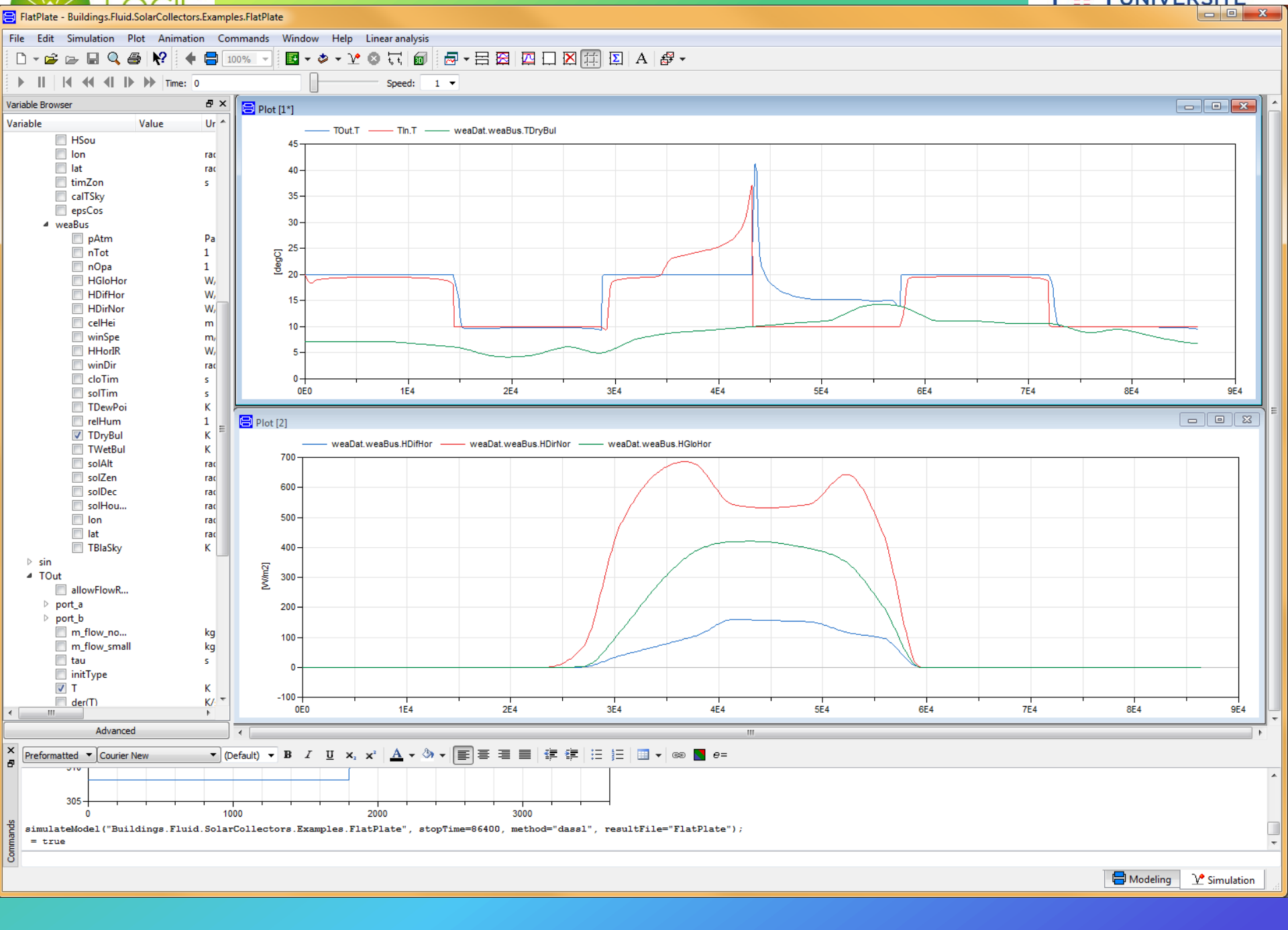
GENSIM scientific school
24-28 oct. 2016 – Porticcio, Corsica

