Part 2: Solving via Gurobi in Python

In the following section, Problems 1 and 2 are now solved using Gurobi in Python.

Problem 1

In problem 1, we are serching for optima of the objective function

$$f(\mathbf{x}) = x_1^2 + x_2^2 - x_3^2,$$

s.t.

$$8x_1^2 + 24x_2 - 15x_3 \leq 129$$

$$x_1^2 + 2x_2^2 + 4x_3^2 \geq 15.$$

We begin with setting up the Variables and Constraints of the Problem in gurobipy.

In [10]:
```
from gurobipy import Model, GRB

model1 = Model('prob1')

# Define the variables
x1 = model1.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x1")
x2 = model1.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x2")
x3 = model1.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x3")

#add constraints
model1.addConstr(2 * x1 ** 2 - 37 * x2 + 9 * x3 == 18, "c1")
model1.addConstr(5 * x1 + x2 + 5 * x3 ** 2 == 24, "c2")
```

Out[10]:
```
<gurobi.QConstr Not Yet Added>
```

Then we first define the objective function with the goal to maximize, run the solver and print the results. Gurobi, looking for the global optimum finds the same point as found in the matlab analysis as candidate point D.

In [11]:
```
# define objective for maximization
model1.setObjective(3*x1 + 5*x2 - 3*x3**2, GRB.MAXIMIZE)

# solve
model1.setParam('OutputFlag', 0)
model1.optimize()

# print results
for v in model1.getVars():
```

```
    print(f"{v.varName}: {v.x}")
print(f"Optimal objective value: {model1.objVal}")
```

```
x1: -97.16265952910427
x2: 509.8127734171107
x3: -0.010244122590746783
Optimal objective value: 2257.5755736720976
```

Likewise, the global minimum is found corresponding to candidate Point B from the Matlab analysis.

In [12]:
```
# define objective for maximization
model1.setObjective(3*x1 + 5*x2 - 3*x3**2, GRB.MINIMIZE)

# solve
model1.optimize()

# print results
for v in model1.getVars():
    print(f"{v.varName}: {v.x}")
print(f"Optimal objective value: {model1.objVal}")
```

```
x1: -10.14600494474171
x2: 4.164109941389492
x3: -3.756751649559144
Optimal objective value: -51.95701399667372
```

Overall, as Gurobi attempts to find global optima, we can only compare these meaning that analysis beyond taken the result is dificult. This might for example also be the case when multiple global optima exist, of which we might not be aware when using Gurobi.

Problem 2

In problem 2, our aim is to maximize the following function.

$$f(\mathbf{x}) = x_1^2 + x_2^2 - x_3^2,$$

s.t.

$$8x_1^2 + 24x_2 - 15x_3 \leq 129$$

$$x_1^2 + 2x_2^2 + 4x_3^2 \geq 15.$$

where as in the previous section, we reformulate the second constraint as

$$-x_1^2 - 2x_2^2 - 4x_3^2 \leq -15$$

to make sure that the optimization is in standard form for maximization. We then define the Variables, Objective function and Constraints in Gurobi and run the solver. When looking at the results the solver outputs, we find the variables and function value to be extremely large and the solver to be giving a

warning to the Problem being unbounded.

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

model2 = Model('prob2')

# Define the variables
x1 = model2.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x1")
x2 = model2.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x2")
x3 = model2.addVar(lb = -GRB.INFINITY, ub= GRB.INFINITY, name="x3")

# define objective
model2.setObjective(x1**2 + x2**2 - x3**2, GRB.MAXIMIZE)

#add constraints
model2.addConstr(8 * x1**2 + 24 * x2 - 15 * x3 <= 129, "c2")
model2.addConstr(- x1**2 - 2*x2**2 - 4*x3**2 <= -15, "c1")

# solve
model2.optimize()

# print results
for v in model2.getVars():
    print(f"{v.varName}: {v.x}")
print(f"Optimal objective value: {model2.objVal}")
```

```
Gurobi Optimizer version 12.0.1 build v12.0.1rc0 (mac64[x86] - Darwi
n 24.3.0 24D81)

CPU model: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
Thread count: 6 physical cores, 12 logical processors, using up to 1
2 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:

CPU model: Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz
Thread count: 6 physical cores, 12 logical processors, using up to 1
2 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 participati
ng
         in product terms.
         Presolve was not able to compute smaller bounds for these v
ariables.
         Consider bounding these variables or reformulating the mode
l.

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 un
its)

    Nodes    |    Current Node    |     Objective Bounds      |
Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/N
ode 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 0.16 seconds (0.01 w
ork 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, ga
p 0.0000%
x1: -32000.0
x2: -8191999871.0
x3: 0.0
Optimal objective value: 6.710886188748802e+19
```

Looking at the problem again, we notice that the Porblem is indeed unboundes, with $x_1 = 0$, $x_2 = -\infty$, $x_3 = 0$ yielding $f(x) = \infty$.

As Gurboi searches only for a single global optima, algorithms like gradient ascent will move to arbitrarily high values for unbounded problems. In the output of the Gurobi solver we even see how the algorithm initially finds the same local maximum as we did with a function value of 47.4 before the runaway occours. The approach taken in MATLAB with the conditions used to identify optima do not yield the same arbitrarily high values as they are limited to fininding saddle points or maxima/minima with gradients equal to zero. Significantly different results in both approaches are thus not unexpected for an unbounded problem.

The Optimization Problem should thus be reconsidered as in its current unbounded form a objective function $f(x) = \infty$ is possible. Overall this is a good example on why black-box solvers need to be used with caution and a good understanding of their limitations, while in the meantime, the approach taken in the matlab section did not give any warnings about the unbounded nature of the problem and might have remainded unnoticed if we had not compared the results to the Gurobi Results. Thus, overall this shows how both approaches have strengths and weaknesses of which one should be aware when applying them to problems.