Windows/Linux Installation I

Install a scientific Python distribution

- Any “Scientific Python” should do, but it must include NumPy, SciPy, and matplotlib.
- For Windows, e.g., Python(x,y):
  `https://code.google.com/p/pythonxy/`
- For Ubuntu/Debian, e.g., Spyder (an IDE):
  ```
  sudo apt-get install spyder
  ```

Download CasADi2.3.0

- Windows/Linux zip file available at
  `http://sourceforge.net/projects/casadi/files/CasADi/`
- Unzip to a convenient location (e.g., `C:/Python2.7/casadi` for Windows or `/opt/casadi` for Casadi)

Windows/Linux Installation II

Download our `mpctools` Python package

- Download Mercurial repo (zip link on left):
  `https://hg.cae.wisc.edu/hg/mpc-tools-casadi`
- Unzip to a convenient location.
- Move the `mpctools` sub-folder to where you unzipped `casadi`; the remaining files (examples and documentation) can be left where they are.

To CasADi and `mpctools` to your Python path

- Open a Python interpreter (run `python` from a terminal/command prompt)
- Run the commands `import site; print site.getsitepackages()` to see where your site packages are stored
- In one of the site package folders, make a text file called `casadi.pth`, and type the path to your CasADi installation directory

Mac Installation (Difficult)

Install Python, NumPy, SciPy, matplotlib

- E.g., via Homebrew: `brew install python`
- Packages via `pip`: `pip install ipython matplotlib numpy scipy`

Build and Install CasADi 2.3.0

- You'll have to build from sources.
- See `https://github.com/casadi/casadi/wiki/InstallationMac` for details.
- We can only provide minimal support for this option.

Our Python package

- Download Mercurial repo:
  `https://hg.cae.wisc.edu/hg/mpc-tools-casadi`
- Unzip and move the `mpctools` folder to somewhere on your Python Path

Making Sure Everything Works

First, open a Python interpreter and run `import casadi, mpctools`.

- If this doesn't work, make sure your CasADi folder shows up in `import sys; print sys.path`.
- If you have multiple Python distributions on your machine, don't (or at least make sure you're using the one you think you are).
- Make sure you are using Python 2.7 (not 3.x).

Then, try to run the examples in `mpc-tools-casadi`.

- `runall.py` will run everything and tell you if there are errors.
- You won't see any output, however.

What's in `mpc-tools-casadi`?

A Python package: `mpctools`.

- We recommend putting the `mpctools` folder with CasADi, but it can be anywhere on your Python path.
- In Python, use `import sys; print sys.path` to see what folders are on your path.

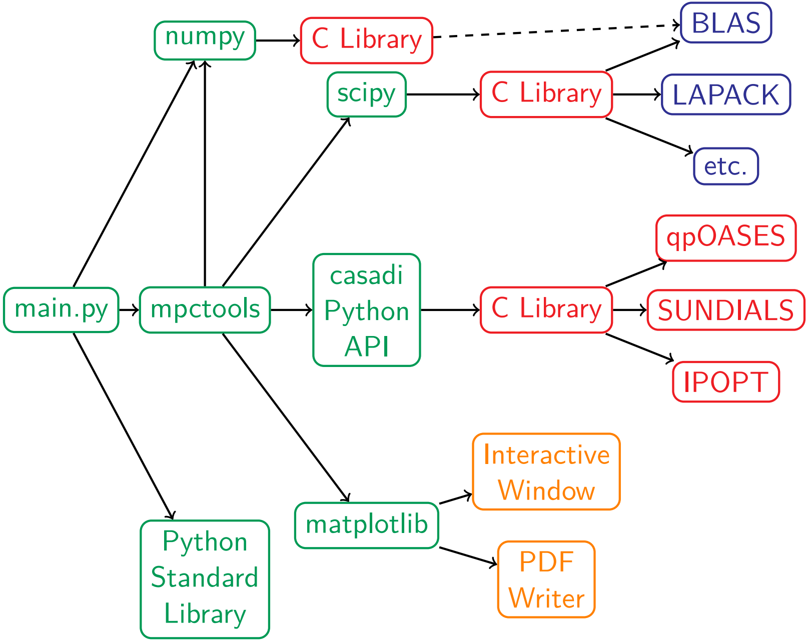
A cheatsheet (in the `doc` folder).

- Should get you started writing your own code.
- Compares plain CasADi vs. CasADi + `mpctools`.

A bunch of example files.

- `nmpcexample.py`: Example of linear vs. nonlinear MPC.
- `cstr_startup.py`: startup and a setpoint change (with no disturbances) for the CSTR system from Example 1.11.
- `nmheexample.py`: NMHE (with EKF to update prior) for a batch reactor (See Example 4.27 in the textbook).

Software Relationships


numpy
C Library
BLAS
scipy
C Library
LAPACK
etc.
qpOASES
main.py
mpctools
casadi Python API
C Library
SUNDIALS
IPOPT
Interactive Window
matplotlib
PDF Writer
Python Standard Library