Workflow automation tools

Simon Rouchier, Ph.D.
Université Savoie Mont-Blanc
simon.rouchier@univ-smb.fr

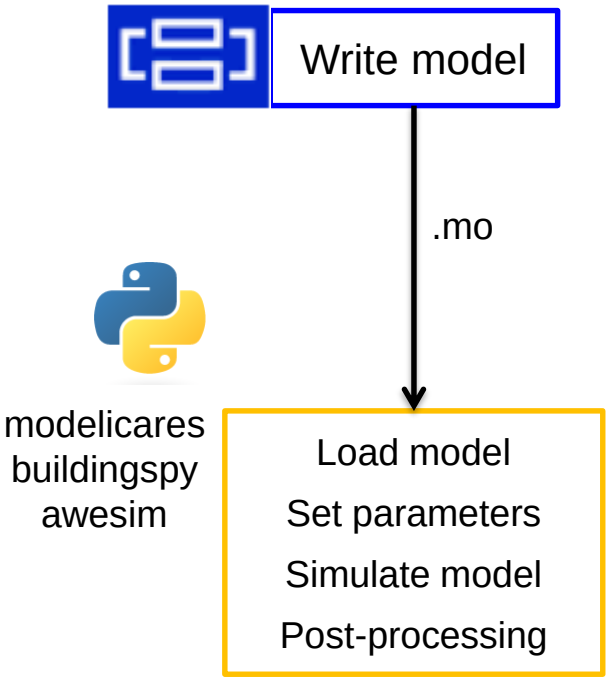

FlatPlate - Buildings.Fluid.SolarCollectors.Examples.FlatPlate
File Edit Simulation Plot Animation Commands Window Help Linear analysis
Time: 0
Speed: 1
Variable Browser
Variable
Value
HSou
lon
lat
timZon
calTSky
epsCos
weaBus
pAtm
nTot
nOpa
HGloHor
HDifHor
HDirNor
celHei
winSpe
HHorIR
winDir
cloTim
solTim
TDewPoi
relHum
TDryBul
TWetBul
solAlt
solZen
solDec
solHou...
lon
lat
TBlaSky
sin
TOut
allowFlowR...
port_a
port_b
m_flow_no...
m_flow_small
tau
initType
T
Advanced
Plot [1*]
TOut.T
TIn.T
weaDat.weaBus.TDryBul
[degC]
0E0
1E4
2E4
3E4
4E4
5E4
6E4
7E4
8E4
9E4
Plot [2]
weaDat.weaBus.HDifHor
weaDat.weaBus.HDirNor
weaDat.weaBus.HGloHor
[W/m2]
Preformatted
Courier New
(Default)
simulateModel("Buildings.Fluid.SolarCollectors.Examples.FlatPlate", stopTime=86400, method="dassl", resultFile="FlatPlate");
= true
Commands
Modeling
Simulation

Why do we need workflow automation tools ?

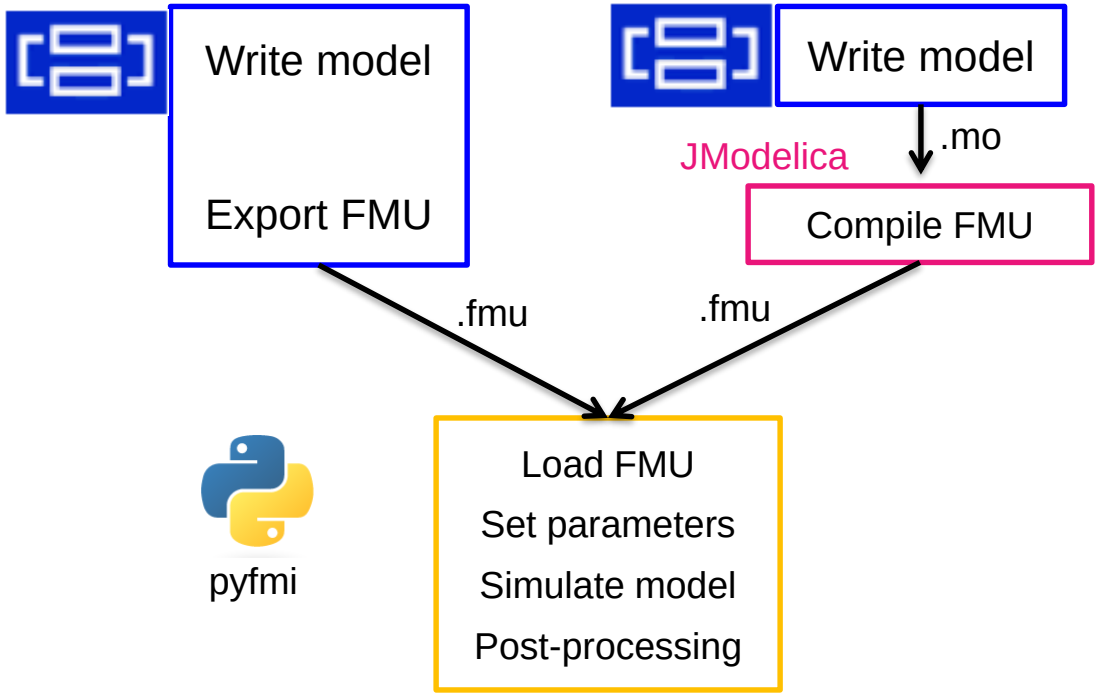
- better post-processing capabilities
- setting up series of simulations (parametric studies, optimisation, system identification...)
- parallel computing

