

This document will help you to install python, pyomo and setup everything.

Anaconda is a python distribution. It makes it easy to install python and will provide access to most python libraries.

Installing Anaconda:

Use the following links according to your OS.

Windows users: <https://docs.anaconda.com/anaconda/install/windows/>

macOS users: <https://docs.anaconda.com/anaconda/install/mac-os/>

Linux users: <https://docs.anaconda.com/anaconda/install/linux/>

Installing the Latest Pyomo Release:

- 1) Open Anaconda Navigator -> open terminal
- 2) Use any of the following two options to install pyomo
 - Install pyomo with pip using the following command: `pip install pyomo`
 - Installing pyomo with conda using the following command: `conda install -c conda-forge pyomo`
- 3) Install optimization solvers ipopt and glpk using the following commands:
 - Glpk: `conda install -c conda-forge glpk`
 - Ipopt: `conda install -c conda-forge ipopt=3.11.1`

- 4) After finishing the installation, test that pyomo works with a simple linear programming example:

```
from pyomo.environ import *
#Create a simple model
model = ConcreteModel()
model.x = Var(bounds=(45.0,100.0),initialize=35.0)
model.y = Var(bounds=(0.0,100.0),initialize=5.0)
model.c1 = Constraint(expr=(50.0*model.x + 24.0*model.y <= 2400.0))
model.c2 = Constraint(expr=(30.0*model.x + 33.0*model.y <= 2100.0))
model.objective = Objective(expr=model.x + model.y - 50.0, sense=maximize)
#Solve LP using solver glpk
solver = SolverFactory("glpk")
solver.solve(model)
model.pprint()
```

- 5) Test that pyomo works with a simple non-linear programming example:

```
from pyomo.environ import *
#Create a simple model
model = ConcreteModel()
model.x = Var(bounds=(1.0,10.0),initialize=5.0)
model.y = Var(within=Binary)
model.c1 = Constraint(expr=(model.x-4.0)**2 - model.x <= 50.0*(1-model.y))
model.c2 = Constraint(expr=model.x**2+1.0 <= 50.0*model.y)
```

```

model.objective = Objective(expr=model.x, sense=minimize)
#Solve MINLP using ipopt
solver = SolverFactory('ipopt')
solver.solve(model)
model.pprint()

```

Install Gurobi

Add the Gurobi channel into your Anaconda platform, and then install the gurobi package from this channel.

- `conda config --add channels http://conda.anaconda.org/gurobi`
- `conda install gurobi`

Check if the installed solver gurobi works with a simple example:

```

from pyomo.environ import *
#Create a simple model
model = ConcreteModel()
model.x = Var(bounds=(1.0,10.0), initialize=5.0)
model.y = Var(within=Binary)
model.c1 = Constraint(expr=(model.x-4.0)**2 - model.x <= 50.0*(1-model.y))
model.c2 = Constraint(expr=model.x**2+1.0 <= 50.0*model.y)
model.objective = Objective(expr=model.x, sense=minimize)
#Solve MINLP using gurobi
solver = SolverFactory("gurobi", solver_io="python")
solver.solve(model)
model.pprint()

```

Install BARON:

1. Note, you'll be installing the demo license and the BARON demo mode handles up to 10 constraints and variables and up to 50 nonlinear operations.
2. Use the following link: <https://minlp.com/baron-downloads>
3. Download and install the executable package
4. Store it somewhere in a directory or a path that is easy to access or that you prefer
5. Add the directory/gurobi path to system path. Use the following links for any help:
 - MacOS: <https://www.architectryan.com/2012/10/02/add-to-the-path-on-mac-os-x-mountain-lion/>
 - Windows: <https://www.architectryan.com/2018/03/17/add-to-the-path-on-windows-10/>

check if the installed solver BARON works with a simple example:

```

from pyomo.environ import *
#Create a simple model
model = ConcreteModel()
model.x = Var(bounds=(1.0,10.0), initialize=5.0)
model.y = Var(within=Binary)
model.c1 = Constraint(expr=(model.x-4.0)**2 - model.x <= 50.0*(1-model.y))
model.c2 = Constraint(expr=model.x**2+1.0 <= 50.0*model.y)
model.objective = Objective(expr=model.x, sense=minimize)

```

```
#Solve MINLP using BARON
solver = SolverFactory("baron")
solver.solve(model)
model.pprint()
```

Install CPLEX

Use the following command on terminal

- `conda install -c ibmdecisionoptimization cplex`

Check if the installed solver CPLEX works with a simple example:

```
from pyomo.environ import *
#Create a simple model
model = ConcreteModel()
model.x = Var(bounds=(1.0,10.0),initialize=5.0)
model.y = Var(within=Binary)
model.c1 = Constraint(expr=(model.x-4.0)**2 - model.x <= 50.0*(1-model.y))
model.c2 = Constraint(expr=model.x**2+1.0 <= 50.0*model.y)
model.objective = Objective(expr=model.x, sense=minimize)
#Solve MINLP using cplex
solver = SolverFactory("cplex_direct")
solver.solve(model)
model.pprint()
```