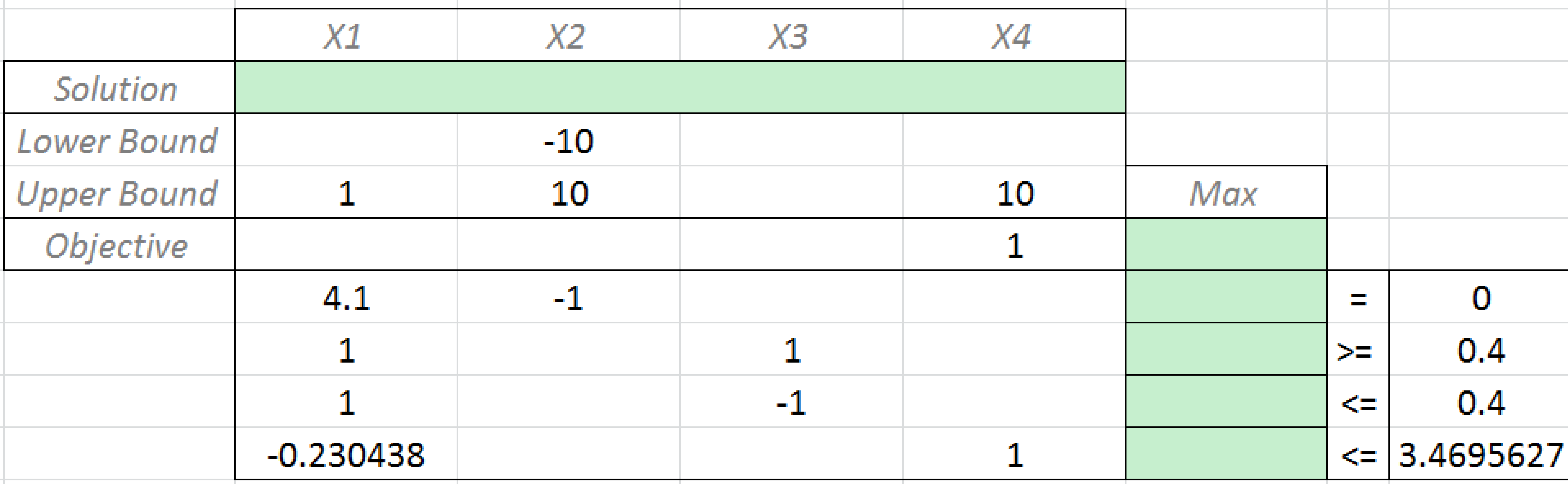
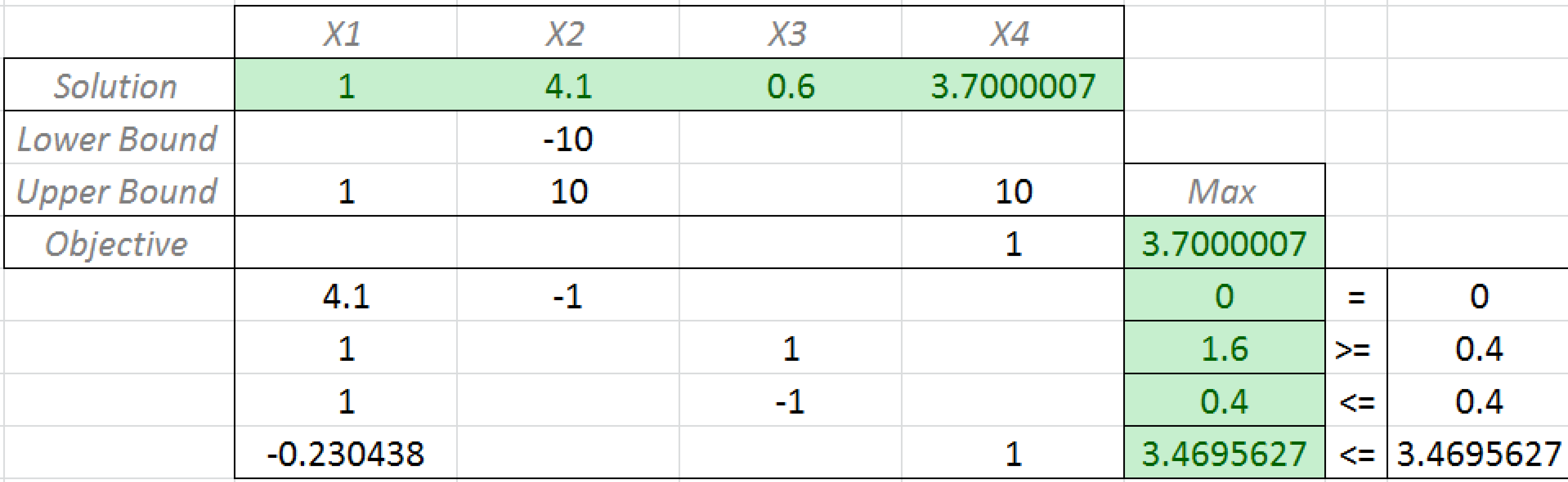
Fun with Gurobi v6.5.2

