Using SmartGridToolbox

Dan Gordon

May 23, 2018

1 Introduction

SmartGridToolbox is C++14 library for writing software that simulates electricity grids. In particular, is is intended to simulate future grids and smart grids.

This is a very broad brief. We may be interested in exploring optimal control strategies for renewable energy and storage, simulating a microgrid, running coarse grained simulations of a large network; we may need to develop visualisations, to perform academic research, to run simulations for testing during the development of commercial products, or to run scenarios on real networks that inform operations and planning. Including all of this in a monolithic software package would be hard. There is always some aspect of such packages that won't work well with some end uses, and it would be hard to ensure such software could be interfaced with a wide range of existing software.

To handle this problem, we decided that it would be useful to have a software library written in a high-level, general purpose programming language, that could be easily extended and adapted to meet a large variety of needs. C++ was chosen, as it is fast, is fairly universal, and it may be interfaced directly or indirectly with most other modern languages. While requiring a steeper learning curve than, say, python, modern C++ (C++11 and onwards) is still a fairly expressive language. The speed aspect means that it should be possible to write fast, industrial grade software in SmartGridToolbox.

1.1 Who is SmartGridToolbox intended for?

SmartGridToolbox will be of particular interest to academic researchers and graduate students who need to develop and use advanced simulations, and also to people who intend developing software aimed at the energy industry, with particular emphasis on renewables and smart or optimal control strategies for the future grid.

Using SmartGridToolbox requires reasonable C++ programming skills. SmartGridToolbox is not intended as an out of the box end-user application for simulating smart and future grids. It is, rather, primarily a software library, that could be thought of as the starting point for writing these kinds of applications. While useful applications can be written for SmartGridToolbox that involve a minimal degree of coding, perhaps a few dozen lines of code, it is also possible to greatly extend the library to suit a particular need by writing new smart grid component classes, power flow solvers, control algorithms and the like.

1.2 Installing and building SmartGridToolbox

SmartGridToolbox is supported on Linux (e.g. Ubuntu/Debian) and macOS. Building SmartGridToolbox requires various third party libraries and software to be installed.

1.2.1 GNU Autotools

1.2.2 Boost libraries

1.2.3 Armadillo linear algebra library

1.2.4 yaml-cpp

1.3 Simple tutorial

2 API Reference

All SmartGridToolbox classes and functions are defined in the a the `Sgt` namespace. There are also a few macros, mainly for logging and error reporting/handling.

2.1 Core utility classes

2.1.1 Complex numbers

SmartGridToolbox uses `Sgt::Complex` to handle complex numbers. `Sgt::Complex` is just an alias for `std::complex<double>`.

2.1.2 Time

For reasons of simplicity and convenience, SmartGridToolbox uses the `Sgt::Time` class to handle both absolute times and time durations. Absolute times are defined relative to the unix epoch (1 Jan, 1970 00:00:00 UTC). `Sgt::Time` is actually an alias for `boost::posix_time::time_duration`.

2.1.3 Linear algebra; vectors and matrices

SmartGridToolbox makes extensive use of the Armadillo library for vector and matrix classes, and linear algebra purposes. In particular, vectors are normally represented as `arma::Col<double>` or `arma::Col<Complex>`, and matrices are represented as `arma::Mat<double>` or `arma::Mat<Complex>`. Please see the Armadillo documentation for more detail.