

User Manual: High Performance Control C++ Library

Written by David Dos Santos

This manual explains the installing and use of the High-Performance Control C++ library.

As the name suggests, this library has been designed to contribute to the development of high-performance algorithms in the field of control systems. The project currently looks at nonlinear stochastic models of high dimensions.

Currently the library only contains an implementation of the Markov Approximation Method, and also utilises the Continuous Decomposition Method¹.

¹ <http://www.alexgorodetsky.com/index.html>

Installation

The following installations have been tested only on their counterpart version number (i.e. only on Windows 10, only on Ubuntu 16.04 etc.)

All platforms need the following packages:

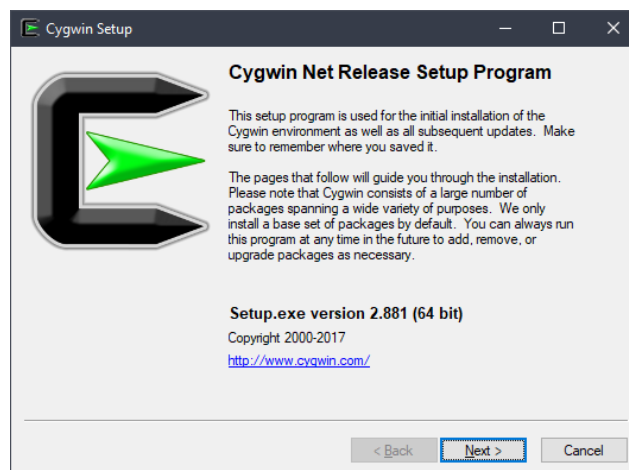
- Git (<https://git-scm.com/downloads>)
- CMake²
- BLAS³ and LAPACK⁴
- GFortran⁵

If using Windows, install the Git package only as the other 3 will best be installed using Cygwin. The installation instructions for Windows 10 covers all necessary steps for this.

For Windows 10:

To install the necessary files, you will need Cygwin
(<https://www.cygwin.com/install.html>)

- When you open the setup you will have something like this. Click Next.



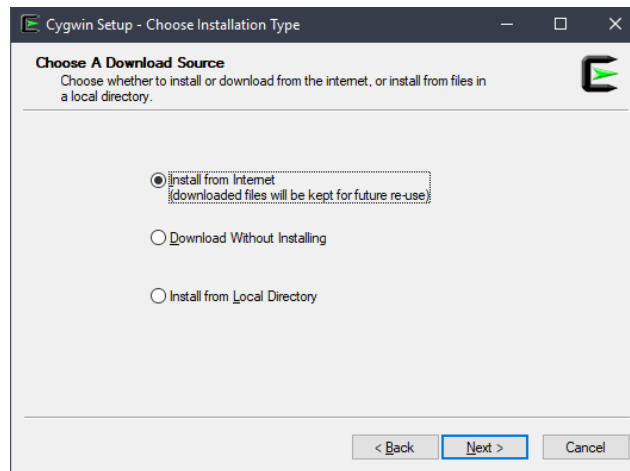
- In the Cygwin installation, choose the download source 'Install from Internet'. Click Next.

