REQUIREMENTS ANALYSIS FOR ENGINEERING COMPUTATION

S. Smith[a], **L. Lai**[a] and **R. Khedri**[a]

[a]Computing and Software Department
McMaster University
Hamilton, Ontario, CANADA, L8S 4K1
smiths@mcmaster.ca, lail@mcmaster.ca, and khedri@mcmaster.ca

Software engineers advocate that the first step in system development should be a systematic elicitation, analysis and documentation of the requirements, because it is much easier and cheaper to correct mistakes and misconceptions at the beginning of the process than it is to try and fix problems during implementation and maintenance. The software engineering community agrees on the necessity of a complete and consistent software requirements document for evaluating any software system quality, including reliability [1]. Although requirements documentation has been demonstrated to be effective in other application areas, this stage of the software development is often neglected when solving engineering computation problems. This paper argues that the reliability of engineering computation can be significantly improved by adopting software engineering methodologies for requirements analysis and specification.

The argument that requirements analysis can improve the reliability of engineering computation is first made by observing the strong similarity between the waterfall model of the software life cycle and the standard model of the scientific method. The strong similarity implies that engineering computation can benefit from methods that have proved to be successful in software engineering. The paper also shows that a systematic process for analyzing and specifying requirements is beneficial because of the following: *a*) confidence is increased that all special cases have been considered, *b*) the range of applicability of the model is clearly identified, and *c*) the sensitivity of the model is considered in advance of its construction. The paper also discusses how an unambiguous set of requirements can help by ending arguments about the relative merits of one system over another, which occurs when both systems are designed using different implicit requirements. Requirements documentation can also improve the verification and validation phase, because the requirements unambiguously state the standards the system must meet. Finally, the paper shows that communication between domain experts is facilitated by formally specifying the requirements.

Besides discussing why requirements analysis is critical for reliable engineering computation, this paper also provides advice on methods to document the requirements. In particular, a requirements template is proposed for specifying engineering computation systems. To make the mathematical specification easily understandable by all stakeholders, the technique of using tabular expressions is advocated. To clarify the presentation, a case study is presented of the documentation for a system for analyzing statically determinant beams.

References

[1] Davis, A. et al. Identifying and Measuring Quality in a Software Requirements Specification, *Proc. of 1st International Software Metrics Symposium*, IEEE, pp 141-152, 1993.