Fortran Modernisation Workshop - Current Usage of Software Engineering

Wadud Miah[*], Filippo Spiga[†], Fatima Chami[‡], Kyle Fernandes[§]

July 20, 2016

Abstract

This article will present the results of the usage of software engineering practices by the academic computational science community who use the Fortran programming language. The statistics were gathered from running the Fortran Modernisation Workshop.

1 Introduction

The Fortran Modernisation Workshop [1] has been running in the UK and a number of workshops have been scheduled. The registration process gathers data from the computational science community who use the Fortran programming language and asked the following question:

1. Which Fortran standard are you using? Fortran 66, 77, 90, 95, 2003, 2008 and object oriented programming (OOP);

2. Are you using any software engineering techniques for code development?

[*]wadud.miah@nag.co.uk
[†]fs395@cam.ac.uk
[‡]fatima.chami@durham.ac.uk
[§]kj333@cam.ac.uk

3. Are you using any unit testing frameworks for Fortran?
4. Are you using any in-situ visualisation libraries in Fortran?
5. Are you using any version control system?

The results of the above questions are described below and their ramifications. The Fortran Modernisation Workshop covers the following topics:

- Software engineering for computational science;
- Modern Fortran standards and how to write optimized and efficient Fortran;
- NetCDF and HDF5 scientific file formats for data sharing in Fortran;
- GNU Automake to automate the build process;
- pFUnit unit testing framework for testing Fortran codes;
- Doxygen for Fortran code documentation;
- Git version control for collaborative code development;
- In-memory visualisation using PLplot in Fortran;
- IEEE Floating Point Exception Handling
- Fortran interoperability with C, Python and R;

This article will argue that there is a clear need to run the Fortran Modernisation Workshop.

2 Fortran Standards Usage

The current usage of Fortran standards is shown in Figure 1 which shows there is still usage of the Fortran 77 standard which is a nearly 40 year old standard. Fortran 90 is mostly used which is expected as it was a major revision of the Fortran language. Usage of Fortran 2003 and 2008 is still lacking which still requires promoting within the community. In addition, adoption of object oriented programming (OOP) is still widely lacking which also needs to be promoted within the community.

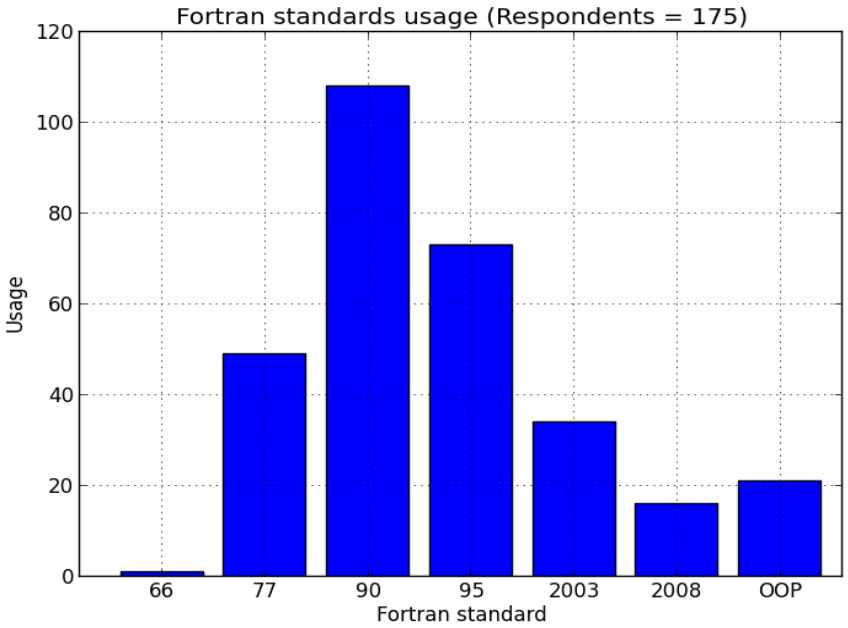

Figure 1: Fortran standards usage

The aim of the Fortran Modernisation Workshop is to stop people using the Fortran 77 standard (unless they are maintaining legacy code) and increase the usage of Fortran 2003 and 2008, as well as wider use of object oriented programming.

3 Software Engineering Techniques

The usage of software engineering techniques used by the community is shown in Figure 2.

