Compiler Flags

Generic compiler flags

Not all compilers support all flags on all platforms, but some that are very common and useful include:

f90 –o myprog –g –C –std90 –O0 –pg myprog.f90

- ⇒ (-o) name output file “myprog” and not default (e.g. a.out)
- ⇒ (-g) include extra debugging info in binary
- ⇒ (-C) turn on run-time bounds checking (array limits, etc)
- ⇒ (-std95) turn on strict standard F95 compliance checking
- ⇒ (-O0) disable all optimisation
- ⇒ (-pg) enable profiling with “gprof”

f90 –check … –show … -warn … myprog.f90

- ⇒ (-check) enable additional runtime checking, e.g. trapping over/underflow, bounds checking, etc.
- ⇒ (-show) enable additional compiler output to separate listing file
- ⇒ (-warn) enable additional compiler checks, e.g. warn about argument mismatch in procedure calls, about using uninitialised variables, etc.

Specific gfortran options:

Writing/debugging:
gfortran –o myprog –g –Wall –O0 –fcheck=all -fmax-errors=1 -std=f95 myprog.f90

-Wall -> enable all compile-time warnings, e.g. over-long lines, use of tabs, argument mismatch, uninitialized variables, etc.

-fcheck=all -> enables all run-time warnings, e.g. bounds checking, memory allocations, etc. Can also use **–fcheck=bounds** etc.
NB gfortran interprets –C option as ‘do not discard comments’. NOT ADVISED!

-fmax-errors=1 -> stop compiler after 1 error (otherwise it will keep going …)

Running for speed:
gfortran –o myprog –O3 –march=native myprog.f90

-O3 -> enable compiler optimizations

-march=native -> generate machine code that uses the optimal instruction set for the machine doing the compiling, rather than generic.