A Simple LP in Excel Solver



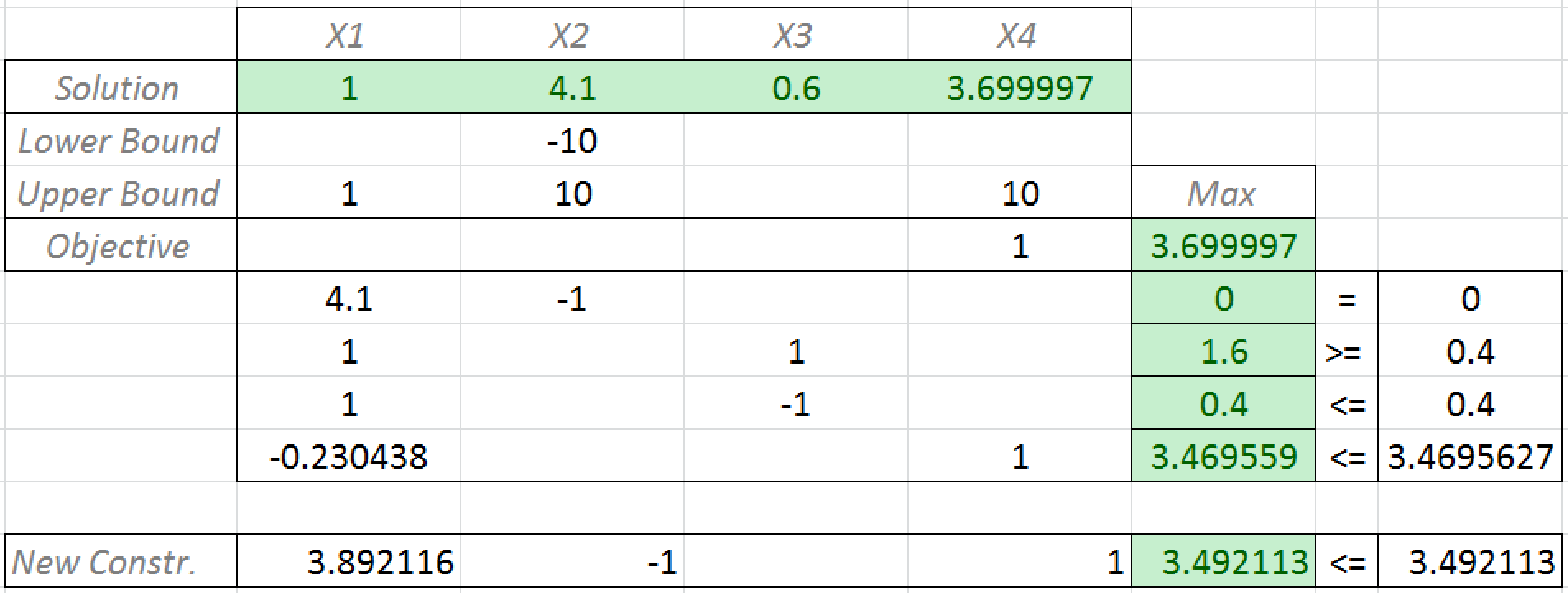
	X1	X2	X3	X4			
Solution							
Lower Bound		-10					
Upper Bound	1	10		10	Max		
Objective				1			
	4.1	-1				=	0
	1		1			>=	0.4
	1		-1			<=	0.4
	-0.230438			1		<=	3.4695627

Excel: Solve (1)



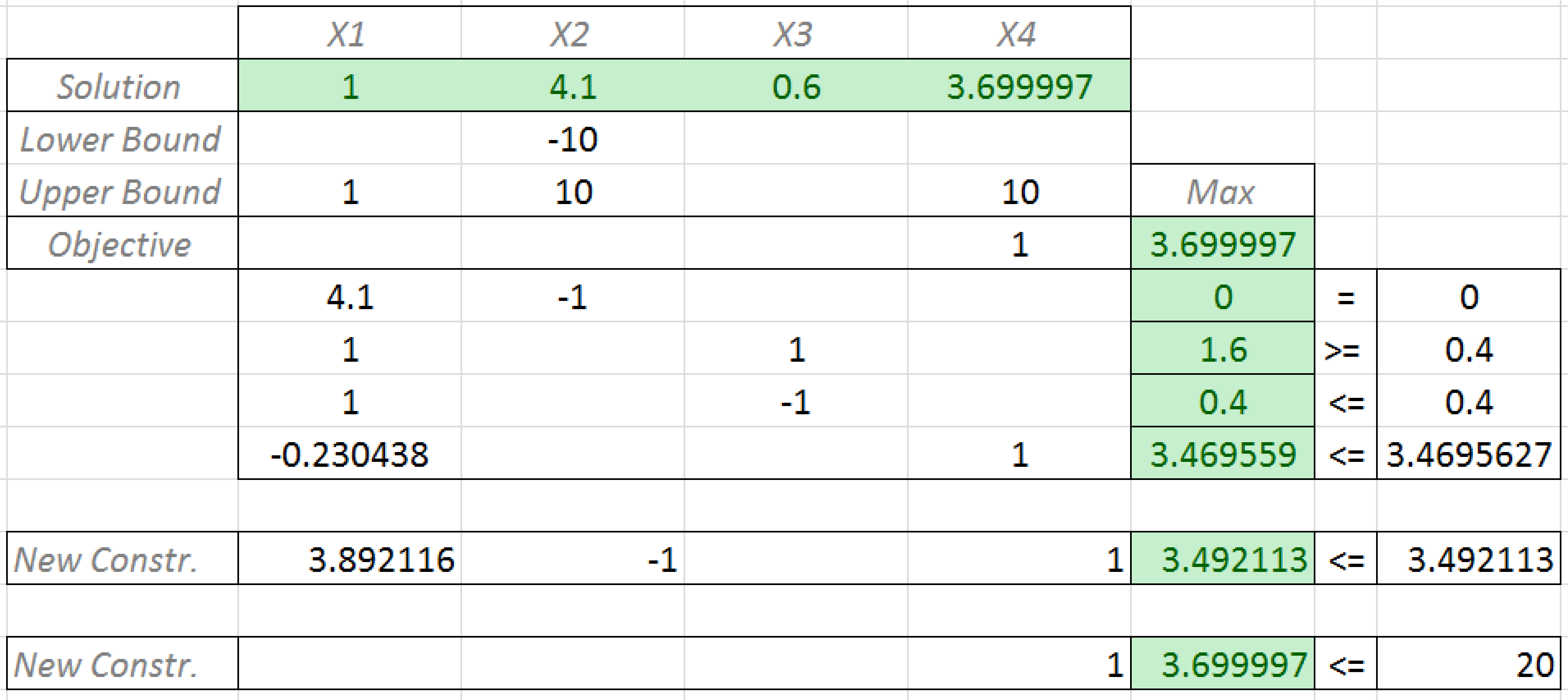
	X1	X2	X3	X4			
Solution	1	4.1	0.6	3.7000007			
Lower Bound		-10					
Upper Bound	1	10		10	Max		
Objective				1	3.7000007		
	4.1	-1			0	=	0
	1		1		1.6	>=	0.4
	1		-1		0.4	<=	0.4
	-0.230438			1	3.4695627	<=	3.4695627

Excel: Solve (2)