² <https://git-scm.com/downloads>

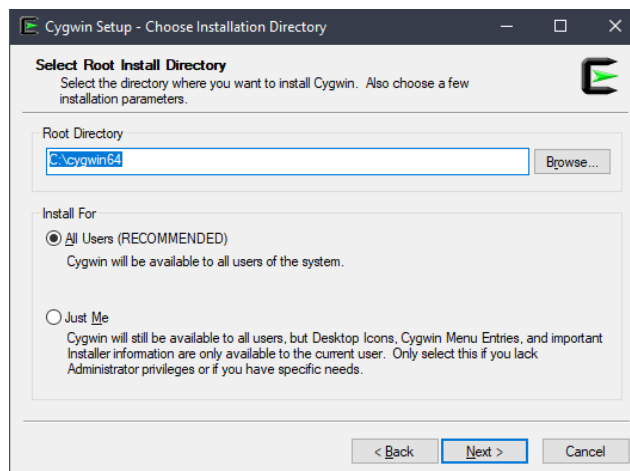
³ <http://www.netlib.org/blas/>

⁴ <http://www.netlib.org/lapack/>

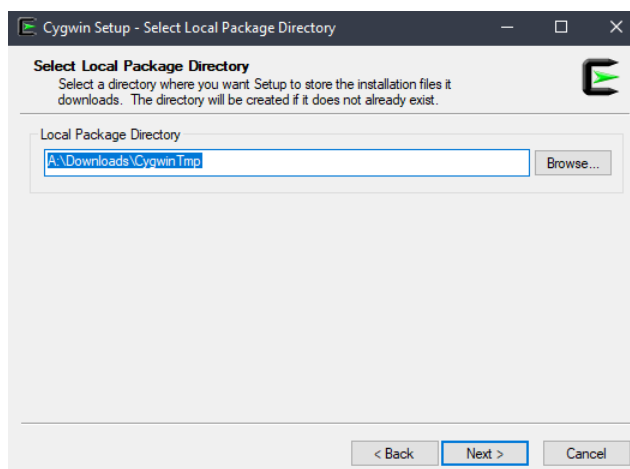
⁵ <https://gcc.gnu.org/wiki/GFortran>



- Select the root directory where you want to install it (and remember this as you may have to add it to your PATH variable later). Click Next.

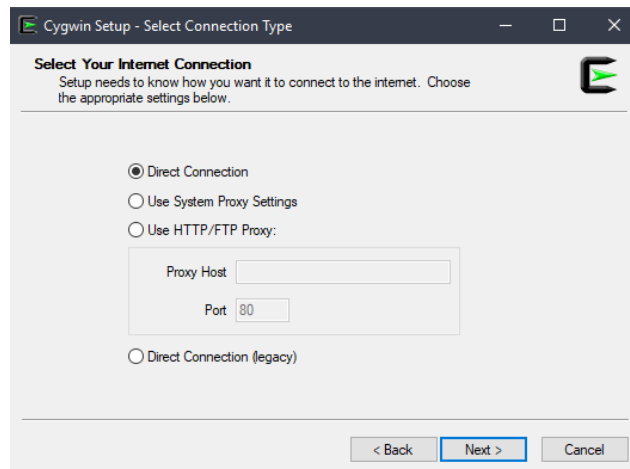


- Select a location to place the temporary downloads (the temporary downloads can be deleted after installation is complete, best to pick somewhere where you will remember to clear it later)

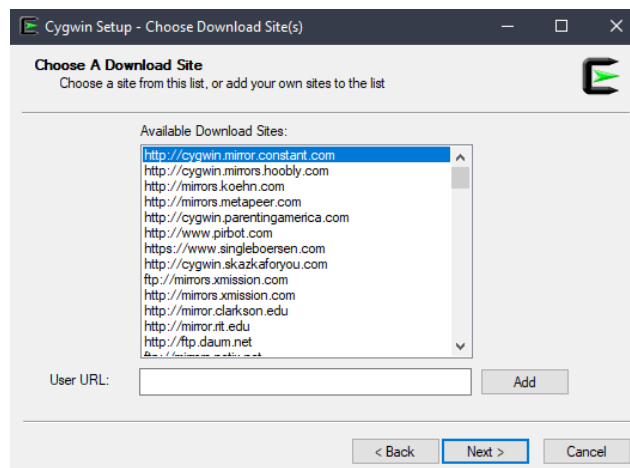


-

- If you have a proxy to use you can enter those details now, else just leave it as Direct Connection.

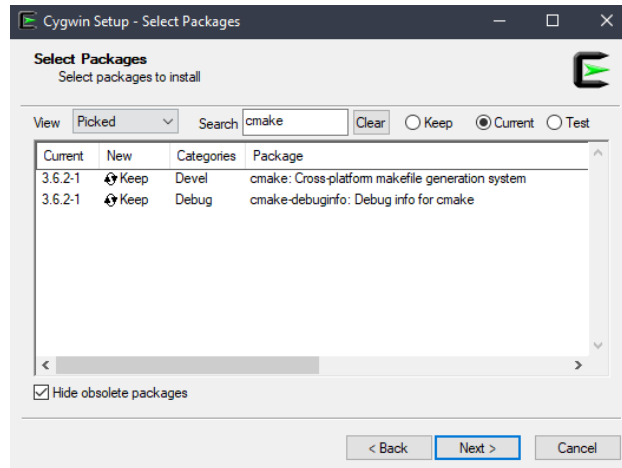


- There are a bunch of different mirrors (or download locations) to choose from, but the only real difference between them is speed and security (due to protocols being used). The most secure are obviously the https, and least secure are ftp. The first one on the list is probably the most reliable.