Workflow

Option 1: direct coupling between Modelica and Python



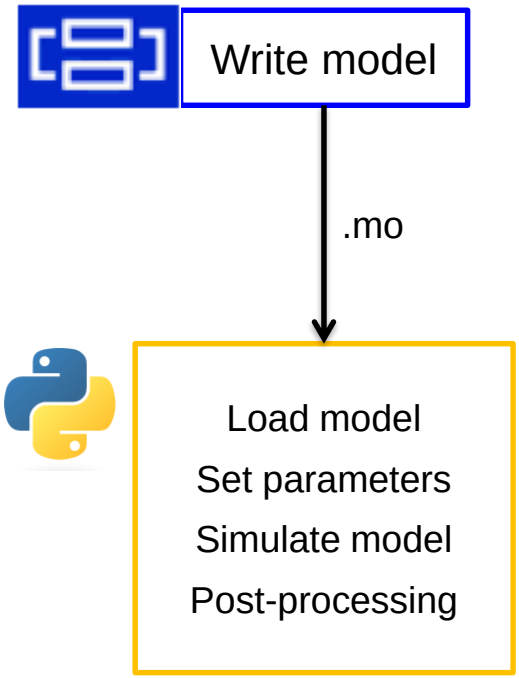
Option 2: using FMUs



Software prerequisites

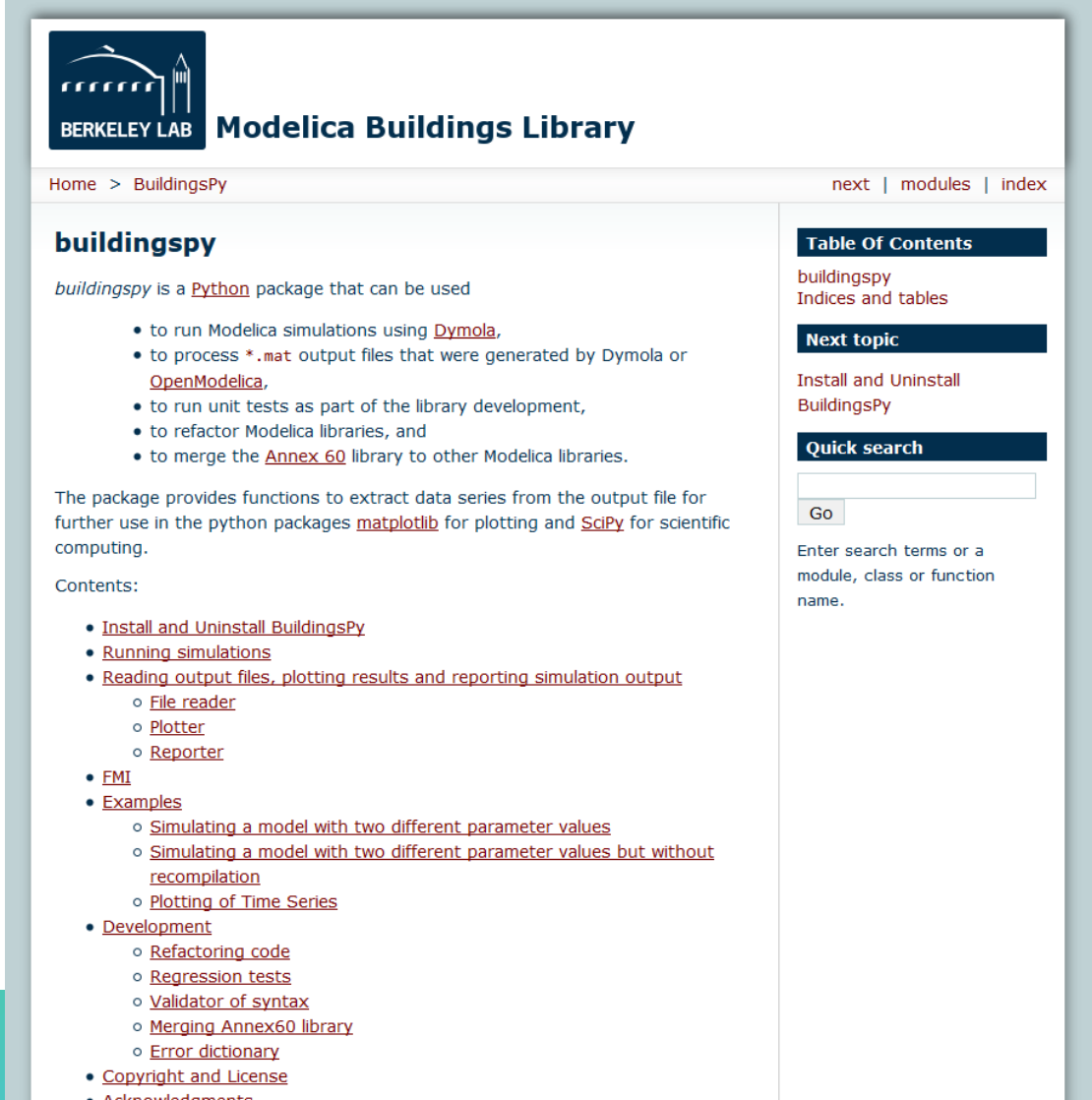
- Modelica environment (preferably Dymola)
- Python environment https://www.continuum.io/downloads
- Python packages
  - BuildingsPy http://simulationresearch.lbl.gov/modelica/buildingspy/
  - PyFMI https://pypi.python.org/pypi/PyFMI

BuildingsPy



BuildingsPy

http://simulationresearch.lbl.gov/modelica/buildingspy/



BERKELEY LAB **Modelica Buildings Library**

Home > BuildingsPy — next | modules | index

buildingspy

buildingspy is a Python package that can be used

- to run Modelica simulations using Dymola,
- to process `*.mat` output files that were generated by Dymola or OpenModelica,
- to run unit tests as part of the library development,
- to refactor Modelica libraries, and
- to merge the Annex 60 library to other Modelica libraries.