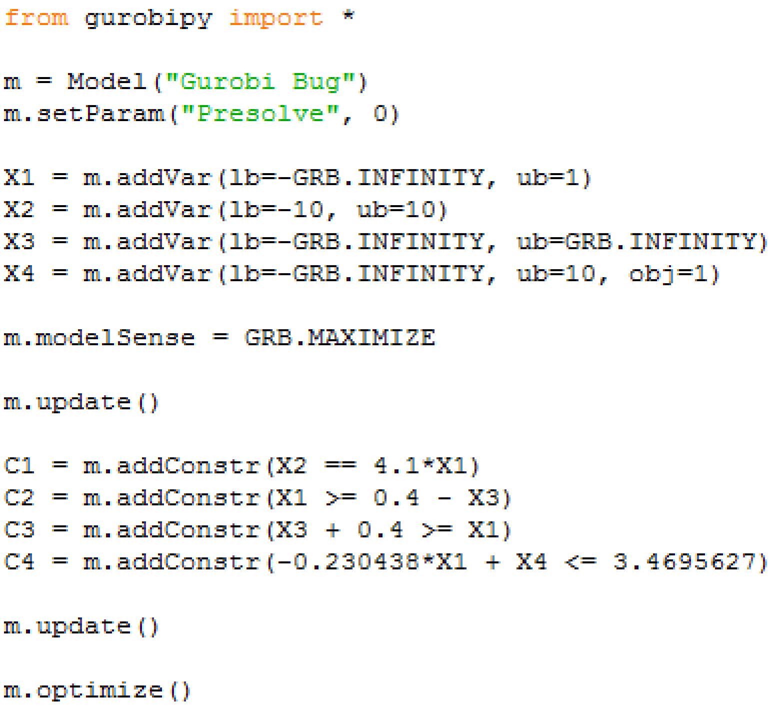
	X1	X2	X3	X4			
Solution	1	4.1	0.6	3.699997			
Lower Bound		-10					
Upper Bound	1	10		10	Max		
Objective				1	3.699997		
	4.1	-1			0	=	0
	1		1		1.6	>=	0.4
	1		-1		0.4	<=	0.4
	-0.230438			1	3.469559	<=	3.4695627
New Constr.	3.892116	-1		1	3.492113	<=	3.492113

Excel: Solve (3)

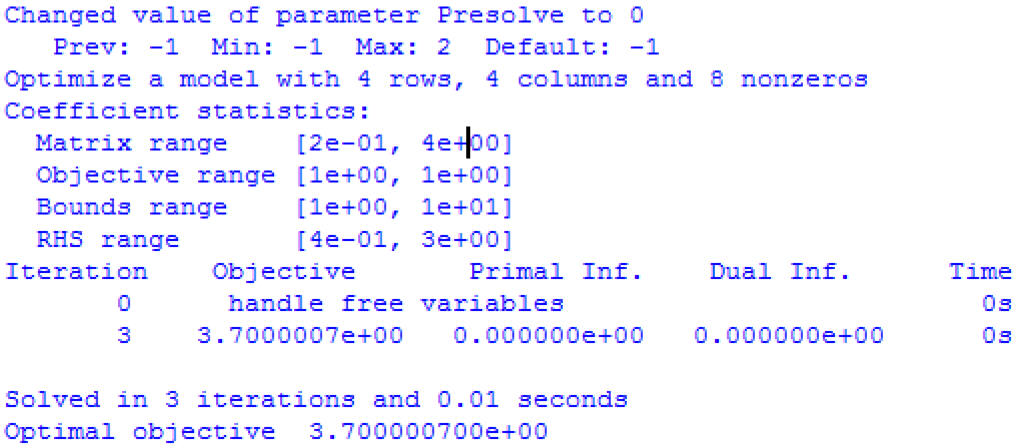


	X1	X2	X3	X4			
Solution	1	4.1	0.6	3.699997			
Lower Bound		-10					
Upper Bound	1	10		10	Max		
Objective				1	3.699997		
	4.1	-1			0	=	0
	1		1		1.6	>=	0.4
	1		-1		0.4	<=	0.4
	-0.230438			1	3.469559	<=	3.4695627
New Constr.	3.892116	-1		1	3.492113	<=	3.492113
New Constr.				1	3.699997	<=	20

GurobiPy: Solve (1)

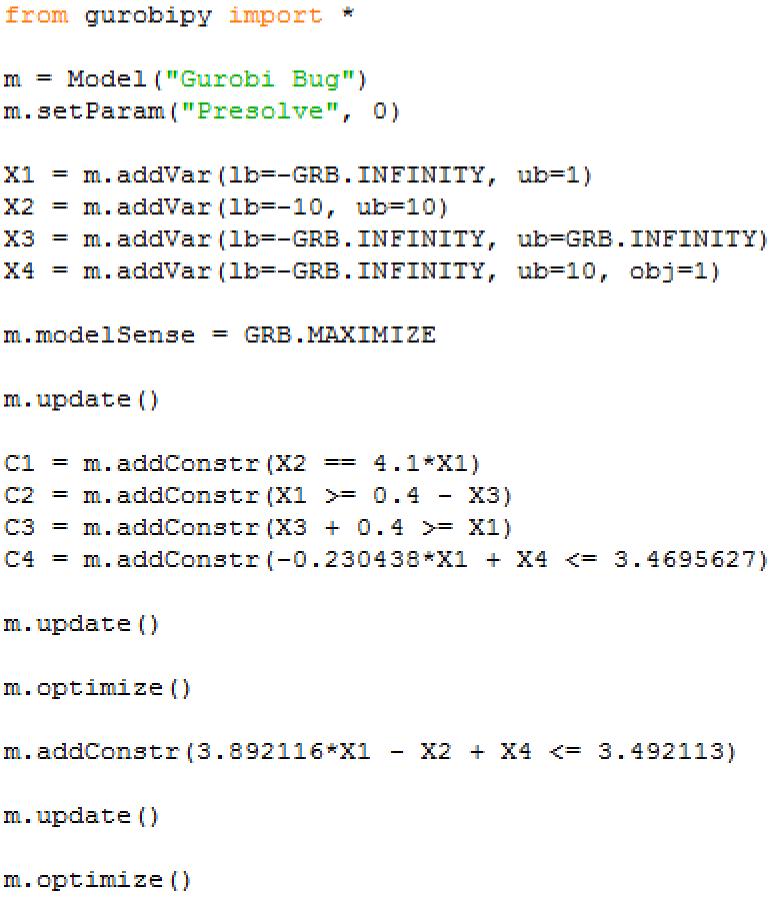


```
from gurobipy import *

m = Model("Gurobi Bug")
m.setParam("Presolve", 0)

X1 = m.addVar(lb=-GRB.INFINITY, ub=1)
X2 = m.addVar(lb=-10, ub=10)
X3 = m.addVar(lb=-GRB.INFINITY, ub=GRB.INFINITY)
X4 = m.addVar(lb=-GRB.INFINITY, ub=10, obj=1)

m.modelSense = GRB.MAXIMIZE

m.update()

C1 = m.addConstr(X2 == 4.1*X1)
C2 = m.addConstr(X1 >= 0.4 - X3)
C3 = m.addConstr(X3 + 0.4 >= X1)
C4 = m.addConstr(-0.230438*X1 + X4 <= 3.4695627)

m.update()

m.optimize()
```

