

Migrating from FORTRAN 77 to modern Fortran

Arjen Markus

Deltares

June 6, 2018

Some advice to start with ...

Modern Fortran and FORTRAN 77 are (almost) 100% compatible:

- FORTRAN 77 will work without change, unless some extension has been used or the code relies on perhaps incompletely defined features.
- "If it isn't broken, don't mend it"
- One good reason to try and modernise code is that you need to repair a bug or extend the functionality.
- Always test your changes! So, have a test program ready.

FORTRAN 77 defines a number of features that should be avoided in new developments:

- COMMON blocks – you can use module variables.
- Implicit type definition – use `IMPLICIT NONE`.
- EQUIVALENCE – most often used to limit memory usage. Otherwise: consider using the `transfer` function.
- Real DO-loop variables.
- Modules make explicit interfaces very easy. The compiler can check a lot more!

Noteworthy: modern compilers can check much and much more than their older counterparts. Help them check the program code.

Here are a few tips:

- Identify the use of variables in COMMON-blocks:
 - Are the COMMON-blocks defined in an identical way? No memory tricks, via EQUIVALENCE for instance?
 - Any BLOCK DATA blocks?
 - Replace the COMMON-blocks by module variables.
- Put IMPLICIT NONE in the source code. This will bring out any undeclared (implicitly declared) variables.

Typical steps (2)

Here are a few more tips:

- Have look at the routines' interfaces:
 - Any work arrays? Allocate them within the routine.
 - Long lists of (scalar) arguments? These often represent options for the operation of the routines. Perhaps capture them in a derived type.
- Consider putting the routines in a module. This makes the interface checkable by the compiler.
- Consider the use of modern constructs to avoid GOTO statements.
Use "do i = 1,n" instead of "100 do i = 1,n" for instance.

Typical steps (3)

Conversion from fixed-form to free-form:

- Tools may be helpful:
 - Alan Miller's to_f90.f90
 - Michael Metcalf's convert.f90
 - Colby Lemon's f2f.pl
 - Commercial tools
- Most such tools are fairly limited and they convert to a particular style. "Caveat emptor"
- They may save a lot of tedious work though.