The package provides functions to extract data series from the output file for further use in the python packages matplotlib for plotting and SciPy for scientific computing.

Contents:

- Install and Uninstall BuildingsPy
- Running simulations
- Reading output files, plotting results and reporting simulation output
  - File reader
  - Plotter
  - Reporter
- FMI
- Examples
  - Simulating a model with two different parameter values
  - Simulating a model with two different parameter values but without recompilation
  - Plotting of Time Series
- Development
  - Refactoring code
  - Regression tests
  - Validator of syntax
  - Merging Annex60 library
  - Error dictionary
- Copyright and License

Table Of Contents

buildingspy
Indices and tables

Next topic

Install and Uninstall BuildingsPy

Quick search

Go

Enter search terms or a module, class or function name.

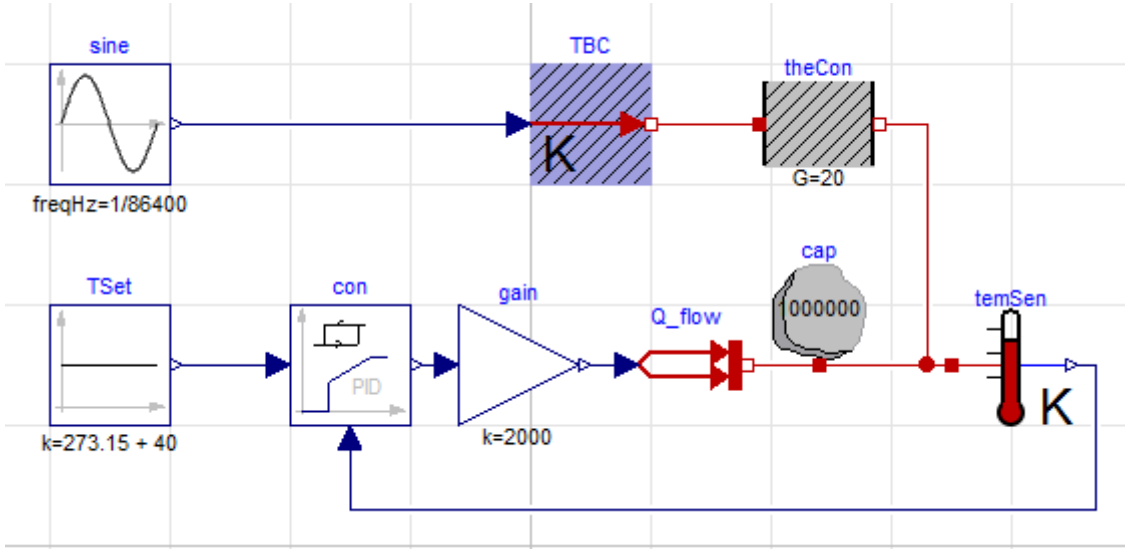
class Simulator

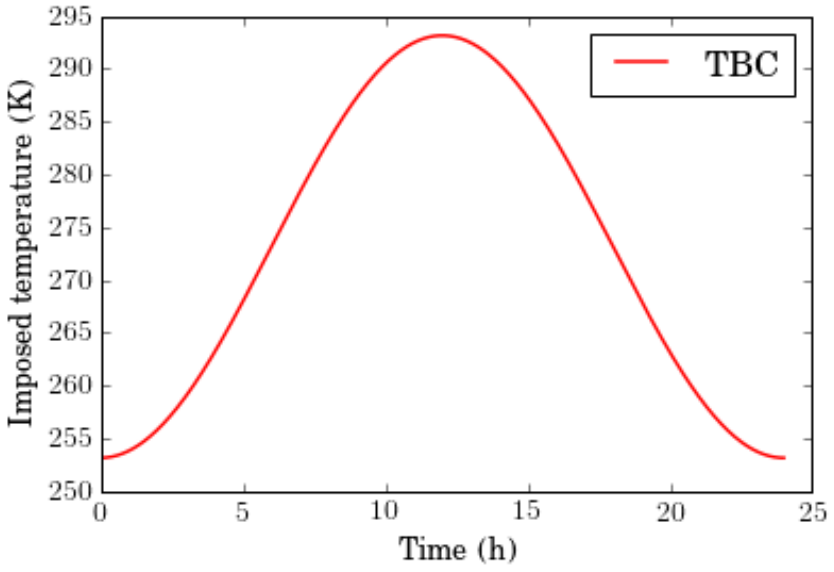
Used to run Modelica simulations, add model modifiers, parameter declarations, set solver type, results directory, stop time...

