grad = [-0.6134092  -0.63393082]
-0.00462700245146494
x = -3.014556715858021, y = 2.0899121264231475, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0
x6ffe85661170>, analytic_grad = [ 0.56749782 -0.8238687 ]
-0.2738713463130749
x = 1.747710410660912, y = 2.324854893342369, auto_grad = <function unary_to_nary.<locals>.nary_operator.<locals>.nary_f at 0x6
ffe85661560>, analytic_grad = [-0.35169955 -0.23173455]
```

8.d Plotting

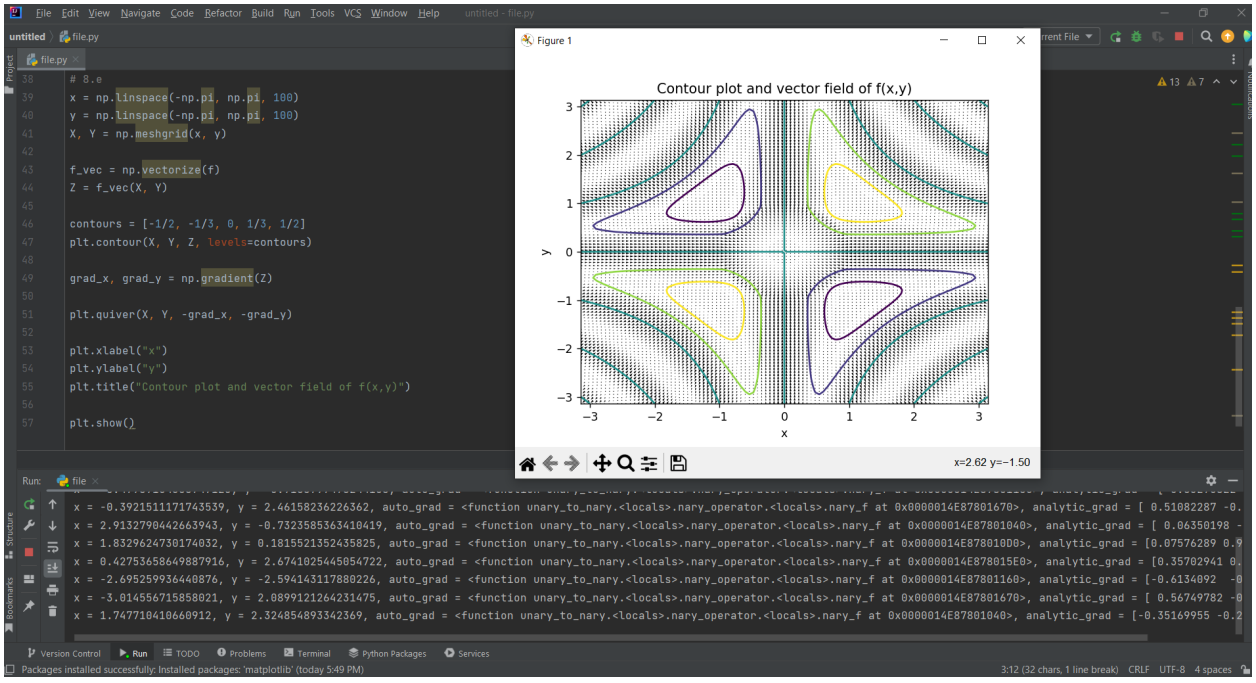
Note: Even though the code run in SageMath and produced the numbers, when the plot was called the kernel died. When I copied the code in IntelliJ the plot run as shown below.

DAMA50 - 4th assignment

