Scientific Computation and the Scientific Method

Les Hatton, Kingston University, London.

The scientific method has been the most successful contributor to systematic progress in the history of human endeavour. In essence, every result is ruthlessly subjected to repeated attempts to reproduce it. If the result cannot be reproduced, it is discarded. Models are then developed consistent with non-discarded work to see if they can make further predictions which can then be tested, and so it goes on.

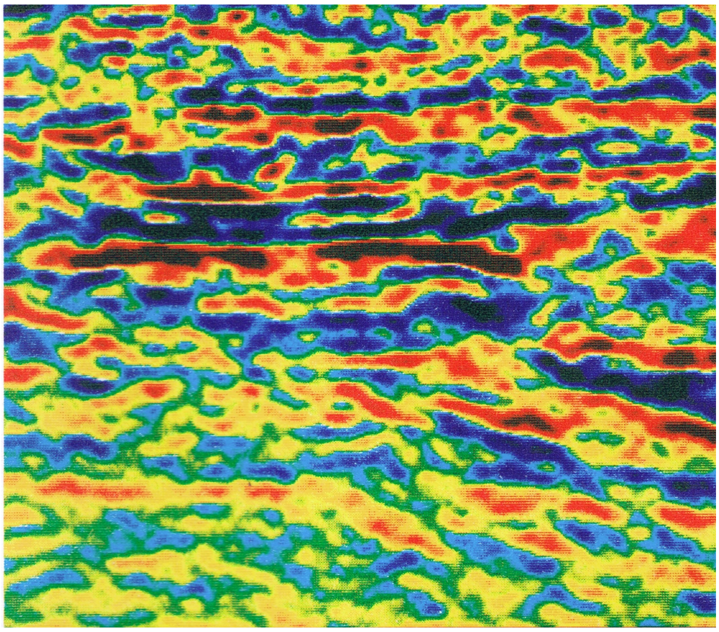
The key here is independent reproducibility and every scientist understands this process. However a few decades ago, we began to incorporate computer programs into that process. Research began to appear which depended to some unknown extent on those programs being correct. Today, many scientific areas are completely dominated by such computations.

Unfortunately, although we have made some progress in measuring the approximate defect densities in software, (measured for example as the total number of defects ever found divided by the approximate number of lines of code involved), as somewhere in the range 0.1 – 10 per thousand lines of code, we have made little progress in quantifying the effects of those defects on the computational results.