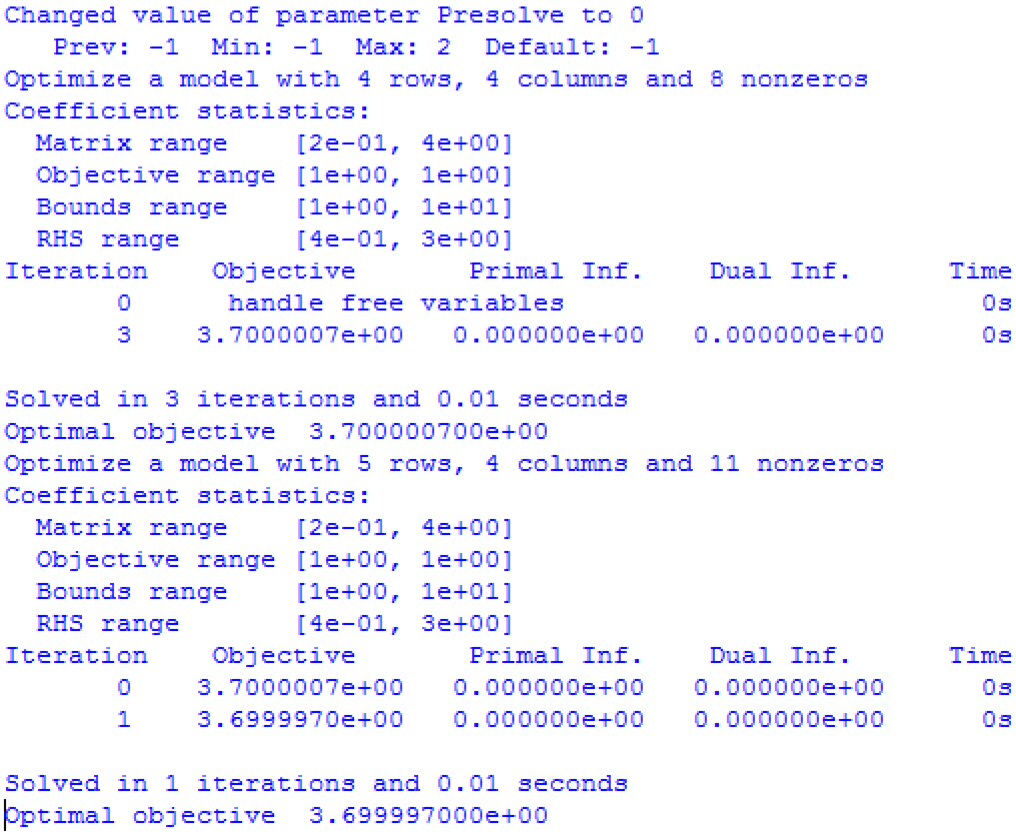


```
Changed value of parameter Presolve to 0
   Prev: -1  Min: -1  Max: 2  Default: -1
Optimize a model with 4 rows, 4 columns and 8 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0      handle free variables                          0s
       3    3.7000007e+00   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.01 seconds
Optimal objective  3.700000700e+00
```

GurobiPy: Solve (2)

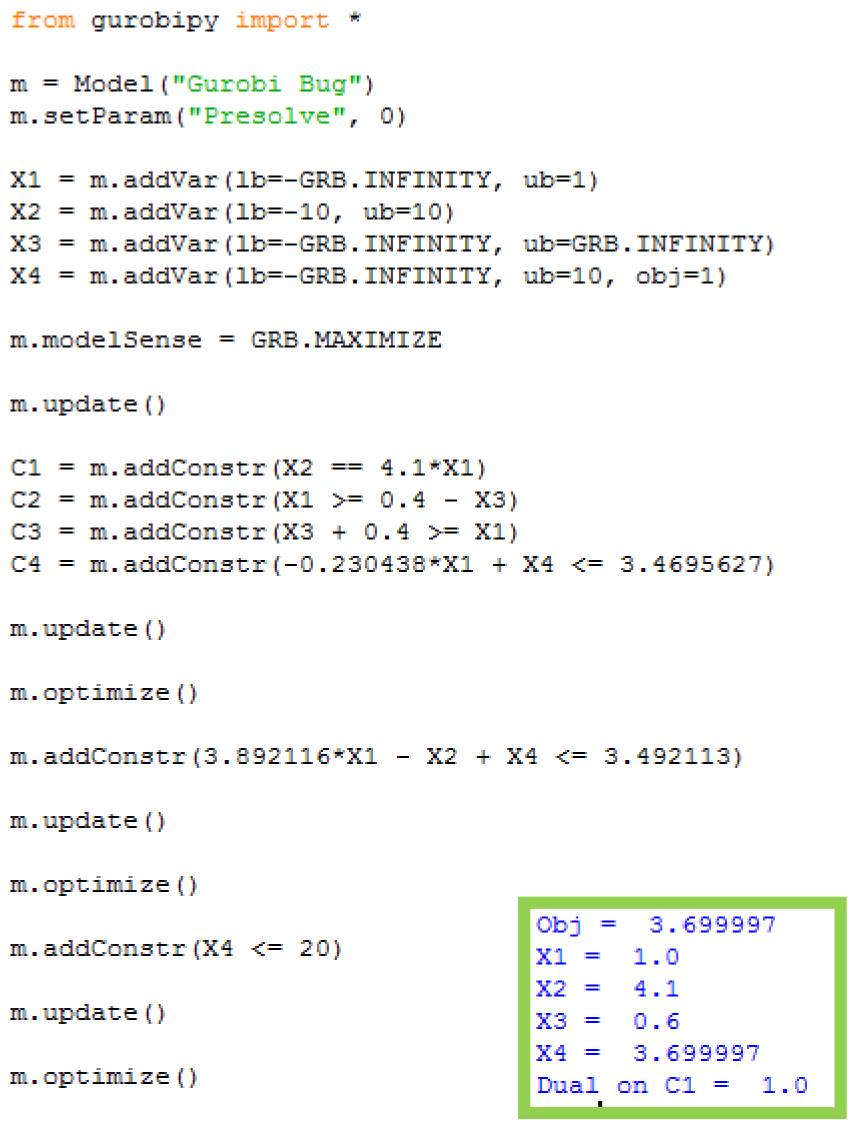


```
from gurobipy import *

m = Model("Gurobi Bug")
m.setParam("Presolve", 0)

X1 = m.addVar(lb=-GRB.INFINITY, ub=1)
X2 = m.addVar(lb=-10, ub=10)
X3 = m.addVar(lb=-GRB.INFINITY, ub=GRB.INFINITY)
X4 = m.addVar(lb=-GRB.INFINITY, ub=10, obj=1)

m.modelSense = GRB.MAXIMIZE

m.update()

C1 = m.addConstr(X2 == 4.1*X1)
C2 = m.addConstr(X1 >= 0.4 - X3)
C3 = m.addConstr(X3 + 0.4 >= X1)
C4 = m.addConstr(-0.230438*X1 + X4 <= 3.4695627)

m.update()

m.optimize()

m.addConstr(3.892116*X1 - X2 + X4 <= 3.492113)

m.update()

m.optimize()
```