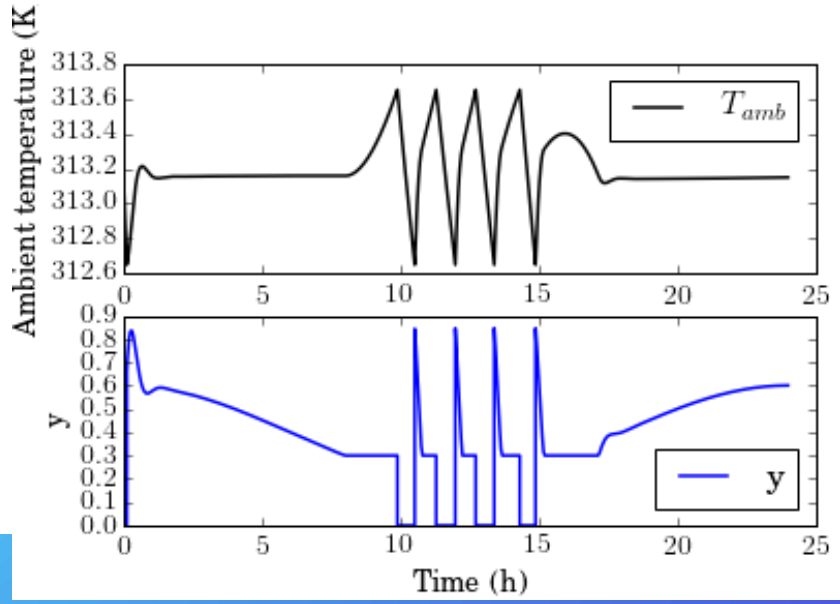
class Reader

reads *.mat files that were generated by Dymola or OpenModelica

Buildings.Controls.Continuous.Examples.PIDHysteresis

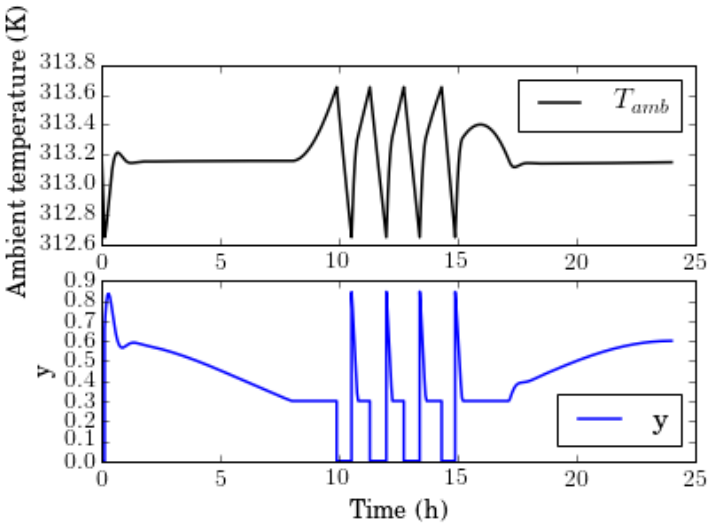






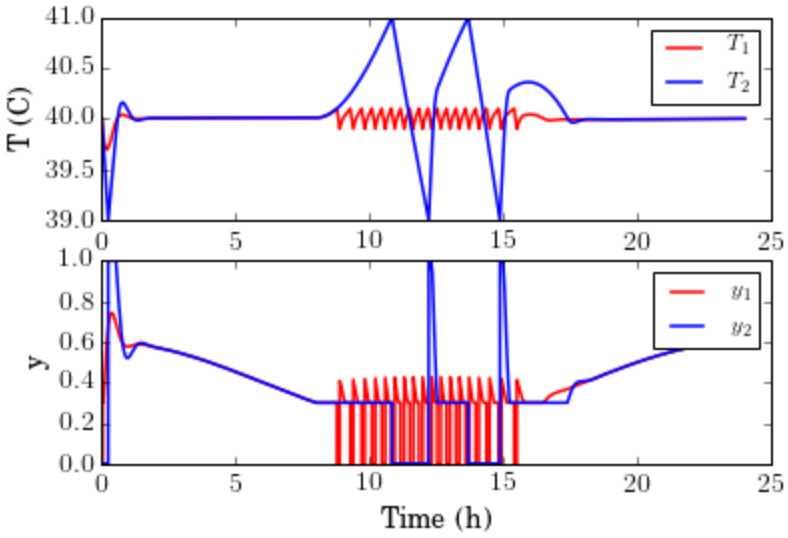
