

1.1 Installing a Pyomo/Python Development Environment

Step 1. Install Anaconda

Developing scientific and engineering applications in [Python](#) requires an interpreter for a particular version of the Python language, a collection of previously developed software libraries, and additional development tools including editors and package managers. Together these elements comprise a Python distribution.

There are many [Python distributions](#) available from commercial and free sources. The Anaconda distribution available from [Anaconda.com](#) is among the most complete and best known distributions currently available, and is available as a [free download](#) or in commercially supported enterprise version. Anaconda includes

- a Python interpreter,
- a user interface [Anaconda Navigator](#) providing access to software development tools,
- pre-installed versions of major python libraries,
- the [conda](#) package manager to manage python packages and environments.

Installation Procedure:

1. If you have previously installed Anaconda and wish to start over, then a first step is to [uninstall the earlier version](#). While it is possible to maintain multiple versions of Anaconda, there are problems that can arise when installing new packages. Uninstalling prior installations of Anaconda installations is the easiest way to avoid those problems.
2. [Download](#) a version of Anaconda appropriate for your laptop. Unless you have a specific reason to use an earlier version of the Python language, download the 64-bit graphical installer for the latest version of Python (currently Python 3.6).
3. Locate and launch the graphical installer from your download directory. Either follow the prompts or consult these more [detailed instructions](#). Normally you will want to use the default choices to install Anaconda into your home folder (a.k.a. directory) for your use only. Generally there is no need to install the optional Microsoft VSCode.
4. [Verify](#) that your installation is working. For example, you should be able to locate and launch a new application Anaconda Navigator.
5. Install any available package updates. Open a command line (either the Terminal application on a Mac located look in the Applications/Utilities folder, or the Command Prompt on Windows), and execute the following two commands on separate lines.

```
conda update conda
conda update anaconda
```

If everything is working correctly, these commands should download and install any recent updates to the Anaconda package.

Step 2. Install Pyomo

The following commands install Pyomo and dependencies. These commands should be executed one at a time from a [terminal window on MacOS](#) or a [command window on Windows](#).

```
conda install -c conda-forge pyomo
conda install -c conda-forge pyomo.extras
```

Step 3. Install Solvers

Solvers are needed to compute solutions to the optimization models developed using Pyomo. The solvers [glpk](#) (mixed integer linear optimization) and [ipopt](#) (nonlinear optimization) cover a wide range of optimization models that arise in process systems engineering and provide good starting point for learning Pyomo. These are installed with the following commands (again, executed one at a time from a terminal window or command prompt).

```
conda install -c conda-forge glpk
conda install -c conda-forge ipopt
```

At this point you should have a working installation of a Python/Pyomo development environment. If you're just getting started with Pyomo, at this point you should be able to use Anaconda Navigator to open a Jupyter session in a browser, then download and open this notebook in Jupyter. If all's well, the Pyomo model in the cells below should produce useful results.

Step 4. Optional: Compile Ipopt with HSL Solvers

The Ipopt package uses third-party packages for solving linear subproblems which are generally the most time-consuming steps in the solution algorithm. By default, ipopt is distributed with the [mumps](#) solvers. If you expect make extensive use of Ipopt for nonlinear problems, or to be solving larger models with 10's of thousands of variables, then you may wish to recompile Ipopt to use high-performance linear solvers available under license from other sources.

In particular, [HSL makes available a collection of high-performance solvers for use with Ipopt](#) under a free personal license or a free academic license (commercial licenses are also available). These solvers frequently provide a 10x improvement in solution speed. [Instructions for downloading, compiling, and installing the necessary software are available from COIN-OR](#).

When used with Anaconda, it is convenient to install the recompiled Ipopt directly in the anaconda binary library. Given the instructions above, use the option `--prefix=~anaconda3` when calling `configure`.

Step 5. Optional: Install Additional Solvers

The glpk and ipopt solvers are sufficient to handle meaningful Pyomo models with hundreds to several thousand variables and constraints. These solvers are available under open-source licenses and an excellent starting point for building applications in Pyomo. These may be all you need for many problems. However, as applications get large or more complex, there may be a need to install additional solvers.

Gurobi

[Gurobi](#), for example, is a state-of-the-art high performance commercial solver for large-scale linear, mixed-integer linear, and quadratic programming problems. Unlike glpk, Gurobi is a multi-threaded application that can take full advantage of a multi-core laptop. Gurobi offers free licenses with one-year terms for academic use, and trial licenses for commercial use. If your application has outgrown glpk, then you'll almost certainly want to give Gurobi a try.

To install for use in Pyomo, download the standard Gurobi installer and perform the default installation. Follow instructions for registering your copy and obtaining a license. Because Gurobi will be installed outside of the default Anaconda installation, you will need to specify the actual Gurobi executable. On MacOS, for example, the executable is `/usr/local/bin/gurobi.sh`.

COIN-OR CBC

Solving larger problems may wish to install a multi-threaded solver such as the COIN-OR [cbc](#) solvers.

```
conda install -c conda-forge coinbc
```

High performance commercial solvers, such as and [CPLEX](#), are also available at no cost for many academic uses (this is a fantastic deal!), and with trial licenses for commercial use.

In []: