Before we begin the tutorial, please complete steps 1 and (optional) 2. This will ensure that you have the software, data and expertise needed to begin using Switch.

1 Software setup (approx. 1 hour)

This section describes how to install Switch on your computer so it is ready to solve power system planning problems. Switch depends on a collection of mostly open-source software:

- **Switch core modules** and **your extension modules** define a power system optimization model in relatively easy-to-understand Python code
- **Switch** loads the modules you have selected and the corresponding data, constructs an optimization model, solves it and reports results. All of this is done with code written in the Python language.
- **Pyomo** is a general-purpose optimization modeling framework for Python. Switch uses this package to define the elements of your optimization model and call a solver.
- Pyomo converts the Switch model into a standardized, computer-readable form and sends it to an **external solver** (e.g., glpk, cbc, cplex or gurobi). The external solver does the intense computation required to find an optimal plan.
- **Anaconda** provides an easy, standardized way to install Python, Pyomo, glpk and other tools on any computer system.

The instructions below will show you how to setup the following:

- Anaconda Python environment
- Switch and software it depends on (Pyomo and glpk)
- Example models to use in this tutorial
- Atom text editor

All of these tools are open-source and cross-platform, so you should be able to use them on any computer.

1.1 Installing Anaconda and Python

Download and install Anaconda from https://www.anaconda.com/distribution/. I recommend selecting the Python 3+ version, but Python 2.7 will probably work too. This requires about 2 GB of disk space. (You can reduce the disk usage by installing Miniconda instead, from https://docs.conda.io/en/latest/miniconda.html.)

Note that you *do not* need administrator privileges to install the Anaconda environment or add packages to it. If you want to install without administrator privileges on Windows, you should choose "Change Install Location" on the "Installation Type" screen, and set it to install just for you, not other users.

You can skip the option to install Microsoft Visual Studio Code, PyCharm or another code editor; we will use Atom instead.

1.2 Installing Switch, Pyomo and glpk

Open Terminal.app (OS X) or an Anaconda command prompt (Start > Anaconda > Anaconda Prompt). Then type this command and press Enter or return:

```
conda install -c conda-forge switch_model
```

Follow the prompts to install Switch and related software.

You can check that everything has been installed by running these commands:

```
glpsol --version
python --version
pyomo --version
switch --version
```

1.3 Installing data for this tutorial

The hardest part of working with Switch is assembling all the inputs you need to create a complete, working model of your power system. For this tutorial, we will work with some pre-existing datasets, rather than building inputs from scratch. Later, when you want to build your own models, you will probably find it useful to base your inputs off of these examples or others (e.g., at https://github.com/switch-model/switch/tree/master/examples or https://github.com/switch-hawaii/).

Open Terminal.app (OS X) or an Anaconda command prompt (Start > Anaconda > Anaconda Prompt). Then type this command and press Enter or return:

```
conda install conda install git
```

Follow the prompts to install "git".

Then use this command to install the tutorial data in a subdirectory called "switch_tutorial" within the current directory. (You may want to use the "cd" command first to get to a good location to install it.)

```
git clone --depth=1 https://github.com/switch-model/switch_tutorial.git
```

1.4 Installing Atom text editor

For this tutorial, I assume you are using the Atom text editor to view and edit code and data files. You can use a different text editor if you like, but it should be capable of doing programming-oriented tasks, like indenting groups of lines in a text file as a group.

You can download and install the Atom text editor from https://atom.io. On a Mac, "installation" just consists of copying the Atom app from your download folder to your Applications folder.

If you need more info on installing Atom, see
https://flight-manual.atom.io/getting-started/sections/installing-atom/.

If you'd like a quick intro to the program, see
https://flight-manual.atom.io/getting-started/sections/atom-basics/.

1.5 Installing a commercial solver (optional)

The conda command automatically installs the open-source glpk solver along with Switch. This and other open-source solvers (e.g., cbc) are able to solve small test cases, but not fast enough to solve large models. So I generally use proprietary solvers (cplex or gurobi) in practice. These are expensive for professional use, but it is possible to get a trial license before you buy a long-term one. Academics can also get full licenses for free. Note that it is important to get a license (temporary or long-term) for the *full* version of the software, not the free or community version that only supports small problem sizes.

You can complete this tutorial using only glpk, but if you would like to experiment on your own with the larger models, you should install cplex or gurobi. You can obtain licenses and download these solvers from here:

Professional:
https://www.gurobi.com/products/gurobi-optimizer/
https://www.ibm.com/products/ilog-cplex-optimization-studio/pricing

Academic:
https://www.gurobi.com/academia/
https://developer.ibm.com/docloud/blog/2019/07/04/cplex-optimization-studio-for-students-and-academics/

Once you have installed these, you can test that they are available by running one of these pairs of commands from a command prompt or terminal window:

Gurobi on Windows:

```
gurobi.bat
exit()
```

Gurobi on Mac or Linux

```
gurobi.sh
exit()
```

CPLEX

```
cplex
quit
```

2 Introduction to Pyomo and Python (optional, 1-9 hours)

This section points you to some useful, quick introductions to Python and Pyomo. This is optional, but recommended if you will be using Switch extensively or defining custom behaviors (new technologies, rules or policies).

2.1 Introduction to Python (1-6 hours)

I recommend reading sections 3 and 4 of the Python introduction at https://docs.python.org/3/tutorial/ . If you would like a deeper understanding, sections 5 and 6 are also worth reading.

If you want to run sample code from the Python tutorial, you can do so by opening Terminal.app (Mac) or an Anaconda command prompt (Windows), then running "python", then copying code from the tutorial into there. Generally the code marked with ">>>" or "..." is code that you can type to the Python interpreter. But you shouldn't copy the >>> or ... symbols themselves.

2.2 Introduction to Optimization and Pyomo (1-3 hours)

I recommend going through the following sections at https://pyomo.readthedocs.io/ to get an introduction to Pyomo, the optimization software used by Switch.

- Pyomo Overview
- Pyomo Modeling Components
- Solving Pyomo Models

This will enable you to read and write Switch code, which is just Pyomo code applied to power system modeling. i.e., a Switch model is just a Pyomo AbstractModel used to optimize the design of a power system.

Notation: In the Pyomo introduction, you will see problems with a notation like this:

min $c_1x_1 + c_2x_2$
s.t.
$a_{11}\, x_1 + a_{12}\, x_2 \geq b_1$
$a_{21}\, x_1 + a_{22}\, x_2 \geq b_2$
$x_1 \geq 0$
$x_2 \geq 0$

This is a common way to describe mathematical optimization problems. It means "find values for x_1 and x_2 that will minimize the value of $c_1x_1 + c_2x_2$, such that all the specified constraints are satisfied."

In this problem, the *x* values are called **decision variables** (these are numbers that will be chosen when the problem is run), the *a*, *b*, and *c* values are **parameters** (data you know when

you setup the problem), $c_1x_1 + c_2x_2$ is the **objective function** (the value to be minimized or maximized), and the other equations are the **constraints**.

3 Basic modeling with Switch

3.1 Setting up a simple model

3.2 Solving Switch models

3.3 Where to find documentation for Switch

4 Comparing scenarios with Switch

4.1 Defining and solving scenarios

4.2 Data and configuration for case study in Switch 2.0 paper (https://doi.org/10.1016/j.softx.2019.100251)

4.3 Analyzing results

5 Customizing Switch

5.1 Writing your own module (cogen example)

5.2 Debugging models

5.3 Input data workflows