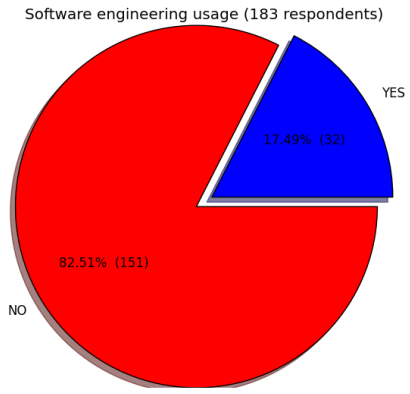

Figure 2: Usage of software engineering techniques

Figure 2 shows the vast majority of the Fortran code development community (82%) are not using any software engineering techniques.

4 Unit Testing Frameworks

The usage of unit testing frameworks used by the community is shown in Figure 3

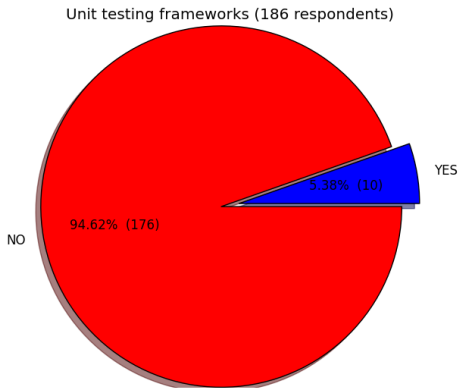

Figure 3: Usage of unit testing frameworks

Figure 3 shows that 94% of the community are not using any unit testing frameworks. This is not to say that 94% of the community are not testing their codes, but unit testing frameworks do simplify the process of testing which results in higher quality codes and better research.

5 In-situ Visualisation

The usage of in-situ visualisation libraries used by the community is shown in Figure 4

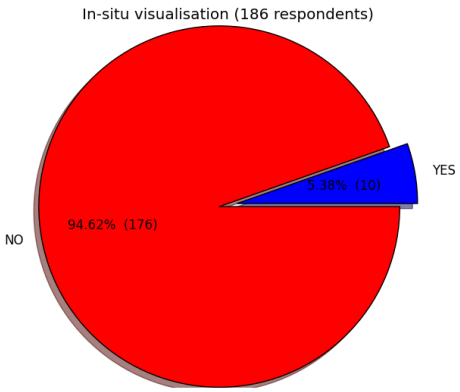

Figure 4: Usage of in-situ visualisation libraries

Figure 4 shows that 94% of the community are not using any in-situ visualisation techniques which means computational jobs (using LSF, PBS, SLURM) could be running for longer than expected. In-situ visualisation allows researchers to test the results of the codes whilst it is running. This is to determine whether it has their simulation has converged or is resulting in non-physical effects due to incorrect models and/or parameters. If any of the described events have occurred, the computational job can obviously be terminated, resulting in saved CPU cycles and energy, and increased productivity for the academic researcher.

6 Version Control

The usage of version control systems used by the community is shown in Figure 5.

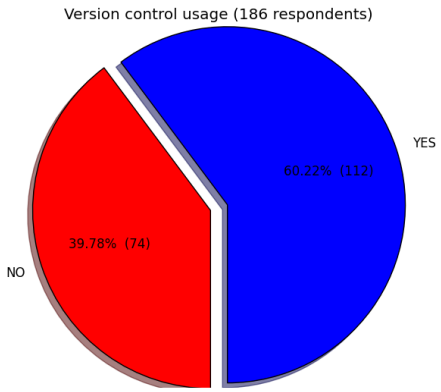

Figure 5: Usage of version control systems

Figure 5 shows 39.78% of the community do not use version control systems whilst 60.22% of the community do. Version control systems are an important tool for reproducible research as well as a tool for increased productivity, so this needs to be addressed within the academic community.

7 Conclusion

This article has attempted to show the need of the Fortran Modernisation Workshop which addresses the shortfalls highlighted in the data from the registration process. It is the authors' view that the Fortran Modernisation Workshop will address the shortfalls highlighted and increase the academic researcher's productivity and help them produce better research software and better science.

References

[1] Wadud Miah, Numerical Algorithms Group. Fortran Modernisation Workshop. `http://www.nag.co.uk/content/fortran-modernization-workshop`, 2016.