Exercise 1 – Single simulation

Load and run the PIDHysteresis model



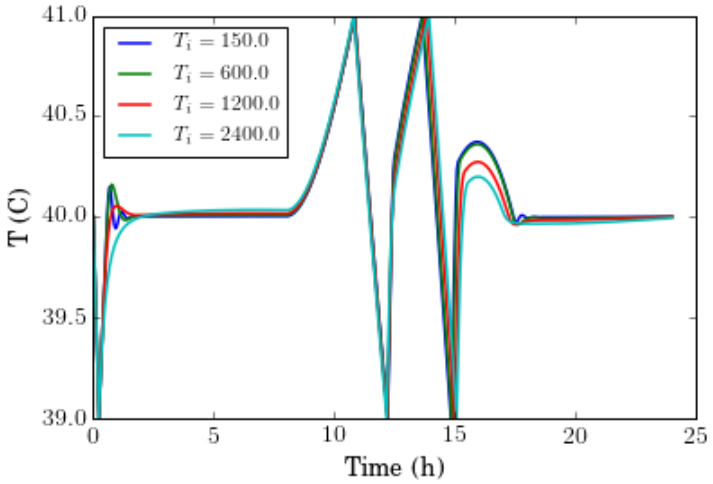
Exercise 2 – Several simulations

2 simulations with different PID offsets

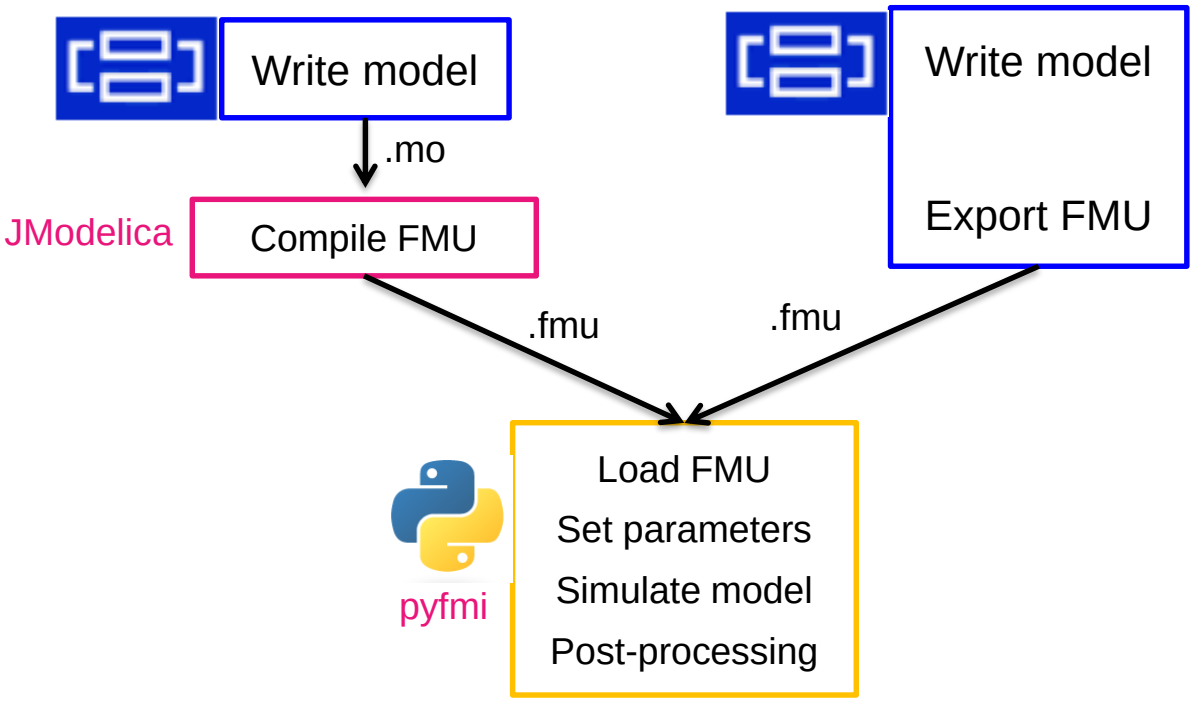


Exercise 3 – Without recompilation

