d) Gurobi-Python Problem 1

Global Maximum

In [ ]:
```python
from gurobipy import Model, GRB

# Create a new model
model = Model("Problem_1_max")

# Define continuous variables
x1 = model.addVar(lb=-GRB.INFINITY, name="x1")
x2 = model.addVar(lb=-GRB.INFINITY, name="x2")
x3 = model.addVar(lb=-GRB.INFINITY, name="x3")

# Set the objective function: f(x) = 3*x1 + 5*x2 - 3*x3^2
model.setObjective(3*x1 + 5*x2 - 3*x3*x3, GRB.MAXIMIZE)

# Add equality constraints using addQConstr since they are quadratic:
# Constraint 1: 2*x1^2 - 37*x2^2 + 9*x3 == 18
model.addQConstr(2*x1*x1 - 37*x2 + 9*x3 == 18, name="c1")

# Constraint 2: 5*x1 + x2 + 5*x3^2 == 24
model.addQConstr(5*x1 + x2 + 5*x3*x3 == 24, name="c2")

# Optimize the model
model.optimize()

# Print the results
if model.status == GRB.OPTIMAL:
    print("Optimal solution:")
    print(f"  x1 = {x1.x}")
    print(f"  x2 = {x2.x}")
    print(f"  x3 = {x3.x}")
    print(f"Optimal objective value = {model.objVal}")
else:
    print("No optimal solution found.")
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (win64 - Windows 10.0 (19045.2))

CPU model: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads

Optimize a model with 0 rows, 3 columns and 0 nonzeros
Model fingerprint: 0x5cc95152
Model has 1 quadratic objective term
Model has 2 quadratic constraints
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  QMatrix range    [2e+00, 5e+00]
  QLMatrix range   [1e+00, 4e+01]
  Objective range  [3e+00, 5e+00]

CPU model: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads

Optimize a model with 0 rows, 3 columns and 0 nonzeros
Model fingerprint: 0x5cc95152
Model has 1 quadratic objective term
Model has 2 quadratic constraints
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  QMatrix range    [2e+00, 5e+00]
  QLMatrix range   [1e+00, 4e+01]
  Objective range  [3e+00, 5e+00]
  QObjective range [6e+00, 6e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [0e+00, 0e+00]
  QRHS range       [2e+01, 2e+01]

Continuous model is non-convex -- solving as a MIP

Presolve time: 0.00s
Presolved: 6 rows, 4 columns, 18 nonzeros
Presolved model has 2 bilinear constraint(s)
Warning: Model contains variables with very large bounds participating
         in product terms.
         Presolve was not able to compute smaller bounds for these variables.
         Consider bounding these variables or reformulating the model.

Variable types: 4 continuous, 0 integer (0 binary)
Found heuristic solution: objective 2257.5755700

Root relaxation: unbounded, 2 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  postponed    0      2257.57557          -      -     -    0s
     0     0  postponed    0      2257.57557          -      -     -    0s
     0     2  postponed    0      2257.57557          -      -     -    0s
*   97     2              19    2257.5755740          -      -   1.1    0s

Explored 129 nodes (111 simplex iterations) in 0.05 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)

Solution count 2: 2257.58 2257.58

Optimal solution found (tolerance 1.00e-04)
Best objective 2.257575573672e+03, best bound 2.257575573961e+03, gap 0.0000%
Optimal solution:
  x1 = -97.16265952910427
  x2 = 509.8127734171107
  x3 = -0.010244122590746783
Optimal objective value = 2257.5755736720976
```

