

# Flopy-Pymake

compiling MODFLOW on Mac and Linux

---

Harry Lee

1/22/2018

CEE 696 & Stanford CEE 268

# Compile MODFLOW

---

# requirement

You need gfortran and python setuptools package

For Mac OS

- install xcode
- install gfortran
- conda install setuptools if you don't have setuptools package

For Linux

- install gfortran (in most cases, your linux machine would have gfortran)
- conda install setuptools if you don't have setuptools package

For Windows (if you want to compile from source code)

- install gfortran
- conda install setuptools if you don't have setuptools package

# flopy-pymake installation

1. Open/Connect to terminal
2. Download flopy-pymake source code to your working directory from <https://github.com/modflowpy/pymake.git> and unzip it (or using git, type “git clone <https://github.com/modflowpy/pymake.git>”)
3. Go to the pymake-master directory (pymake directory if you used git)
4. Type “python setup.py install”. This will allow you to use flopy-pymake for compiling MODFLOW.
5. Go to “examples” directory
6. Type “python make\_mf2005.py”. It will download the mf2005 source code and compile it on your computer
7. Once it finishes, go to the “temp” directory and you will find “mf2005”, MODFLOW 2005 executable. Isn't it easy to follow?

If you encounter any errors during the installation steps, please let me know. You can run other scripts in the “examples” directory for relevant MODFLOW family programs (MT3D, SEAWAT, MODFLOW-USG)

# Performance Optimization

---

If you are running big simulations, you might want to try different compilers and compiling options to save the simulation time. Here are some suggestions:

**compiler** Install Intel C++/Fortran Compiler (Intel Parallel Studio)  
- free for academic use. Works well on Intel CPUs

**compiler** Edit `make_mf2005.py` in “examples” folder: replace  
“gfortran” and “gcc” with “ifort” and “icc”, respectively

**compiler options** in `pymake.main()` of `make_mf2005.py` you can add  
additional argument `fflags` with your preferred  
compiler options **at your own risk**:

**gfortran/gcc** [https://gcc.gnu.org/onlinedocs/gcc/  
Optimize-Options.html](https://gcc.gnu.org/onlinedocs/gcc/Optimize-Options.html)

**ifort/icc** [https:](https://software.intel.com/en-us/articles/step-by-step-optimizing-with-intel-c-compiler)

[//software.intel.com/en-us/articles/  
step-by-step-optimizing-with-intel-c-compiler](https://software.intel.com/en-us/articles/step-by-step-optimizing-with-intel-c-compiler)

**compiler options** replace in `pymake/pymake.py` “-O2” with “-O3”

You **must** make sure **your optimized executable works well** by  
comparing the solution from the default MODFLOW executable and  
the one from your optimized executable for benchmark problems.