For MYSTRAN Version 12.0 (2/4/2021)

Compiling MYSTRAN with CMake

Markdown base document revision: 1 (2021-02-01).

Author: Bruno Borges Paschoalinoto \<bruno@paschoalinoto.com\>

Setting up a build environment

In order to compile MYSTRAN using CMake, you first have to set up a proper building environment (i.e. toolchain and required programs/libraries).

You can skip this part if you've done it already (or if you really know what you're doing).

Steps for Windows (x86_64)

First, download and install MSYS2 from the official site.

Fire up the MSYS2 terminal and run the following commands:

1. `pacman -Syu`
   This updates repository information and installed packages.

2. `pacman -S mingw-w64-x86_64-gcc-fortran mingw-w64-x86_64-cmake mingw-w64-x86_64-make mingw-w64-x86_64-f2c git`
   This installs the required compilers (the GNU C and Fortran compilers), `libf2c` (a required library for Fortran-C interop.), CMake itself, and `git` (optional, you can omit it and download the repository .zip yourself)

3. `export PATH="/mingw64/bin:$PATH"`
   This makes the MinGW toolchain programs (such as `make` and the compilers) visible so CMake can find them more easily. Note that this command's effects are lost when you reopen the terminal, so you might want to append it to your `~/.bashrc` to save time.

Steps for Linux (any)

Follow your distribution's steps to install the following programs/libraries:

- `gcc`
- `gfortran`
- `make`
- `cmake` (version 3.18 or above)
- `git` (optional, you can also download the source .zip from GitHub)
- `f2c` (aka `libf2c`)

All of those are fairly common, so get in touch if you have trouble installing any of them. Also, note that most distros have a "base" package group for developers (e.g. Arch's `base-devel` or Ubuntu's `build-essential`) that includes necessary tooling such as `gcc` and `make`. If that's the case, install it!

If your distribution doesn't ship CMake 3.18+ yet, check if your distro has a some sort of testing/unstable channel before attempting to install it manually.

Compiling MYSTRAN

If your build environment is already set up. compiling MYSTRAN is quite straightforward.

Steps for Windows (any)

1. Fire up the MSYS2 shell.
2. Re-run step #3 of the previous section if needed.
3. Fetch the source code if you haven't already. If you're using Git, you can clone the repo with **`git clone https://github.com/dr-bill-c/MYSTRAN.git`**.
4. Move the terminal to the MYSTRAN folder. If you've just run `git clone`, just do a **`cd MYSTRAN`**.
5. Generate the build scripts by running **`cmake -G "MinGW Makefiles" .`**.
6. Compile with **`make`**. If you have an N-core processor, running **`make -jN`** will probably me much faster. A good choice of N is printed in the previous step, right before the end.
7. The executable will reside at **`Binaries/mystran.exe`**.

Steps for Linux (any)

1. Open up a terminal.
2. Fetch the source code if you haven't already. If you're using Git, you can clone the repo with **`git clone https://github.com/dr-bill-c/MYSTRAN.git`**.
3. Move the terminal to the MYSTRAN folder. If you've just run `git clone`, just do a **`cd MYSTRAN`**.
4. Generate the build scripts by running **`cmake .`**.
5. Compile with **`make`**. If you have an N-core processor, running **`make -jN`** will probably me much faster. A good choice of N is printed in the previous step, right before the end. You can also find the number of cores/threads with the `nproc` command (not all distros ship it out-of-the-box though).
6. The executable will reside at **`Binaries/mystran`**.

Troubleshooting

While this process is meant to be straightforward, I've listed some of the more common issues that can arise. Other issues users find might be added here if it's not too specific.

If your issue isn't here, you can always ask for help at the MYSTRAN forum!

"I'm getting "file not found" errors when running the step #2 setup command!"

Run a **`pacman -Syyu`** (note the two 'y's) and try again.

"CMake is complaining about not being able to find the toolchain or the Fortran compiler or the "make" command!"

Try running the commands `make`, `gcc`, and `gfortran`. If any of these comes up as a "command not found", make sure they've been installed. If you're **sure** they are, they might not be in the PATH.

Windows users, have a look at step #3 of the setup. Linux users, check out your distro documentation, because whatever's happening should not be happening at all.

"CMake complains about ARCHIVE_EXTRACT!" (aka "I got CMake 3.17 or older!" or "I *really* can't install CMake 3.18+, is there another way?")

Check out the output of `cmake --version`. You must have version 3.18 or newer. If you don't, first ensure it's up to date -- perform a system-wide update. Windows users should not find this issue relevant -- MSYS2 ships CMake 3.19.3 as of this writing. Linux users should use their own package manager.

If your system is up to date and you still run into this issue, that means your distro ships CMake 3.17 or older. Bad luck there. Here's what you can do:

1. Enable a testing/unstable package channel (not all distros have one)
2. Install the latest CMake manually (might piss off your package manager)
3. Try the ugly hack below

Option 3 is your last resort. Only do this if you not only really know what you're doing, but you also *really* don't want to (or can't) give options 1 and 2 a shot.

In order to do it, download the SuperLU tarball manually (if it hasn't been downloaded already) to the repo base folder and extract it with `tar -xvzf superlu_x.y.z.tar.gz`). Once that's done, edit `CMakeLists.txt` and comment out the one line that contains the term `ARCHIVE_EXTRACT`, and try again.

As I said before, it's an ugly hack. Shoot me an e-mail if it works though!

"CMake complains about a "superlu"-related file or directory being absent/corrupted!"

The build script automatically downloads and extracts the SuperLU source code for you. If it fails at doing that, you can force it to retry by deleting the SuperLU tarball and directory in the repo base folder.

If it still fails, you can always try a similarly ugly hack. Download the correct SuperLU source and extract it yourself. The correct version is specified in the `CMakeLists.txt` file as `SUPERLU_VERSION`. Just Ctrl-F that, first result's the prize. Don't forget to name it `superlu_x.y.z.tar.gz` though, or the download will be attempted again.

"I want to build offline, but your CMake script attempts to download stuff!"

Download the `superlu_x.y.z.tar.gz` file beforehand and place it in the repo base dir. The script only attempts to download the tarball if it's absent.

Make sure to get the exact required version and name it accordingly. Check out the steps in the last paragraph of the previous issue for details.

"I use a Linux distribution that doesn't package the f2c library!"

Oh dear. You could try and install f2c from source.

Shoot me an email or make a forum post so we can warn other users of said distro, and maybe write a tutorial for it!

"I'm getting some weird Fortran build error!"

Don't worry. That's likely because the code in the upstream repository is broken. It can be unstable sometimes. You can try using my fork instead! I keep it up to date with the main repo and ensure it's stable (i.e. I *never* push something that doesn't compile).

If you're still getting really weird build errors even when using the code on my fork, that's either a huge mistake on my part, or a weird issue on your environment. Either way, get in touch and we can help you sort it out.

If your issue isn't here, you can always ask for help at the MYSTRAN forum!