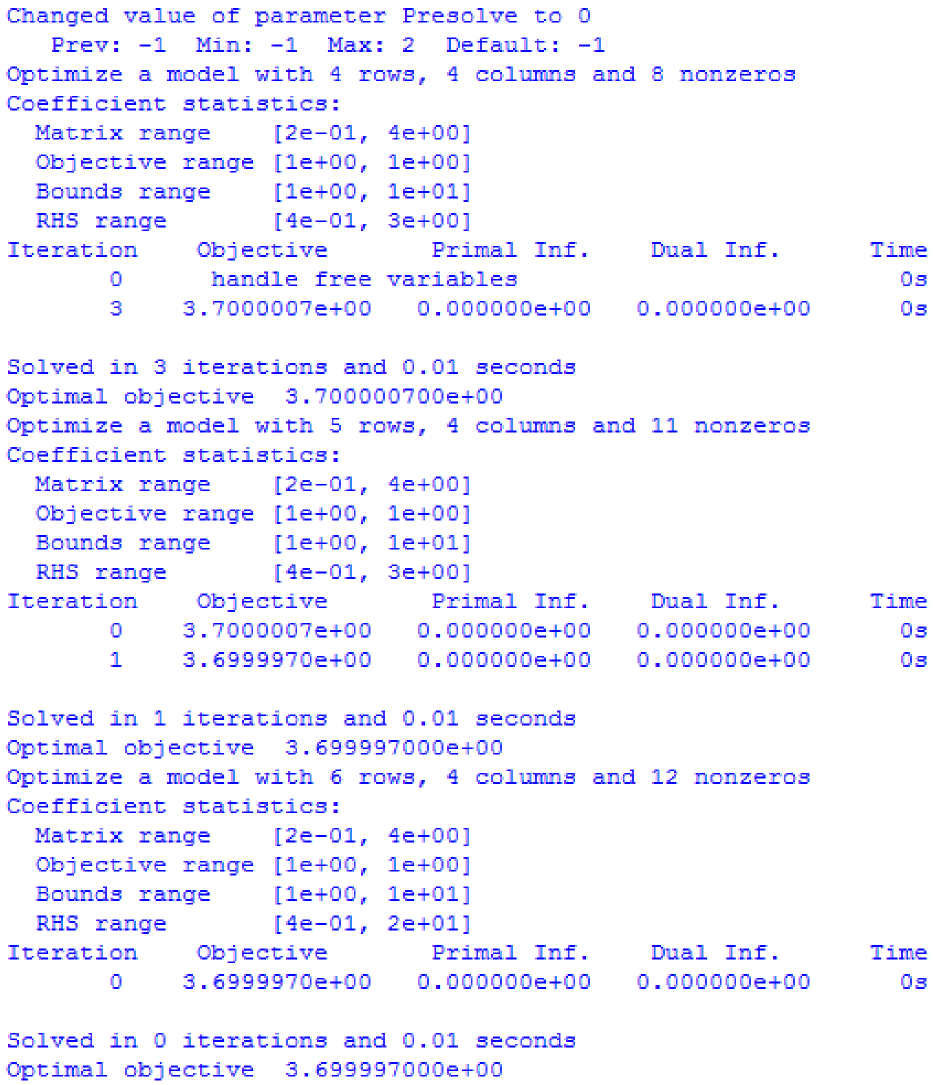
```
Changed value of parameter Presolve to 0
   Prev: -1  Min: -1  Max: 2  Default: -1
Optimize a model with 4 rows, 4 columns and 8 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0      handle free variables                          0s
       3    3.7000007e+00   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.01 seconds
Optimal objective  3.700000700e+00
Optimize a model with 5 rows, 4 columns and 11 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.7000007e+00   0.000000e+00   0.000000e+00      0s
       1    3.6999970e+00   0.000000e+00   0.000000e+00      0s

Solved in 1 iterations and 0.01 seconds
Optimal objective  3.699997000e+00
```

GurobiPy: Solve (3)



```
from gurobipy import *

m = Model("Gurobi Bug")
m.setParam("Presolve", 0)

X1 = m.addVar(lb=-GRB.INFINITY, ub=1)
X2 = m.addVar(lb=-10, ub=10)
X3 = m.addVar(lb=-GRB.INFINITY, ub=GRB.INFINITY)
X4 = m.addVar(lb=-GRB.INFINITY, ub=10, obj=1)

m.modelSense = GRB.MAXIMIZE

m.update()

C1 = m.addConstr(X2 == 4.1*X1)
C2 = m.addConstr(X1 >= 0.4 - X3)
C3 = m.addConstr(X3 + 0.4 >= X1)
C4 = m.addConstr(-0.230438*X1 + X4 <= 3.4695627)

m.update()

m.optimize()

m.addConstr(3.892116*X1 - X2 + X4 <= 3.492113)

m.update()

m.optimize()

m.addConstr(X4 <= 20)

m.update()

m.optimize()
```

```
Obj =  3.699997
X1 =  1.0
X2 =  4.1
X3 =  0.6
X4 =  3.699997
Dual on C1 =  1.0
```

