from sympy import symbols, diff, solve, Matrix

x, y, l = symbols('x y lambda')

f = x\*\*2 + y\*\*2

g = x + y - 1

# Define the Lagrangian

L = f - l \* g

# Compute partial derivatives

partials = [diff(L, var) for var in (x, y, l)]

# Solve the system of equations

solution = solve(partials, (x, y, l), dict=True)[0]

# Extract the optimal values

optimal\_x = solution[x]

optimal\_y = solution[y]

# Compute the Hessian matrix

# Compute the Hessian matrix using a list of lists

hessian\_list = []

# Iterate over var2

for var2 in (x, y, l):

# Initialize a row for var2

row = []

# Iterate over var1

for var1 in (x, y, l):

# Calculate the second-order partial derivative and append to the row

row.append(diff(L.diff(var1), var2))

# Append the row to the Hessian list

hessian\_list.append(row)

# Create an instance of the Matrix class from the list of lists

hessian\_matrix = Matrix(hessian\_list)

# Display the Hessian matrix

print(hessian\_matrix)

hessian\_determinant = hessian\_matrix.det()

if hessian\_determinant > 0:

print("Stationary point is a local minimum.")

elif hessian\_determinant < 0:

print("Stationary point is a local maximum.")

else:6

print("Second-order test inconclusive (saddle point or test fails).")

# Display the result

print("Optimal solution:")

print(f"x: {optimal\_x}")

print(f"y: {optimal\_y}")