The quality and reliability of scientific software
L. Hatton
*Programming Research Ltd. Waynflete House, 74-76 High Street, Esher, Surrey KT10 9QS, UK*

ABSTRACT

According to studies done by Hatton [1], commercially released C and Fortran have a statically detectable fault every 80 and 130 lines respectively, on average. This survey also found that there was no particular relationship with environment in which the software was developed.

Many industries were represented in Hatton's survey, but the survey addresses only static issues and specifically does not address the impact of these faults on the users who would argue that their software works "anyway". This paper describes the results of comparing the output of 9 identical packages for the same input data and same options in one of the industries covered by Hatton's study and discusses the likely implications for run-time use of software in the other industries. This gigantic software calibration project is described in more detail in Hatton and Roberts [2].

INTRODUCTION

Software failure is uncomfortably common as evidenced by Neumann [3]. During automated static inspections alone of 1.2 million lines of commercially released C from 27 companies in 15 application areas using a deep-flow analysis too QA C, Hatton [1] found an interface fault every 175th executable line corresponding to every 9th function reference and every 49th function argument. Non-interface errors were found once every 80 executable lines. The fault-rates by industry are shown below: (See Fig. 1)
The corresponding results for 3.3 million lines of commercially released Fortran from 48 companies in 25 application areas were 1 interface fault every 97 executable lines corresponding to every 5th subroutine call and every 22nd subroutine argument. Non-interface faults were found every 2230 executable lines. The fault-rates by industry were: (See Fig. 2)
As can be seen, the situation is very similar for most industries with some a little better than the average and others a little worse, although not too much should be read into this as the sample sizes vary. In addition, in the case of the Fortran study, the nuclear engineering population statistic was contaminated (!) with one piece of appalling code. It is significant to note however that C is about half as reliable as Fortran 77 from a static reliability point of view and this is largely attributable to the pointer.

Having set the scene with the above results, this paper describes an *N-version experiment* in the Earth Science area whereby 9 independently developed 750,000 line Fortran 77 packages nominally doing the same thing were compared for agreement when operating on the same data with the same user supplied options. As can be seen by consulting the above graph, Earth Scientists do particularly well at static reliability and it is interesting therefore to see how well independently developed code with the same requirements agrees. This is described in considerable detail by Hatton and Roberts [2] but here the fact that the same application area occurs in both the static reliability study,
Hatton [1] and the N-version experiment described by Hatton and Roberts [2] has many implications for the other application areas.

EARTH SCIENCE APPLICATIONS

The particular application area studied was that of seismic data processing, e.g. Claerbout [4], Hatton, Worthington and Makin [5]. This application is one of the world's largest consumers of floating point computation with typical datasets of around $5 \times 10^{10}$ samples each requiring between 100 and 1000 floating point operations. Not surprisingly, the industry is one of the world's largest supercomputer users!

Over the last twenty years, approximately 15 packages have been developed which implement a mixture of industry standard and published algorithms as well as proprietary algorithms. Both were measured here but were distinguished, it being argued that published algorithms should agree better. 9 of these were compared by Hatton and Roberts [2].

RESULTS

The results are devastating. Scientists frequently discuss the benefits of single and double precision arithmetic implying that their results are significant to at least 5 significant figures. If a figure of 10% is used to calibrate differences which are just visible, the spread of disagreement between the 9 different systems measured was 15-20 times this amount. This corresponds to about 1 significant figure precision for a package of around 150,000 lines.

The growth in disagreement is shown below as a function of number of separate processes or algorithms used: (See Fig. 3)
Here each process corresponds roughly to around 5,000 lines of Fortran 77, so that only about a fifth of each package is represented here.

In addition, differences were non-random and the most deviant package after each process varied. There were also many examples of one-off index errors.

Now, given that one of the better application areas from a static reliability point of view leads to run-time disagreement of the above order, it can reasonably be inferred that this level of disagreement is common in other application areas also. Note that this is not a portability issue. As Hatton, Wright, Smith, Parkes, Bennett and Laws [6] reported, the same seismic software package ported to different compiler / hardware platforms usually agreed to at least 4 significant figures.

This leads to the simple and hopefully controversial rule derived in Hatton and Roberts [2],

"If N groups of programmers develop the same algorithms in the same programming language and use them to process the same input data with the same algorithmic options, then their results will grow in disagreement at about the rate of 1% per 1000 lines".
As has been seen, this is certainly true of seismic data processing packages, an application area that fares exceedingly well in objective comparisons of static reliability.

These results cast much doubt over any results which are the result of significant computation such as the NASA studies of the cosmic ray microwave background which inferred statistically significant structure at a level of 0.001% of the peak values. In their work, Hatton and Roberts [2] showed that the only process which achieved this level of agreement was the act of reading digital data from tape and converting its floating point format.

CONCLUSION

Not only is software riddled with errors, but when it "works", the statistically reliable precision of the answer is very much less than is thought to be the case. This is certainly true for seismic data processing packages in Earth Science and by comparison with static reliability measurements across many other industries, it is inferred that the same is true for other industries.

This precision can only reliably be measured by carrying out independent N-version experiments. Industries for which this cannot be done are at risk.

ACKNOWLEDGEMENTS

I would like to thank Enterprise Oil plc for their considerable support and also the participating companies who allowed their software to be compared. Although they benefitted by feedback about the many problems we found, their cooperation was indispensable.

REFERENCES

2. Hatton, L and Roberts A. “Measuring the disagreement between different seismic software packages: A seismic calibration


