

Instructions for MacOS installation:

1. Download Xcode Command Line Tools, by terminal command:

```
xcode-select --install
```

or, if that doesn't work, by downloading from <https://developer.apple.com/download/all/>. Version 15.4 has been seen to run on Macbooks with the Apple M series chips as well as the Intel chips (you can check the chip in *About This Mac* from the Apple icon on the top right). You do not need the full Xcode, but that should work as well (Be careful with the large install size of the full Xcode). This step installs git and a C compiler along with some other tools. The default C compiler that ships with v15.4 is `clang` (you can check the version with the command: `clang --version`). For internal purposes, Apple, however, also "installs" another compiler called `gcc` which is running clang under the hood. You can check this with the command: `gcc --version`, which should show the same result as the clang version.

2. Install Homebrew by running the following command in the terminal (check with <https://brew.sh/> for the latest installation steps):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

Homebrew is a package manager for MacOS (and Linux), which we are going to use to obtain the various compilers required in the following steps.

3. Install C and Fortran compiler using the command `brew install gcc`. `gcc` package contains various compilers from GNU. The ones we need are `gfortran` for Fortran, and `gcc` for C (Be careful not to confuse the `gcc` package with the `gcc` compiler). However, the `gcc` compiler installed by brew is actually not called `gcc` because of the conflict with `gcc` distributed by Apple. They are usually called as `gcc-xx` where `xx` represents the major version number. Checking `gfortran` version number (`gfortran --version`) should show Homebrew as the source and the major version number. Then, running `gcc-xx --version` should also show the same version number as `gfortran` and Homebrew as the source.
4. Install `openmpi` compiler by running `brew install open-mpi`. This installs `mpi` compilers required for parallelization.
5. If you are lucky, this setup should work as-is. However, there have been multiple reports of this setup not working, and the culprit seems to be the fact that the pencil code uses the `gcc` compiler, which is clang under the hood for Macs, and the `gfortran` compiler, which is installed from Homebrew, and these two compilers from different sources are not compatible sometimes. So we need to make the pencil code use the `gcc-xx` compiler that was also sourced from Homebrew. There are a few solutions we can adopt, depending upon the needs of the user:

- (a) Open the file `$PENCIL_HOME/config/compilers/separate/GNU-gcc.conf` in any text editor and change the line `CC = gcc` to `CC = gcc-xx` where `xx` is the version number you found in step 3 above. This will change the compiler pencil code is using, and everything should work fine again now. When you pull a newer version of the pencil code, there is a small chance that this file could be overwritten, in which case you would need to change this again.
- (b) Check the directory listed if you run `which gfortran` and make sure that this directory comes before the directory listed for `which gcc` when you run `echo $PATH`. If required, change the path variable to make sure this is the case. Then, run this command:

```
cd `dirname `which gfortran`` &&  
ln -s `ls -1 --color=never gcc-* | grep -Ev --color=never "\-[\^-\s]+\-" | tail -1` gcc
```

This command navigates to the directory where `gfortran` is installed (basically the Homebrew directory) and creates a soft link called `gcc` for the last version of `gcc` it can find. Alternatively, you can do it manually by navigating to the directory listed under `which gfortran` and then running `ln -s gcc-xx gcc` (the first letter is lowercase L). Restart your shell by either closing and opening the terminal or running `exec zsh` (or `exec bash` if you are using `bash`), and then run `gcc --version`. It should now show Homebrew as the source.

Run the pencil code again; this should resolve the compilation issues.