Global Minimum

In [ ]:

```
model = Model("Problem_1_min")

# Define continuous variables
x1 = model.addVar(lb=-GRB.INFINITY, name="x1")
x2 = model.addVar(lb=-GRB.INFINITY, name="x2")
x3 = model.addVar(lb=-GRB.INFINITY, name="x3")


# Set the objective function: f(x) = 3*x1 + 5*x2 - 3*x3^2
model.setObjective(3*x1 + 5*x2 - 3*x3*x3, GRB.MINIMIZE)

# Add equality constraints using addQConstr since they are quadratic:
# Constraint 1: 2*x1^2 - 37*x2^2 + 9*x3 == 18
model.addQConstr(2*x1*x1 - 37*x2 + 9*x3 == 18, name="c1")

# Constraint 2: 5*x1 + x2 + 5*x3^2 == 24
model.addQConstr(5*x1 + x2 + 5*x3*x3 == 24, name="c2")

# Optimize the model
model.optimize()

# Print the results
if model.status == GRB.OPTIMAL:
```

```
    print("Optimal solution:")
    print(f"  x1 = {x1.x}")
    print(f"  x2 = {x2.x}")
    print(f"  x3 = {x3.x}")
    print(f"Optimal objective value = {model.objVal}")
else:
    print("No optimal solution found.")
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (win64 - Windows 10.0 (19045.2))


CPU model: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads

Optimize a model with 0 rows, 3 columns and 0 nonzeros
Model fingerprint: 0x3e262e42
Model has 1 quadratic objective term
Model has 2 quadratic constraints
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  QMatrix range    [2e+00, 5e+00]
  QLMatrix range   [1e+00, 4e+01]
  Objective range  [3e+00, 5e+00]
  QObjective range [6e+00, 6e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [0e+00, 0e+00]
  QRHS range       [2e+01, 2e+01]

Continuous model is non-convex -- solving as a MIP

Presolve time: 0.00s
Presolved: 6 rows, 4 columns, 18 nonzeros
Presolved model has 2 bilinear constraint(s)
Warning: Model contains variables with very large bounds participating
         in product terms.
         Presolve was not able to compute smaller bounds for these variables.
         Consider bounding these variables or reformulating the model.

Variable types: 4 continuous, 0 integer (0 binary)
Found heuristic solution: objective -51.9570140

Root relaxation: unbounded, 2 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  postponed    0       -51.95701          -      -     -    0s
     0     0  postponed    0       -51.95701          -      -     -    0s
     0     2  postponed    0       -51.95701          -      -     -    0s

Explored 45 nodes (70 simplex iterations) in 0.06 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)

Solution count 1: -51.957

Optimal solution found (tolerance 1.00e-04)
Best objective -5.195701399667e+01, best bound -5.195719131412e+01, gap 0.0003%
Optimal solution:
  x1 = -10.14600494474171
  x2 = 4.1641099413894915
  x3 = -3.756751649559144
Optimal objective value = -51.95701399667372
```

Results Discussion

The Gurobi Solver returns the global optima for the problems. As we stated in the previous questions, points xB and xD were global optima, and these Gurobi solutions directly correspond to them.

- **Global Maximum:**

$$x_1 = -97.16, \quad x_2 = 509.81, \quad x_3 = -0.01024$$

- **Global Minimum:**

$$x_1 = -10.15, \quad x_2 = 4.164, \quad x_3 = -3.7568$$

This demonstrates the consistency of both approaches (symbolic analysis via MATLAB and numerical optimization via Gurobi) in identifying the correct optimal solutions. Nevertheless, in the Gurobi–Python approach, constraint qualifications are assumed to hold (or are enforced implicitly).