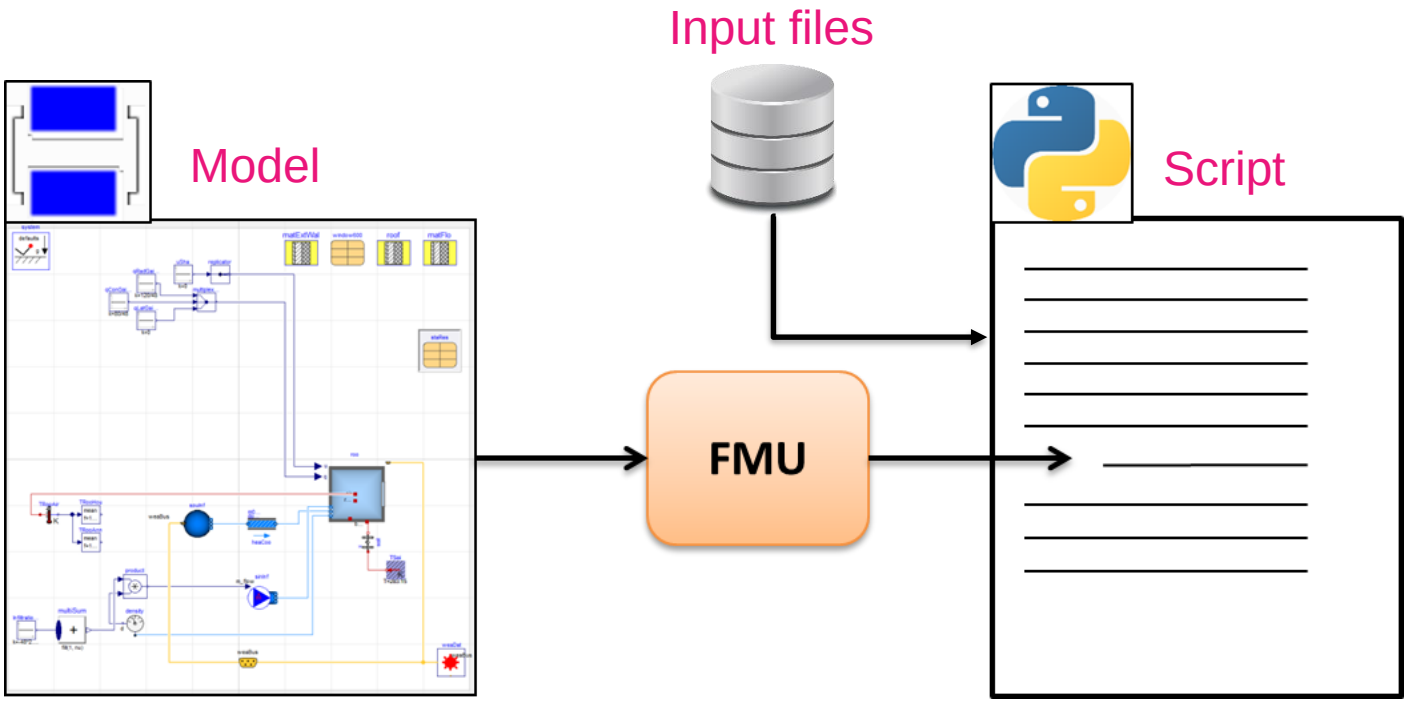
4 simulations with different integrator time constants



PyFMI


Write model
.mo
JModelica
Compile FMU
Write model
Export FMU
.fmu
.fmu
pyfmi
Load FMU
Set parameters
Simulate model
Post-processing