```
# 8.e
```

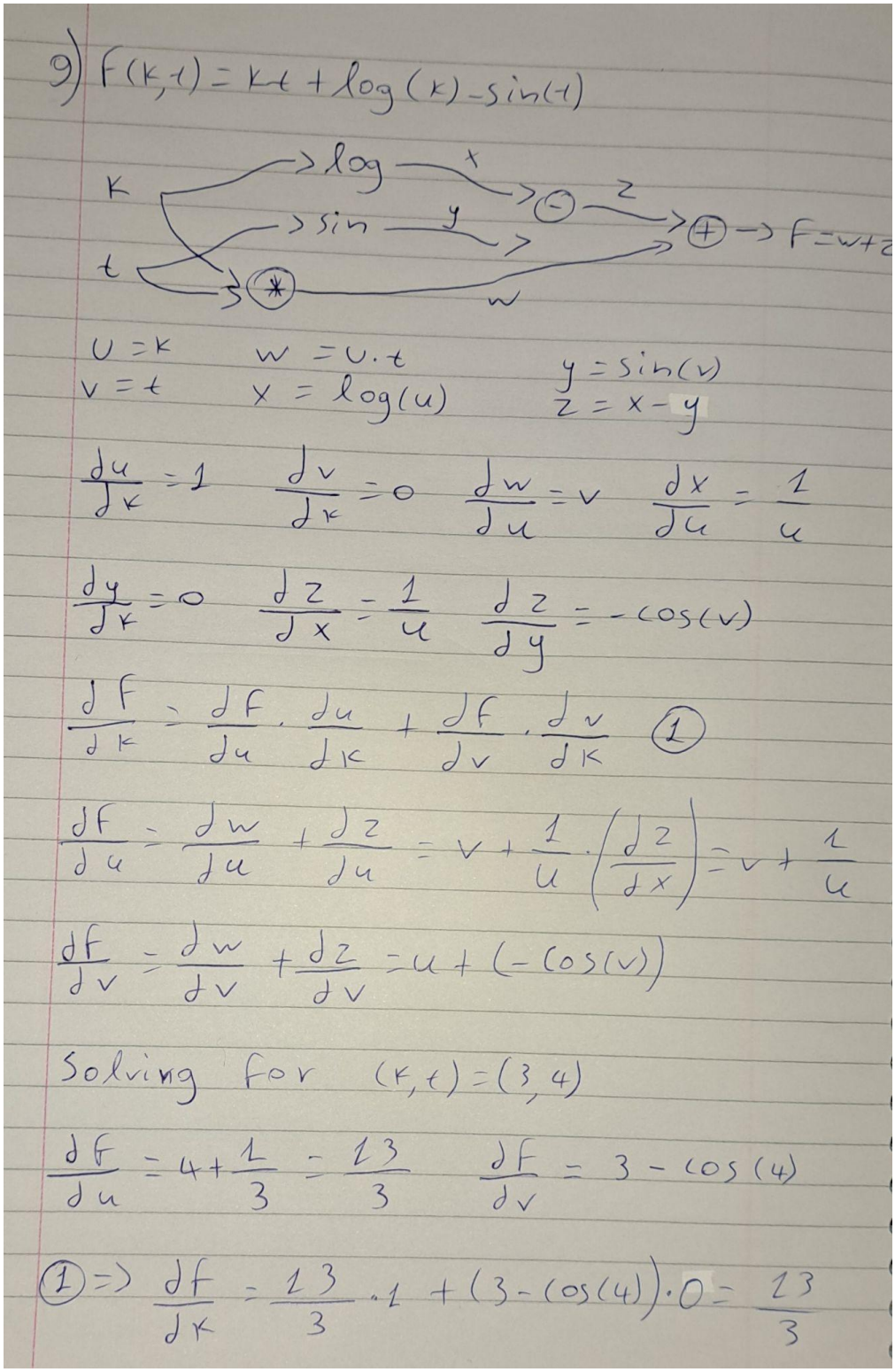
```
x = np.linspace(-np.pi, np.pi, 100)
y = np.linspace(-np.pi, np.pi, 100)
X, Y = np.meshgrid(x, y)

f_vec = np.vectorize(f)
Z = f_vec(X, Y)

contours = [-1/2, -1/3, 0, 1/3, 1/2]
plt.contour(X, Y, Z, levels=contours)

grad_x, grad_y = np.gradient(Z)

plt.quiver(X, Y, -grad_x, -grad_y)

plt.xlabel("x")
plt.ylabel("y")
plt.title("Contour plot and vector field of f(x,y)")

plt.show()
```

```
-0.09685180163186052
-0.09685180163186052
-0.05524572442531308
-0.010573657977722929
0.03543359036253365
0.08095547184459258
0.1241529981977471
0.16324237912330744
0.19656808655874095
0.22267238720523452
0.2403584524342153
0.2487443402868912
0.24730544388518166
0.2359034055677997
0.21479999336866218
0.18465500887036068
0.1465079222117041
0.1017435874245177 6
0.052043053567112495
```



DAMA50 - 4th assignment

Problem 9



DAMA50 - 4th assignment

Problem 10

10) $\frac{df}{dx} = \frac{\cos(x+2y)}{1+\sin(x+2y)}$, $\frac{d^2f}{dx^2} = \frac{-\sin(x+2y)}{(1+\sin(x+2y))^2}$

$\frac{df}{dy} = \frac{2\cdot\cos(x+2y)}{1+\sin(x+2y)}$, $\frac{d^2f}{dy^2} = \frac{-4\sin(x+2y)}{(1+\sin(x+2y))^2}$

$\frac{d^2f}{dx\,dy} = \frac{2\cos(x+2y)}{1+\sin(x+2y)}$

Solving for $(x_0, y_0) = (0, 0)$

$\frac{df}{dx_0} = 1$ $\frac{df}{dy_0} = 2$ $\frac{d^2f}{dx_0^2} = 0$ $\frac{d^2f}{dy_0^2} = 0$ $\frac{d^2f}{dx_0 dy_0} = 2$

$F(x,y) = F(x_0,y_0) + \frac{df}{dx_0}x + \frac{df}{dy_0}y + \frac{1}{2}\cdot\frac{d^2f}{dx_0^2}x^2 + \frac{1}{2}\frac{d^2f}{dy_0^2}y^2 +$

$+ \frac{d^2f}{dx_0 dy_0}\cdot x\cdot y + \dots$

$= \ln(1+\sin(0)) + x + 2y + \frac{1}{2}\cdot 0\cdot x^2 + \frac{1}{2}\cdot 0\cdot y^2 + 2\cdot xy$

$= x + 2y + 2xy + \dots$

$F(x,y) \approx x + 2y + 2xy$