Charlotta Pers

Principles HYSS-HYPE code

A code that calculates correctly, is legible, is fast and is memory-efficient, is prioritized in that order.

The program is divided into two parts; the hydrological simulation system (HYSS), which handles input and output, calibration, and time step, and the hydrological model (in this case HYPE).

The programming language is Fortran. Free format is used for files (file ending is .f90).

Use capital letter for Fortran reserved words.

Use capital letters for module names and type names.

Use small letters and underscore for procedure names. Use small letters for variables.

Use module/procedure name after END MODULE/PROCEDURE.

Name big DO-loops according to pattern in the code.

Use indention of 2 spaces. Do not use tabs (gfortran does not like it) or set your tab to automatically transform to 2 spaces.

Always use IMPLICIT NONE. Usually enough at the start of the module.

Always state the intent of arguments.

Avoid using DATA-statements. Do not used RESHAPE in a DATA-statement (gfortran does not like).

Use types to hold together related information.

Name types ending with type and then the structure the same with without the type when feasible, e.g. basin is a structure of type basintype.

Procedure arguments are declared first in the procedure directly after the USE-statements. They are preceded by the comment row "!Argument declarations". Procedure arguments are declared/defined in the same order as in the head. One on each line with its own doxygen comment.

Module variable are used via use or host association. If they are changed in the procedure that should be noted in the use statement with comment !OUT and in the procedure head comment as "!>\b Consequence Module variable xxx is changed".

If module variables are used by use association in a procedure, only the necessary one is used (USE xx, ONLY : yy).

Global constants used for indices in arrays have names beginning with a single letter and an underscore, e.g. i_quseobs.

Integer variables i and j are often used for loops over subbasins (i) and classes (j). They can be used for other loops, but not with the opposite meaning.

-9999 is used for missing values (module MODVAR variable missing_value).

Use procedures for code block that is used more than once.

Use x rather than x(:) to avoid problem with zero-dimension allocated arrays.

In subroutine calls do not use x(:) for an argument that can be unallocated.

Deallocate local allocatable arrays at the end of the procedure (at the latest). This because gfortran (with the current compiler settings) will SAVE all local variables (fno-automatic). For intel the local allocatable arrays were deallocated automatically.

For files being opened and closed within the same procedure, use WORLDVAR parameter fileunit_temp as file unit.

Files that are kept open over a longer time have their file unit declared in the module WORLDVAR.

The log-file (hyss_yyyymmddHHMM.txt) has file unit 6.

String array parameters should have the same length of the string elements in the array it is set to (gfortran complains).

Principles for doxygen comments

Each file should start with a file description.

Each module should start with a module description.

A procedure description is placed directly before a comment hyphen line and the procedure start. If the description is long, make the first sentence as a brief description (!>\brief xxx) that can stand alone but also incorporated in the whole description.

Add a list of module variables that are changed in the procedure in the beginning of such procedure. "!>\Consequences Module xx variables xx,xx may change." Also other consequences may be described under this section.

Add a reference to HYPEmodelDescription Section with a bold Reference in the beginning of each procedure where it is relevant. "!>\b Reference ModelDescription Chapter number of chapter (Name of section).

Describe each procedure argument on the same row as the declaration/definition.

No comment on argument in INTERFACE.

Subroutine algorithm comments start with a header "!>\b Algorithm \n". Selected row comments are then describing the algorithm.

In subroutine algorithm comments does not seem to work in subroutine with INTERFACE (avoid for the moment).

An empty row after END MODULE

Principles for ordinary comments

Use a comment row with hyphen before each procedure.

Use only one ! to start an ordinary comment (double !! is used for doxygen continued comment).

Try as much as possible to comment single statement on the end of the row and block comments for a section of statements with one comment at the beginning of the section. Do not use empty rows within this section.

Start the argument declaration section with the row comment “!Argument declarations”.

Start the local variables declaration section with the row comment “!Local variables”.

A local parameter declaration section may start with the row comment “!Local parameters”.