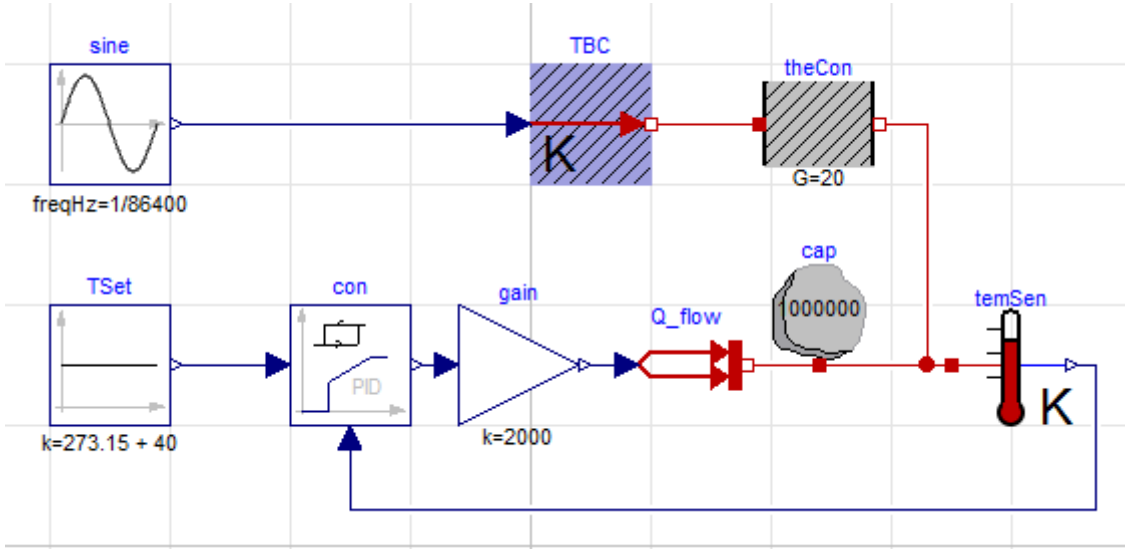
PyFMI

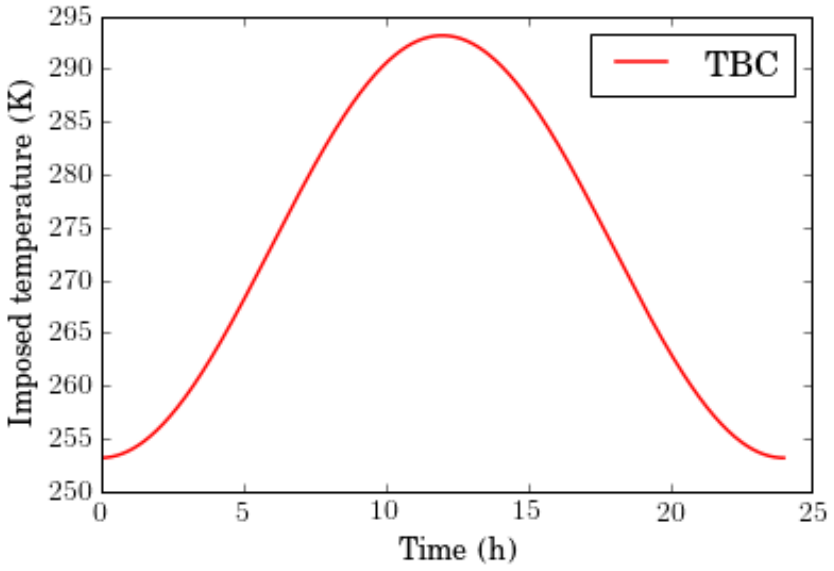


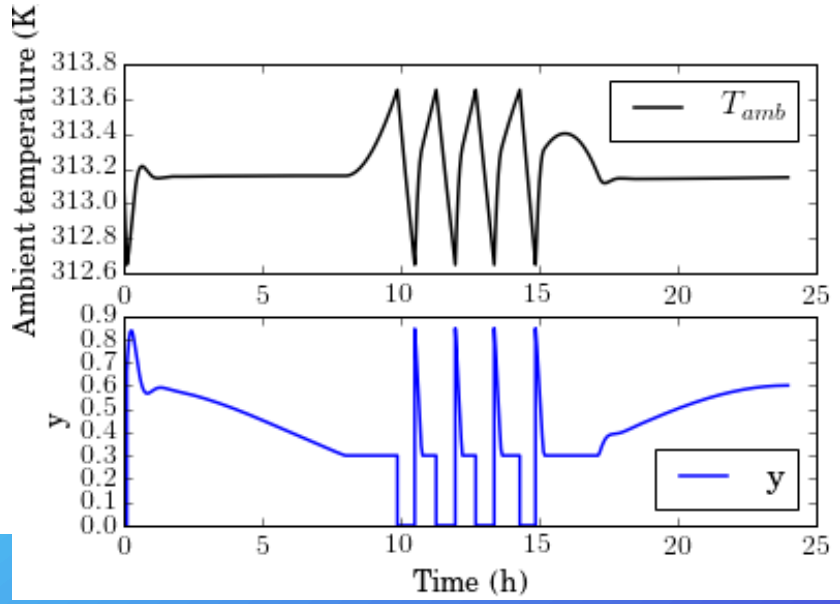
Can read input files outside of the Modelica model
Does not rely on Dymola for the simulation

Only compatible with Python 2.7
Debatable user-friendliness

Buildings.Controls.Continuous.Examples.PIDHysteresis

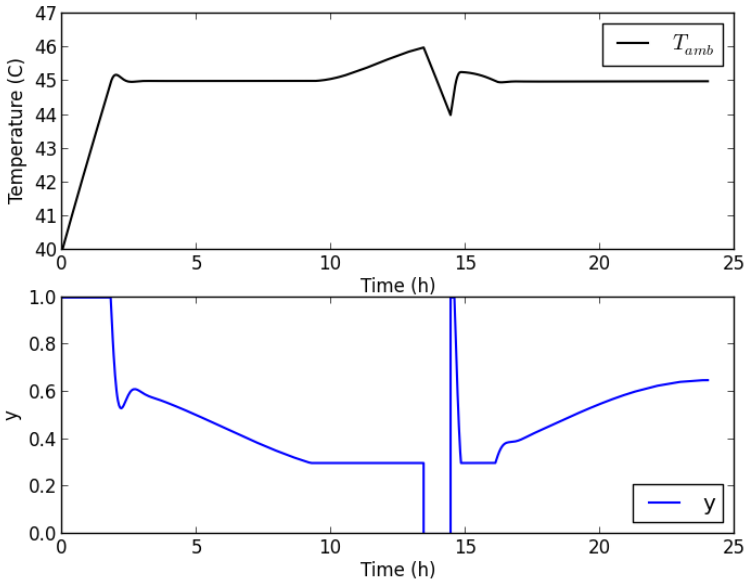






Exercise 1 – Single simulation

Load and run the PIDHysteresis model with a different temperature set point



Exercise 2 - optimisation

Find which value of the PID gain results in the smallest temperature quadratic error