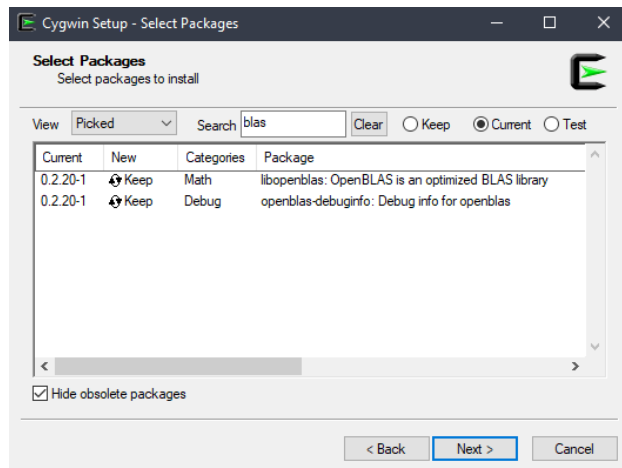


- At the next section, you will be selecting different packages, but don't click the Next button until all packages have been selected. If you forget a package just repeat all processes (because this is how you install new packages to Cygwin anyways)

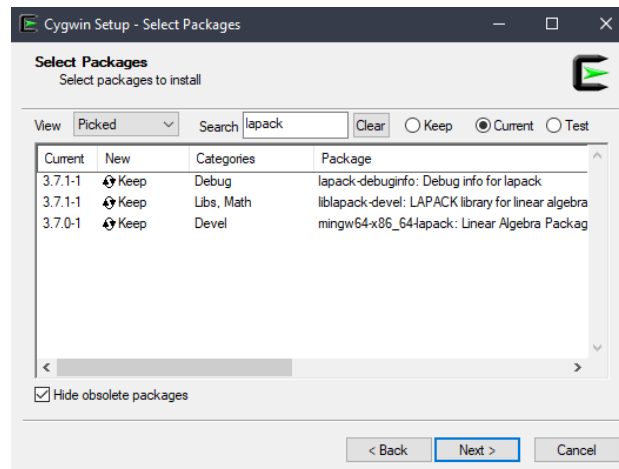
- You need cmake (the cmake-debuginfo is pretty handy too but not necessary). The version you get of cmake may be different, just get the latest version as cmake is backwards compatible. Also note that mine all say “Keep” under the New tab since I already had these packages installed at the time of making this user manual.