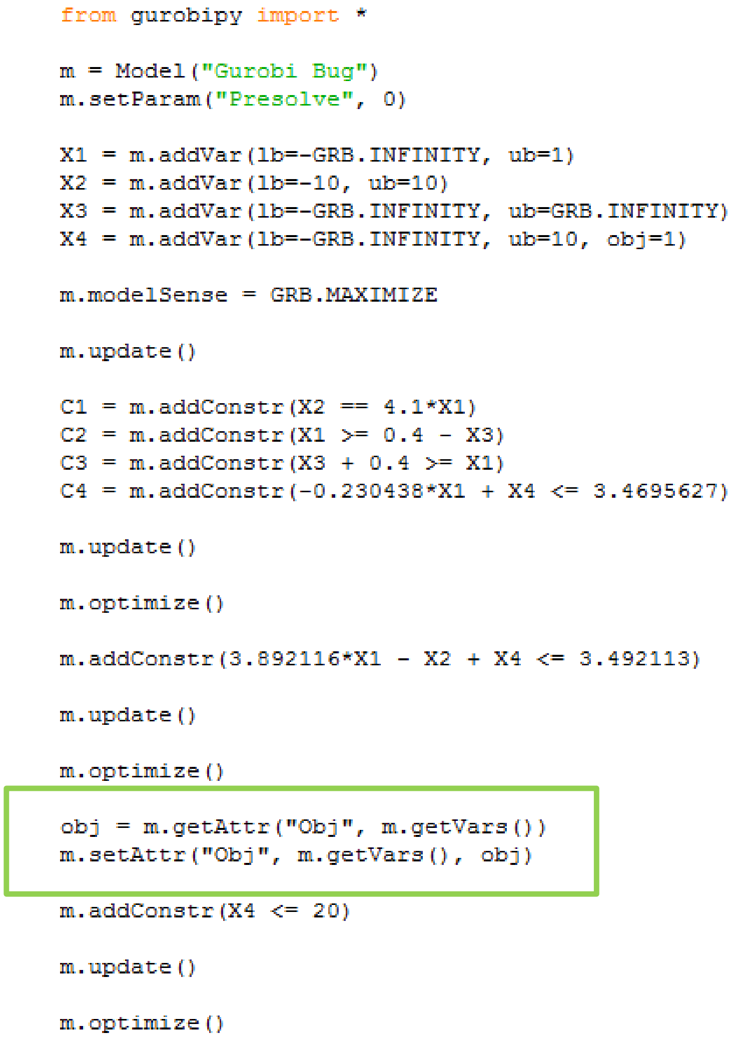


```
Changed value of parameter Presolve to 0
   Prev: -1  Min: -1  Max: 2  Default: -1
Optimize a model with 4 rows, 4 columns and 8 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0      handle free variables                          0s
       3    3.7000007e+00   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.01 seconds
Optimal objective  3.700000700e+00
Optimize a model with 5 rows, 4 columns and 11 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.7000007e+00   0.000000e+00   0.000000e+00      0s
       1    3.6999970e+00   0.000000e+00   0.000000e+00      0s

Solved in 1 iterations and 0.01 seconds
Optimal objective  3.699997000e+00
Optimize a model with 6 rows, 4 columns and 12 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 2e+01]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.6999970e+00   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds
Optimal objective  3.699997000e+00
```

Now lets do something redundant

```
Obj = m.getAttr(“Obj”, m.getVars())

m.setAttr(“Obj”, m.getVars(), obj)
```

Audience Participation Time



```
from gurobipy import *

m = Model("Gurobi Bug")
m.setParam("Presolve", 0)

X1 = m.addVar(lb=-GRB.INFINITY, ub=1)
X2 = m.addVar(lb=-10, ub=10)
X3 = m.addVar(lb=-GRB.INFINITY, ub=GRB.INFINITY)
X4 = m.addVar(lb=-GRB.INFINITY, ub=10, obj=1)

m.modelSense = GRB.MAXIMIZE

m.update()

C1 = m.addConstr(X2 == 4.1*X1)
C2 = m.addConstr(X1 >= 0.4 - X3)
C3 = m.addConstr(X3 + 0.4 >= X1)
C4 = m.addConstr(-0.230438*X1 + X4 <= 3.4695627)

m.update()

m.optimize()

m.addConstr(3.892116*X1 - X2 + X4 <= 3.492113)

m.update()

m.optimize()

obj = m.getAttr("Obj", m.getVars())
m.setAttr("Obj", m.getVars(), obj)

m.addConstr(X4 <= 20)

m.update()

m.optimize()
```

