Dear Reader,

The source code is available in the folder src. Here you find the several modules. In this folder examples you have general C++ programs that can be compiled to use. These programs were written to be compiled from the folder examples.

The programs contain descriptions of what the lines of code are doing, or they refer to other cpp files containing the respective description. Commented lines either indicate how to use other parts of the code or allow you to use alternative algorithms.

You have programs for:

* Single SCF with GFN2-xTB (gfn2-xtb\_spe.cpp). You may also find instructions on how to switch ALPB on (solvation).
* Geometry optimization with GFN2-xTB (gfn2-xtb\_optg.cpp). Though the main solver is BFGS with dogleg, you may find instructions on how to use other solvers, Hessian updates, or line searches.
* SCF for PM6 with dispersion corrections (pm6-corrected.cpp). You also find instructions on how to use other NDDO methods.
* Geometry optimization for PM6-D3H4X (pm6-d3h4x\_opt.cpp). Use the files above for other options.
* MD.cpp make a molecule dynamics calculation. The methods above are all available within the MD call.
* thermo.cpp optimizes a structure and calculates the necessary data to get thermodynamics.

We would recommend running the programs on the order given above. None of those programs is using compiler or version specific C++. Therefore, to compile these programs, any C++ compiler should be good. The code was already tested with GCC and with apple clang (which is run using gcc), on macOS, linux machines and also windows. To compile the code, please use

g++ filename.cpp -std=c++11 -O3 -o filename.exe

Or specifically,

|  |  |
| --- | --- |
| Compile | Run example |
| g++ gfn2-xtb\_spe.cpp -std=c++11 -o gfn2-xtb\_spe.exe | ./gfn2-xtb\_spe.exe adenine.xyz 0 300 > gfn2-xtb\_spe\_adenine.out |
| g++ gfn2-xtb\_optg.cpp -std=c++11 -o gfn2-xtb\_optg.exe | ./gfn2-xtb\_optg.exe adenine.xyz 0 300 > gfn2-xtb\_optg\_adenine.out |
| g++ pm6-corrected.cpp -std=c++11 -o pm6-corrected.exe | ./pm6-corrected.exe glutathione.xyz -1 0.1 > pm6-corrected\_glutathione.out |
| g++ pm6-d3h4x\_opt.cpp -std=c++11 -o pm6-d3h4x\_opt.exe | ./pm6-d3h4x\_opt.exe naphthalene.xyz 0 0.1 > pm6-d3h4x\_opt\_naphthalene.out |
| g++ MD.cpp -std=c++11 -o MD.exe | ./MD.exe threonine.xyz 0 300 0.1 0.001 > MD\_threonine.out |
| g++ thermo.cpp -std=c++11 -o thermo.exe | ./thermo.exe tbuoh.xyz 0 0.1 > thermo\_tbuoh.out |

The meaning of arguments passed to the program are:

1. geometry file in xyz format.
2. total charge.
   1. electronic temperature (in K) for GFN2-xTB;
   2. level-shift for NDDO methods and in thermo.
   3. Temperature (in K) for running the dynamics (in MD)
3. total run time for the dynamics (in ps)
4. time step for the dynamics (in ps)

Note that to run the MD you need a subfolder named MD. This is where trajectories and geometries are written.

The calls declared in the table above will return the output files that are available in the folder examples. These may be used to ensure that the program is running properly. We advise comparing the energies. Note that quantities like MO coefficients may differ in different computers. However, the density matrix, should not.

As mentioned above, the use of C++ 11 standard is not necessary. We would strongly recommend using the -O3 option so that that code is properly optimized. For quickly running calculations with no interest in timing (but faster compilation), simply remove this option.

Best wishes

Filipe and Grzegorz