c) Gurobi-Python Problem 2

In [ ]:
```
# Create a new model
model = Model("Problem_2_max")

# Define continuous variables
x1 = model.addVar(lb=-GRB.INFINITY, name="x1")
x2 = model.addVar(lb=-GRB.INFINITY, name="x2")
x3 = model.addVar(lb=-GRB.INFINITY, name="x3")

# Set the objective function: f(x) = 3*x1 + 5*x2 - 3*x3^2
```

```python
model.setObjective(x1*x1 + x2*x2 - x3*x3, GRB.MAXIMIZE)

# Add equality constraints using addQConstr since they are quadratic:
# Constraint 1: 2*x1^2 - 37*x2^2 + 9*x3 == 18
model.addQConstr(8*x1*x1 + 24*x2 - 15*x3 <= 129, "c1")

# Constraint 2: 5*x1 + x2 + 5*x3^2 == 24
model.addQConstr(-x1*x1 - 2*x2*x2 - 4*x3*x3 <= -15, "c2")

# Optimize the model
model.optimize()

# Print the results
if model.status == GRB.OPTIMAL:
    print("Optimal solution:")
    print(f"  x1 = {x1.x}")
    print(f"  x2 = {x2.x}")
    print(f"  x3 = {x3.x}")
    print(f"Optimal objective value = {model.objVal}")
else:
    print("No optimal solution found.")
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (win64 - Windows 10.0 (19045.2))

CPU model: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads

Optimize a model with 0 rows, 3 columns and 0 nonzeros
Model fingerprint: 0x2382531e
Model has 3 quadratic objective terms
Model has 2 quadratic constraints
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  QMatrix range    [1e+00, 8e+00]
  QLMatrix range   [2e+01, 2e+01]

CPU model: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, instruction set [SSE2|AVX|AVX2]
Thread count: 6 physical cores, 12 logical processors, using up to 12 threads

Optimize a model with 0 rows, 3 columns and 0 nonzeros
Model fingerprint: 0x2382531e
Model has 3 quadratic objective terms
Model has 2 quadratic constraints
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  QMatrix range    [1e+00, 8e+00]
  QLMatrix range   [2e+01, 2e+01]
  Objective range  [0e+00, 0e+00]
  QObjective range [2e+00, 2e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [0e+00, 0e+00]
  QRHS range       [2e+01, 1e+02]

Continuous model is non-convex -- solving as a MIP

Presolve time: 0.00s
Presolved: 10 rows, 8 columns, 24 nonzeros
Presolved model has 3 bilinear constraint(s)
Warning: Model contains variables with very large bounds participating
         in product terms.
         Presolve was not able to compute smaller bounds for these variables.
         Consider bounding these variables or reformulating the model.

Variable types: 8 continuous, 0 integer (0 binary)
Found heuristic solution: objective 47.4102564

Root relaxation: unbounded, 7 iterations, 0.00 seconds (0.00 work units)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0  postponed    0        47.41026          -      -     -    0s
     0     0  postponed    0        47.41026          -      -     -    0s
     0     2  postponed    0        47.41026          -      -     -    0s
*    3     6               2    625000.00000          -      -   4.7    0s
*    8     8               3    750000.00000          -      -   3.1    0s
*    9     8               3    1125000.0000          -      -   2.8    0s
*   13    12               4    1250000.0000          -      -   2.8    0s
*   19    12               5    2250000.0000          -      -   1.9    0s
*   21    12               6    4250000.0000          -      -   1.7    0s
*   26    12               6    8125000.0000          -      -   1.4    0s
*   40    14               7    8250000.0000          -      -   0.9    0s
*   42    14               8    1.625000e+07          -      -   0.9    0s
*   44    14               9    3.225000e+07          -      -   0.8    0s
*   49    14               9    6.412500e+07          -      -   0.8    0s
*   67    26              10    6.425000e+07          -      -   0.6    0s
*   69    26              11    1.282500e+08          -      -   0.5    0s
*   74    26              11    2.561250e+08          -      -   0.5    0s
*  106    40              12    2.562500e+08          -      -   0.4    0s
*  108    40              13    5.122500e+08          -      -   0.4    0s
*  110    40              14    1.024250e+09          -      -   0.4    0s
*  112    40              15    2.048250e+09          -      -   0.4    0s
*  114    40              16    4.096250e+09          -      -   0.4    0s
*  149    40              19    4.098000e+09          -      -   0.4    0s
*  165    40              19    8.193000e+09          -      -   0.3    0s
*  193    38              24    8.224000e+09          -      -   0.3    0s
*  195    38              25    1.641600e+10          -      -   0.3    0s
*  230    38              22    3.277000e+10          -      -   0.3    0s
*  232    38              23    6.553800e+10          -      -   0.3    0s
*  234    38              24    1.310740e+11          -      -   0.3    0s
*  236    38              25    2.621460e+11          -      -   0.3    0s
*  278    44              29    2.621760e+11          -      -   0.2    0s
*  280    44              30    5.243200e+11          -      -   0.2    0s
*  282    44              31    1.048608e+12          -      -   0.2    0s
*  284    44              32    2.097184e+12          -      -   0.2    0s
*  359    44              33    2.097216e+12          -      -   0.3    0s
*  361    44              33    4.194336e+12          -      -   0.3    0s
*  363    44              34    8.388640e+12          -      -   0.3    0s
*  365    44              35    1.677725e+13          -      -   0.2    0s
*  367    44              36    3.355446e+13          -      -   0.2    0s
*  369    44              37    6.710890e+13          -      -   0.2    0s
H  371    44                    4.551109e+14          -      -   0.2    0s
*  547    94              44    5.113881e+14          -      -   0.2    0s
*  592    94              42    5.368710e+14          -      -   0.2    0s
*  749    66              45    5.368719e+14          -      -   0.3    0s
H 1017    46                    6.710886e+19          -      -   0.2    0s
```

```
Explored 2249 nodes (235 simplex iterations) in 0.12 seconds (0.01 work units)
Thread count was 12 (of 12 available processors)

Solution count 10: 6.71089e+19 5.36872e+14 5.36871e+14 ... 2.56125e+08

Optimal solution found (tolerance 1.00e-04)
Best objective 6.710886188749e+19, best bound 6.710886188749e+19, gap 0.0000%
Optimal solution:
  x1 = -32000.0
  x2 = -8191999871.0
  x3 = 0.0
Optimal objective value = 6.710886188748802e+19
```

Results Discussion

The enormous solution returned by Gurobi confirms what we observed in MATLAB. The problem is unbounded.

Our MATLAB analysis showed that there is no finite maximum since the objective can be increased without bound over the feasible set. There is no point at which all the 1st-order conditions (KKT) and 2nd-order conditions hold in such a way that a maximum is reached; rather, you can always find a feasible direction along which $f(x)$ increases. Consequently, Gurobi, when solving the model without imposing artificial bounds, returns an enormous solution as it essentially drives the objective value toward infinity, thereby demonstrating that no finite optimal solution exists.