a) Unbounded

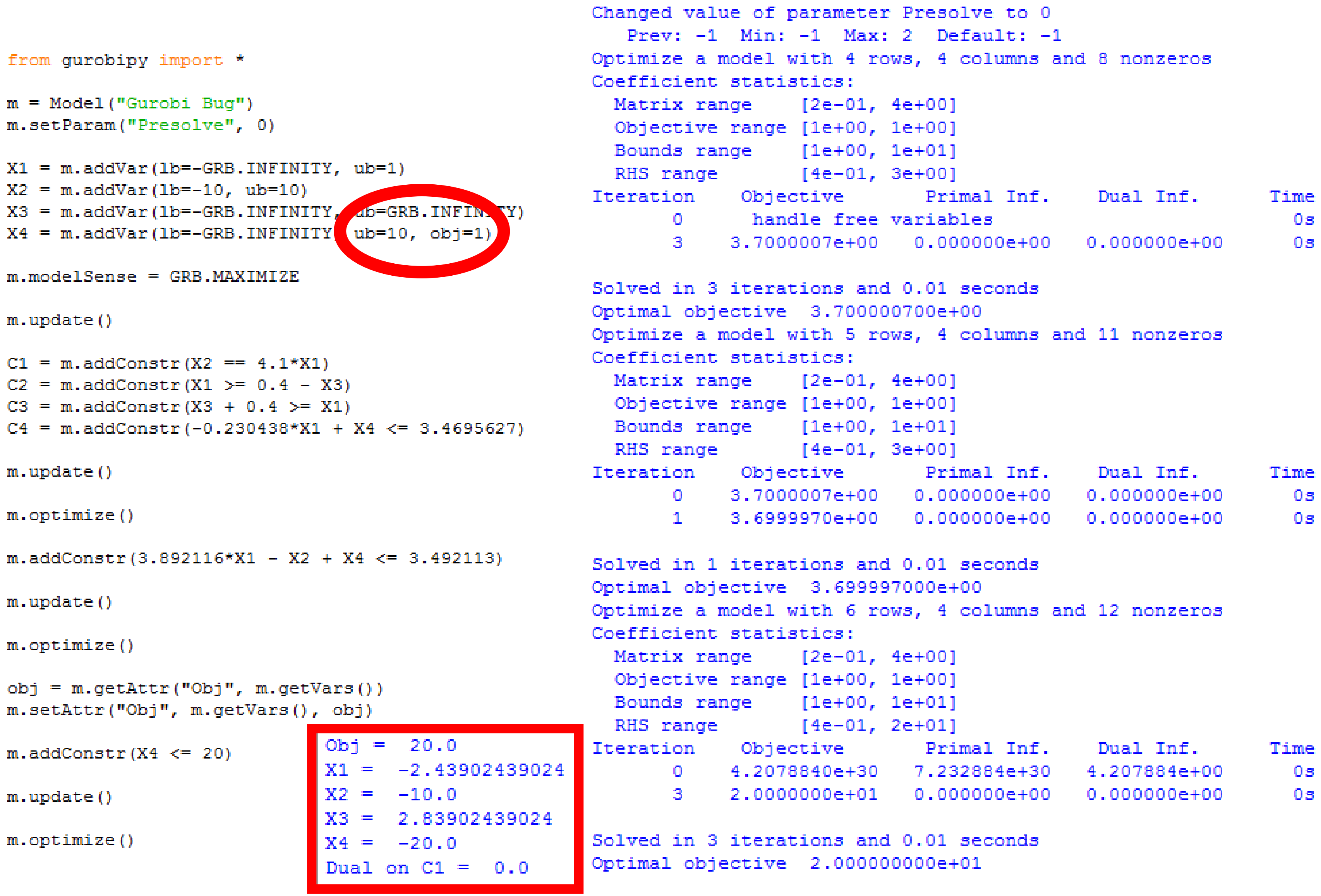
b) Infeasible

c) 3.699997
optimal solution

d) 3.7000007
first solution

e) 20
RHS of the last constraint

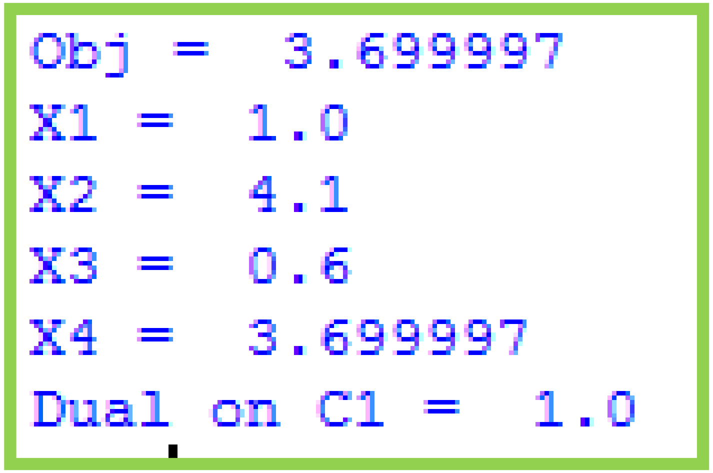
The answer is e) 20!



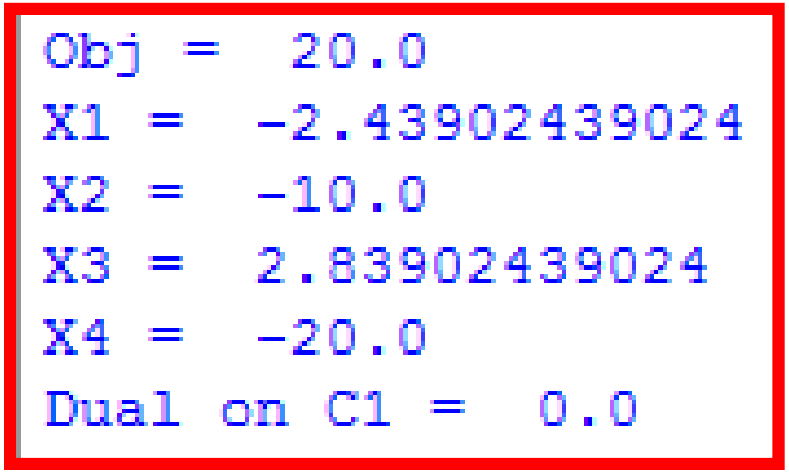
```
from gurobipy import *

m = Model("Gurobi Bug")
m.setParam("Presolve", 0)

X1 = m.addVar(lb=-GRB.INFINITY, ub=1)
X2 = m.addVar(lb=-10, ub=10)
X3 = m.addVar(lb=-GRB.INFINITY, ub=GRB.INFINITY)
X4 = m.addVar(lb=-GRB.INFINITY, ub=10, obj=1)

m.modelSense = GRB.MAXIMIZE

m.update()

C1 = m.addConstr(X2 == 4.1*X1)
C2 = m.addConstr(X1 >= 0.4 - X3)
C3 = m.addConstr(X3 + 0.4 >= X1)
C4 = m.addConstr(-0.230438*X1 + X4 <= 3.4695627)

m.update()

m.optimize()

m.addConstr(3.892116*X1 - X2 + X4 <= 3.492113)

m.update()

m.optimize()

obj = m.getAttr("Obj", m.getVars())
m.setAttr("Obj", m.getVars(), obj)

m.addConstr(X4 <= 20)

m.update()

m.optimize()
```

```
Obj =  20.0
X1 =  -2.43902439024
X2 =  -10.0
X3 =  2.83902439024
X4 =  -20.0
Dual on C1 =  0.0
```

```
Changed value of parameter Presolve to 0
   Prev: -1  Min: -1  Max: 2  Default: -1
Optimize a model with 4 rows, 4 columns and 8 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0      handle free variables                          0s
       3    3.7000007e+00   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.01 seconds
Optimal objective  3.700000700e+00
Optimize a model with 5 rows, 4 columns and 11 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 3e+00]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    3.7000007e+00   0.000000e+00   0.000000e+00      0s
       1    3.6999970e+00   0.000000e+00   0.000000e+00      0s

Solved in 1 iterations and 0.01 seconds
Optimal objective  3.699997000e+00
Optimize a model with 6 rows, 4 columns and 12 nonzeros
Coefficient statistics:
  Matrix range    [2e-01, 4e+00]
  Objective range [1e+00, 1e+00]
  Bounds range    [1e+00, 1e+01]
  RHS range       [4e-01, 2e+01]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    4.2078840e+30   7.232884e+30   4.207884e+00      0s
       3    2.0000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 3 iterations and 0.01 seconds
Optimal objective  2.000000000e+01
```