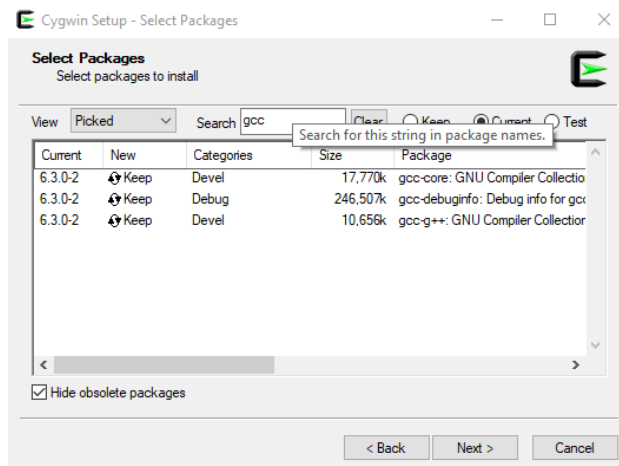
- You need the libopenblas package



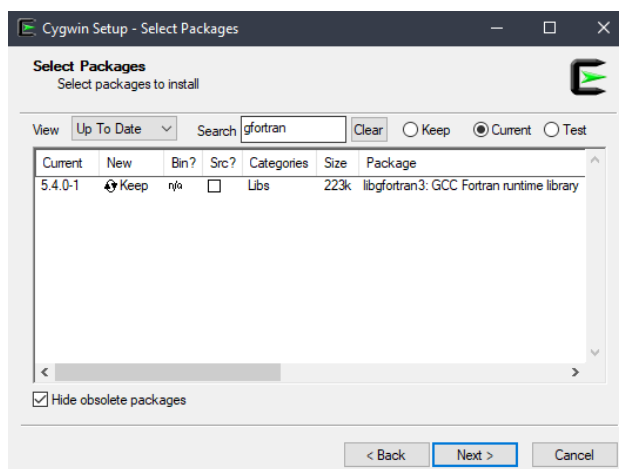
- You need the liblapack package. You don't need the mingw version unless you are using MinGW (which is not covered by this user manual)



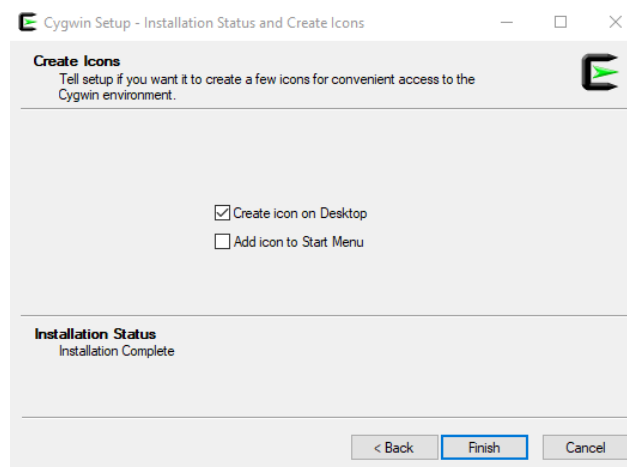
- You need the gcc, gcc-g++ and gcc-debuginfo packages



- The final package will be the libgfortran package, as the blas and lapack are originally Fortran based



- Now click Next and it will have another screen summarising the components to be installed (including their prerequisite parts), click Next.



- Assuming you didn't have any firewall or permission issues, you should finish with this

Now you are ready to copy the Git repository and begin installing the High-Performance-Control library. Run the following commands in a terminal (e.g. Powershell, Command-prompt, or your shiny new Cygwin terminal):

```
git clone https://github.com/revolutionized/High-Performance-Control.git High-Performance-Control
cd High-Performance-Control
```

Once this completes, you can type in the terminal (But not the Cygwin terminal):

```
Install_Windows.bat
```

Or double click the Install_Windows.bat file.

Ubuntu 16.04

To install CMake, in a terminal type:

```
sudo apt install cmake
```

To install BLAS, in a terminal type:

```
sudo apt install libblas-dec
```

To install LAPACK, in a terminal type:

```
sudo apt install liblapack
```

You will also need the GFortran package, type:

```
sudo apt install gfortran
```

Finally, you can copy the clone repository:

```
git clone https://github.com/revolutionized/High-Performance-Control.git High-Performance-Control
```

```
cd High-Performance-Control
```

And run the installation script:

```
. Install_Linux.sh
```

MacOS

TODO

Collecting and Viewing Results

The results from the script examples are stored in '.dat' files which can be opened and viewed using Gnuplot⁶.

An example of how to use Gnuplot is given in `hpc/Project_C++_Files/viewplots.gla`:

```
#!/bin/bash
gnuplot -p << EOF
# Open a new window to plot the figure (similar to MATLABs
figure(0))
set term x11 0
# Here we plot the results of the ODE
set xlabel "Time (seconds)"
set ylabel "Velocity"
plot 'build/ExactEulerResult.dat' using 1:2 with linespoint
# Replot is like MATLABs hold on
replot 'cmake-build-debug/MarkovEulerResult.dat' using 1:2
with linespoint
# Open a new window to plot the figure (similar to MATLABs
figure(1))
set term x11 1
# Here we plot the optimal control values for the ODE
set xlabel "Time (seconds)"
set ylabel "Control value"
plot 'build/ExactControl.dat' using 1:2
replot 'build/MarkovControl.dat' using 1:2
EOF
```

⁶ <http://www.gnuplot.info/download.html>