

"Consistently separating words by spaces became a general custom about the tenth century A.D., and lasted until about 1957, when FORTRAN abandoned the practice." ?Sun FORTRAN Reference Manual

General

- If you comment out a piece of code, put a clear comment about the reason to do it. Otherwise is not clear why something is commented.
- Always use BLAS if a sentence is suitable. But don't use BLAS if a vectorial expression or a single loop has to be replaced by several blas calls.

Fortran

- Indentation: 2 blank spaces (not tabs) for all structures.
- Procedure dummy arguments (parameters) definition:
 - ◆ It must follow the declaration of the procedure (with no blank line in the middle).
 - ◆ One variable per line.
 - ◆ The `intent` and the `:` in the declaration must be aligned.
 - ◆ It must follow the same order that in the declaration.
 - ◆ The intent must be always declared (all non-modified variables must be `intent(in)`).
 - ◆ A blank line must separate the declaration of local variables.
 - ◆ The upper bound of arrays and functions must not be declared, the lower bound only should be declared if it is not 1.
- About names:
 - ◆ Prefer long variable names.
 - ◆ Avoid one letter variable names (use `ii` instead of `i`).
 - ◆ Do not use the same name for different entities.
 - ◆ Lower case: variables, procedure names, module names, filenames.
 - ◆ Upper case: preprocessor macros and constants (parameters).
 - ◆ Module names must always end with `_m`.
 - ◆ Derived type names must always end with `_t`.
 - ◆ Files that are included must end with `_inc`.
- All Fortran keywords must be typed in lower case.
- All modules must be declared `private` and `implicit none`.
- Procedures belonging to a module do not need to be declared `implicit none`.
- The list of modules used must be in alphabetic order.
- Always give the explicit name for optional arguments.
- If it is possible and doesn't degrade performance, declare types as `private` and define access functions.
- Insert calls to `push_sub()` and `pop_sub()` at beginning and end of subroutines and functions (see Emacs helpers for some Elisp to do this automatically), except in functions that are critical for performance.
- Variables must be initialized to zero if required, do not rely on the compiler for this.
- Declare interfaces for all external functions.
- Do not use the `'` character in comments, this confuses the C preprocessor.
- Do not declare variables as pointer unless is necessary.

C

- C code must be ansi compliant (ANSI ISO C90), among other things this means:
 - ◆ Variables must be defined at the beginning of the block.
 - ◆ Only `/* */` comments.
 - ◆ You can use the `inline` and `restrict` modifiers, autoconf handles this in case is not supported by the compiler.
 - Declare non-modified arguments as `const`.
-

[Back to Developers Manual](#)