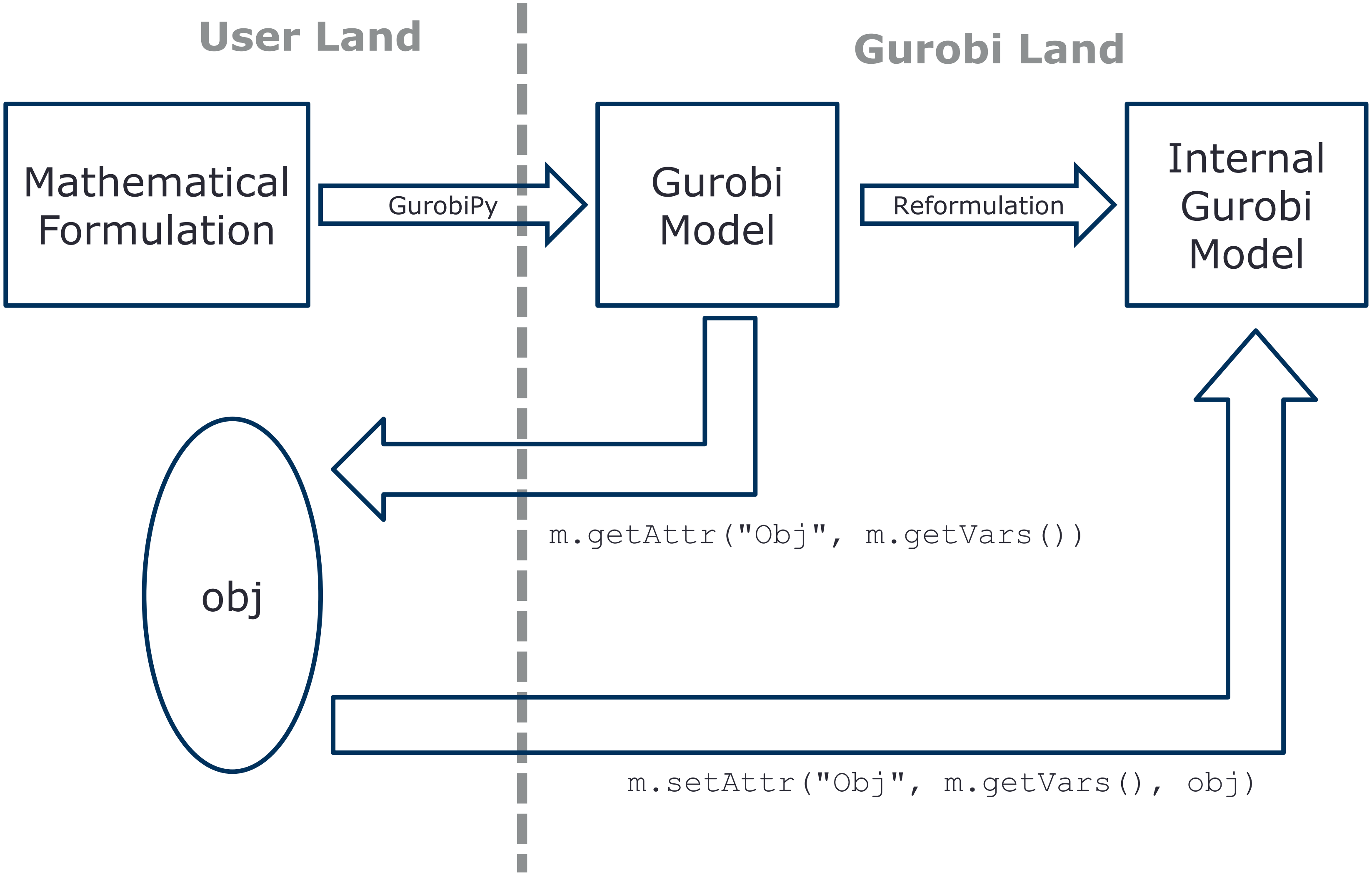


```
Obj =  3.699997
X1 =  1.0
X2 =  4.1
X3 =  0.6
X4 =  3.699997
Dual on C1 =  1.0
```



```
Obj =  20.0
X1 =  -2.43902439024
X2 =  -10.0
X3 =  2.83902439024
X4 =  -20.0
Dual on C1 =  0.0
```

So what’s going on?



```
m.getAttr("Obj", m.getVars())
```

```
m.setAttr("Obj", m.getVars(), obj)
```

So what’s going on?

```
max                     X4
s.t.
      4.1X1 - X2             == 0
         X1       + X3       >= 0.4
         X1       - X3       <= 0.4
-0.230438X1           + X4 <= 3.4695627
X1 ∈ (-∞, 1] X2 ∈ [-10, 10]
X3 ∈ (-∞, ∞) X4 ∈ (-∞, 10]
```

```
max                     X4
s.t.
      4.1X1 - X2             == 0
         X1       + X3       >= 0.4
         X1       - X3       <= 0.4
-0.230438X1           + X4 <= 3.4695627
 3.892116X1 - X2      + X4 <= 3.492113
X1 ∈ (-∞, 1] X2 ∈ [-10, 10]
X3 ∈ (-∞, ∞) X4 ∈ (-∞, 10]
```

```
max                   - X4
s.t.
      4.1X1 - X2             == 0
         X1       + X3       >= 0.4
         X1       - X3       <= 0.4
-0.230438X1           - X4 <= 3.4695627
 3.892116X1 - X2      - X4 <= 3.492113
X1 ∈ (-∞, 1] X2 ∈ [-10, 10]
X3 ∈ (-∞, ∞) X4 ∈ [-10, ∞)
```

```
max                   + X4
s.t.
      4.1X1 - X2             == 0
         X1       + X3       >= 0.4
         X1       - X3       <= 0.4
-0.230438X1           - X4 <= 3.4695627
 3.892116X1 - X2      - X4 <= 3.492113
X1 ∈ (-∞, 1] X2 ∈ [-10, 10]
X3 ∈ (-∞, ∞) X4 ∈ [-10, ∞)
```

```
max                   + X4
s.t.
      4.1X1 - X2             == 0
         X1       + X3       >= 0.4
         X1       - X3       <= 0.4
-0.230438X1           - X4 <= 3.4695627
 3.892116X1 - X2      - X4 <= 3.492113
                        X4 <= 20
X1 ∈ (-∞, 1] X2 ∈ [-10, 10]
X3 ∈ (-∞, ∞) X4 ∈ [-10, ∞)
```

What we know from Gurobi

You need

1. A variable with a finite upper bound and an infinite lower bound

   Value to go variable in SDDP…

2. A model that Gurobi chooses not to rescale

   Hard to control…

3. To set some coefficients in memory and resolve

   Which is what SDDP does…