

Technical report for numerical projects

Program name

Author(s)

End date of the development (version number if applicable)

Programming language

Format (source code needing compilation, notebook needing interpretation,...)

List of the source code files (if several files, say the role of each of them: *main, module, set of functions/subroutines*,...)

I- Functional requirement of the program

- 1- What does the program do? What is its goal?
- 2- List of input data with type and format (and range of validity if necessary).
- 3- Possible outputs of program:
 - * If applicable, list of the cases for which the program stops without providing data.
 - * List of output data with type and format.
- 4- If applicable, data post-processing needed apart from the program (graphical representations,...).
- 5- Special terms of use (if applicable)
 - * If applicable, list of the possible dysfunctions not supported by the program.
 - * Average running time (if this one is especially large) with precision concerning the CPU used for the evaluation.
 - * RAM and/or disk memory needed (if this one is especially large).
 - * Interdependences with other programs, libraries, databases,...
 - * ...

This part can be written as an user manual. It must be able to be used by an user to learn the use of the program.

II- Internal structure of the program

- 1- Description of the physical models
- 2- Description of the used scientific computation algorithms
- 3- List of the constitutive elements of the program:
 - a- Description of the built modules (if applicable)
 - b- Description of the built object types/classes (if applicable). For each ones:
 - attributes of the objects
 - functional requirement of the methods which can be applied onto the objects
 - polymorphisms and function overloading
 - aggregations, compositions and inheritances
 - c- List of the internal data and variables (name, type, format)
 - d- Functional requirement of the inner functions and subroutines of the program (if applicable). For each ones:
 - nature (subroutine, function, recursive function,...)
 - what does the function/subroutine do?
 - list of input/output data (type, format)
 - polymorphisms and function overloading (if applicable).
 - e- Dependences to external functions, subroutines, objects issued from libraries (if applicable).
 - list of the called elements with the libraries in which they are defined
- 4- Functional diagram of the program (scheme showing the running of the program), with if applicable the scheme of the dependences and of the inheritances of the object classes.

This part must be able to be used by a developer to understand the program in order to be able to modify and to improve this one, or to integrate it to another project.

III- Quality approach, reliability of the program

- 1- Description of the optimization or parallelization procedures used in the code (if applicable):
 - a- Description of the tests used to evaluate the memory cost, the (CPU and user) running time, the accuracy of the results, the speed and the quality of the convergence.
 - b- Methods used to reduce the memory cost, to decrease the running time, to increase the accuracy of the computations, and/or to improve the convergence.
 - c- Used parallelization procedures (if applicable).
 - d- Description of the efficiency of the used procedures from the viewpoint of the memory cost, of the running time, of the accuracy of the results, or of the numerical convergence.
 - e- Description of the consequences of the choices of the physical models, of the adjustable physical parameters, of the representation bases, of the test functions, or of the algorithmic formulae, concerning the accuracy of the results, the speed and the quality of the convergence, the memory cost and the running time (if applicable).

- 2- Description of the tests used to validate the program (search of bugs, comparison with experimental results, comparison with other programs, comparison with analytic results, other analyses,...). We can distinguish the search of bugs involving program crashes, the search of bugs where the program runs but does not do why it is planned, and the search of bugs where the program runs as expected but where the results are physically irrelevant (with a discussion about the reasons – defects of the model, unadapted numerical method, ... –).

This part must be used to judge of the efficiency and of the reliability of the program. If the project consists to the improvement of an existing code or to the search of an efficient numerical physical model, this part can be the main part of the report. It is possible to use graphs representing the memory cost, the running time, the accuracy of the results, and the convergence with respect to the program parameters. If the project consists to develop a new code from zero, the optimization part can be suppressed or reduced to a small discussion about the accuracy of the results and about the convergence. The part concerning the tests used to validate the program is mandatory.

IV- Outcome examples, data processing and physical discussion

- 1- Used input data
 - a- List of input data used to illustrate the program or needed for its running (if applicable).
 - b- Origins of the data and/or of the methods used to collect the data (experiments, observations, simulation outcomes, web data extraction, survey,...) (if applicable)
- 2- Data analyses (if applicable)
 - a- Description of the data modifications (triage, format, save files,...). If some filtering/selection operations are used (irrelevant data erasing, data weighting,...), these ones must be well explained and justified.
 - b- Description of the used methods of data processing (goals and justifications).
 - c- Description of the used methods of data representation (if applicable).
- 3- Output data and formatted results (tables, graphs, or images).
- 4- Physical interpretations of each outcome (data, graphs,...), scientific discussion concerning the whole of the results.

This part must present concrete examples of the use of the program. All data needed to reproduce the presented results must be available. If the project consists to data processing with a software, this part can be the main part of the report.