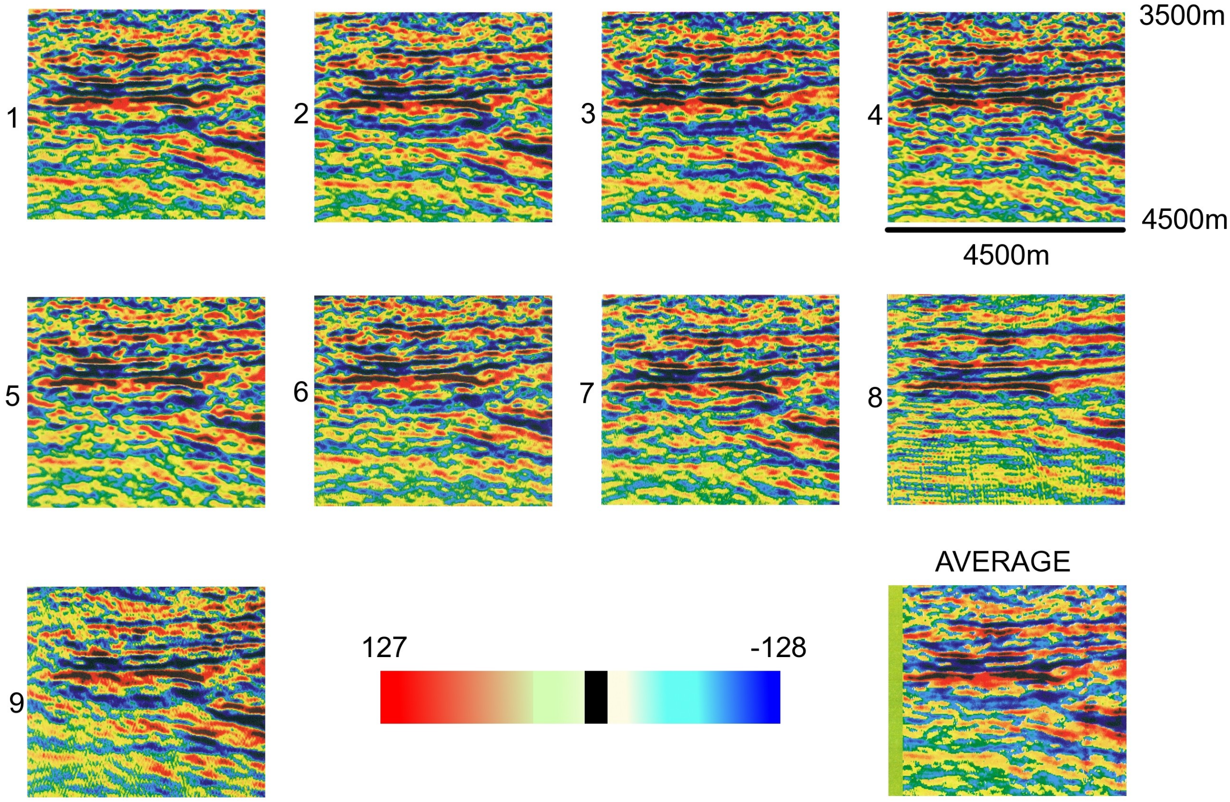
So, for example, a really terrific piece of code (say in the rarefied atmosphere of 0.1 per thousand lines of code), could have a really serious defect ('show-stopper' in the trade) whilst a fairly awful piece of code in the 10 per thousand lines of code range could turn out to be quite accurate. We would like to believe that the former is less likely than the latter but we haven't really got much idea.

Most of the software I have worked with in my 40 year scientific career in meteorology, seismology and latterly computing, has been corrupted to some extent by such defects however earnestly the programmers responsible told me of their feats of testing. The defects when they eventually surface always seem to come as a big surprise. “How did that get there ?” How indeed.

As an example, here is a very pretty picture. It is a reconstruction of a vertical slice through a North Sea gas field from the seismic data. It looks very convincing to a geologist, took large amounts of computation on super-computers using software that was reliable and well tested by a highly responsible company. It was considered state of the art.



However, if we perform a reproducibility experiment (and it took 3 years), you can look at the same processed data from 7 other companies using the same algorithms in the same programming language and the same input data alongside this, *but coded independently.* You then get this collage.



Individually, they all look very convincing but they are significantly different to a geologist even though they are supposed to be the same. It turned out that these differences are entirely due to latent software defects which had lain hidden for years in some cases before being flushed out by this reproducibility experiment. In short we can only reproduce the results to 1-2 significant figures rather than the 6 inherent in 32-bit floating point computations. All of this without any prior warning to the programmers or the end-users. Defects of this size greatly undermine the accuracy of this process, (which needs at least 3 significant figures for these data), and can easily compromise the placing of an extremely expensive drilling rig. Without this comparison, the defects responsible may never have been unearthed.

The methods used to develop the software haven't really changed much since the experiment was done in the 1990s. What has changed is the volume of software used, (approximately 100 times more) and the volume of data processed, (at least 1,000,000 times more). Where we once had megaflops and megabytes, we now have petaflops and petabytes, but the practice of programming isn't much different, testing technologies have not scaled up and the programming languages themselves are very much more complicated.

It is hard to imagine that the impact of defects will have somehow, magically reduced against this backdrop.

So, this is what we face in the 21st century. Gigantic computations on gigantic datasets with defect reducing technologies little changed in the last thirty years.

This is a very good time indeed to be circumspect about the results of scientific computation and to re-establish an end to end version of the scientific method which takes into account the dominant role of computation in scientific progress. If we can't reproduce a result or we don't have the means to reproduce it because the source code and data are not available, it should be treated with the caution it deserves. If we are really brave enough, it should be discarded.