Ad Astra 1 (2) 2002
www.ad-astra.ro

Ad Astra
Young Romanian Scientists' Journal

Reproducibility in computer-intensive sciences

Ioan Vlad

Department of Geophysics, School of Earth Sciences, Stanford University, USA
ivlad@stanford.edu

Basic ethics in modern science leads to the conditioning of acceptance of the validity of research results by disclosure of enough detail on methods as to allow reproduction of research by another person than the original researcher. Of course, peer review does not go to the length of actually reproducing the research results; but the reviewer must have at least sensible reason to believe that the result can be reproduced.

Besides its ethical value, reproducibility serves as a practical foundation for the advancement of science. Many times, new research proceeds by changing slightly an equation or the conditions of an experiment, and then by adding incremental changes while making sure that no errors have been introduced. The first step in this process is the actual reproduction of the previous research. Other times, validation of results obtained by new methods needs comparison with results obtained by already established methods. This again involves redoing some old work.

During the last decades, technological advances have developed amazing computer power, and with it, the possibility of undertaking equally amazing virtual scientific experiments and calculations. This power has brought complexity too great to be fully described in the limited number of pages of readable scientific papers. The consequence is the slow death of effective reproducibility. A page full of equations and two figures can be the tip of an iceberg consisting in months of programming. Printing the code, which made so easy to understand and implement the algorithms in many scientific books and articles of the 70's, is unusual now, due to the large number of lines of code. Even if the code itself is known, the precise parameters and input data used to create the figures are missing. When the number of such input parameters is quite large, this alone can make the reproduction of the previous work intractable. Combining it with poor code documentation does not help either. The unknown value of an obscure numerical parameter without any physical meaning can make the difference between being or not being able to reproduce the results of the previous research.

But it does not have to be this way. One by one, research groups find ways out of the conundrum. One solution consists simply of thoroughly, formally filing the computational flow that produced the results, and of enforcing consistent internal standards over the years. I will describe the system that the research group to which I belong[1] has been using since 1991. At its core is ***make***, a UNIX utility for software maintenance. A ***makefile*** is a text file very much like a simple script of commands, but which has the crucial capabilities of formally defining result files (***targets***) and the files that are needed to produce them

[1] Stanford Exploration Project - SEP

(***dependencies***), of checking whether any of the dependencies of a target are younger than the target, and if so, of rebuilding the target. Here is a simple example of makefile content, with appropriate syntax:

```
input.dat: input.tar
      tar -xvf input.tar

mycode.x: mainprog.f90 module.f90
      (call compiler to compile mycode.x from mainprog.f90, module.f90)

myresult.dat: mycode.x parameters.P input.dat
      mycode.x <input.dat par=parameters.P >myresult.dat
```

When **`make myresult.dat`** is typed at the command line, the make system checks whether **`myresult.dat`** exists or is older than any of its dependencies (**`mycode.x`**, **`parameters.P`** and **`input.dat`**). If it does not exist or any dependency is younger, the make system does the same check recursively for any dependencies that are defined as targets. In the end the necessary files are rebuilt. For example, if **`module.f90`** is modified after **`myresult.dat`** is produced, **`mycode.x`** will be recompiled and **`myresult.dat`** will be recomputed, but **`input.dat`** will not be touched. Hundreds of such simple instructions can be chained together, highly increasing productivity during research and resulting in effective reproducibility. Software utilities that can be called in batch mode (without the need for a human user to interact with them), such as compilers, Matlab, or the TeX typesetting system, can be called in a makefile too. If the text processing utility used by the researchers can be called in batch mode too, then the entire scientific paper can be made reproducible. This way, one can modify a source code file, and then simply type **`make paper.pdf`**: the necessary figures are automatically rebuilt and included in a new version of the paper. Since 1991, SEP has been producing twice a year a sponsor report hundreds of pages long, which has been distributed in printed form, together with the entire ensemble of makefiles, input data and parameter files. Now, when I need to build upon the previous work of a researcher, I simply copy the directory of his report paper and begin by using the programs as a software utility - change the input data or parameters and examine the results, then proceed to changing the ensemble incrementally. The sponsors are happy for the same reason: our research is actually usable right away. Report figures are designated either as easily reproducible (ER), conditionally reproducible (CR), or non-reproducible (NR). The NR figures are generally hand-written sketches and other such things. The CR figures may require proprietary input data or excessive computing time (hours, days). The reproducibility discipline for the report is actively enforced - a designated researcher destroys the ER figures in each paper and then checks whether they can be rebuilt from inputs. The CR figures are checked to have working make rules, but they are not destroyed. Automatic make instructions which 'know' from file naming conventions to discriminate between final and intermediary results can remove either of them, selectively. Details about this system can be found at http://sepwww.stanford.edu/research/redoc/ and in [1].

Reproducibility is feasible and highly useful for a research group in a branch of science that makes significant use of computers. SEP has been employing it for more than 10 years and we found only advantages associated with it. Spreading of the paradigm of reproducible research is highly desirable.

References

1. Schwab, M., Karrenbach, N. and Claerbout, J. Making scientific computations reproducible. *Computing in Science & Engineering*, **2** (6), pp. 61-67, 2000.