USAGE OF COMPUTERS IN CALCULATIONS AFFECTING ENVIRONMENT, SAFETY, AND HEALTH

Introduction

Computer programs are widely used at Fermilab to aid engineering design**.** They are also used as a part of the Laboratory's program for evaluating operational conditions related to environment, safety, and health. These programs range from large computer source codes, supported by their vendors, to non-proprietary specialized programs maintained solely by users. Often, these programs are the quickest, and potentially most accurate, means by which to solve the complex stress, thermal, electrical, magnetic and radiation shielding problems encountered in the design of accelerators and detectors and to evaluate environment, safety, and health conditions. Improper use of these programs could result in unacceptable risks to worker safety and health or to the environment, as well as the execution of flawed designs which are difficult and expensive to correct.

Furthermore, in the present era of multi-user platforms, networks, and shared computer source codes, the opportunity for corruption of the source code is always present. Each critical usage of computer programs should therefore be scrutinized to assure the viability and reliability of the results.

For the purposes of this chapter, no distinction is made between centralized, distributed, or personal computing. All computer programs used under the supervision of Laboratory employees that could affect significantly affect environment, safety, and health-fall under the requirements of this chapter, regardless of platform. Where providers of subcontracted services employ computers as a part of their work, the primary responsibility to assure of the reliability and verifiability of the results rests with the subcontractor.

This chapter describes examples of the measures that should be taken to insure that computer calculations that could have an impact on environment, safety, and health are carried out in a manner that assures their accuracy. Examples of methods by which this

assurance can be achieved are presented. However, it is not suggested that following a specific procedure can ever substitute for professional knowledge and experience that can often suffice to assure the overall reasonableness of the work.

Certifications of computer source codes by governmental bodies, standards organizations, and/or professional societies may be considered in the initial choice of computer application to be employed. However, the person performing calculations should address the ongoing need to verify that corruption of the computer source code has not occurred using one or more of the other techniques described here.

Special Responsibilities

1. Division/Section/Center Heads have responsibility for maintaining an awareness of the use of computers in that affect environment, safety and health in their organizations. They are responsible to take steps to assure that those who make such use of computers within their organization implement the policy set forth in this chapter by performing such calculations in a manner that provides reliability and verifiability using techniques exemplified by those specified here.

2. Users of computers have the responsibility to identify those circumstances in which the use of computers has the potential to have a significant impact on environment, safety, and health. For those applications where such impacts are identified, a variety of techniques may be used to assure that the objectives of this chapter are met.

3. Personnel responsible for the use of computers that could affect environment, safety, and health should have a level of technical competency, based upon education, experience, or special training, commensurate with the calculations which need to be performed. The determination as to technical competency is not intended to be a separate formal qualification; rather it could be covered as part of other routine assessments of employee performance and development.

Techniques for Providing Assurance of Verification

Each computational problem presents its own considerations with respect to verification, and no list of techniques can cover all contingencies. In this section are given a variety of techniques, not intended to be exhaustive, which may be adapted to

use in evaluating the reliability and verifiability of calculations. Professional judgment may be used to select one or more of these techniques to use in a specific application. The users of computer programs should seek out other approaches or adaptations to these suggestions, which might be used to provide more effective or efficient assurance of accuracy for their particular application.

The method of verification chosen, whether from the following list or created by the analyst consistent with the intent of this chapter for specific analysis, should be included in any documentation of the work performed. Each calculation or set of calculations of the type addressed by this chapter should be evaluated by the user of the computer source code to verify that corruption has not occurred and that the results can be used with confidence. When practicable, these evaluations should be documented.

1. ***Analytical Calculations -*** It is often possible to compare calculations made using established analytical formulae or textbook examples to specific results from computer programs. Often, the computer model can be modified to match the analytical model, and the correspondence of the computer and analytical results taken as verification that, for a very similar, though not identical circumstance, the computer program performs accurately. Professional judgment and experience must be brought to bear on the assumptions necessary to modify the computer model to represent the typically simpler analytical problem, such that the relationship between the analytical model and the final computer model is well-understood, and the accuracy of the computer model verified as acceptable.

2. ***Testing -*** There are several ways to employ testing. One method uses measurements taken on earlier designs, and is most useful during the early stages of an analysis when the computer model is being created and tuned to the particular problem. A second type is to perform prototype or scale model testing, which typically occur further along in a design cycle, and over which the person performing the calculations can often exercise considerable control in defining the quantities he/she believes are most useful in the verification of the computer model. Results obtained from previous experience may also be used. In each case, the person performing the calculations must understand exactly how the measurements were made, and be convinced of the validity of the data before using it for verification. Some computer source codes also employ self-testing features which should be used where available and appropriate.

3. ***Comparison Against Published Works -*** Engineering and scientific literature, including international, national, or trade standards documents, often contains results which can, with careful consideration, be used to verify a computer modeling approach. This work may be in the form of measurement, analytical results, or computer modeling of a design similar to, or having aspects salient to, the engineering design in question.

4. ***Check by Another Competent Individual -*** The results of an independent analysis may be used to verify those of the person performing the calculations. This independent analysis should be performed by a competent individual who is experienced in the type of calculations being reviewed, but who is preferably not directly involved with the immediate calculations. This verification may be a set of independent analytical calculations, the creation of an entirely new computer model, detailed review of the principle analyst's model, or any other approach agreed to by both principal analyst and the reviewer to provide verification.

5. ***Benchmarking -*** This is the solving of a problem designed to exercise some aspect of a particular computer program. The answer is known *a priori*, and the results are reviewed to determine both accuracy and computational speed. Commercial programs maintained by their vendors routinely perform such benchmarks, and make them available to the analyst as part of the documentation package. User-maintained computer source codes are typically benchmarked by the person performing the calculations, who has the responsibility to design this benchmark suite such that those aspects of the calculations most relevant to environmental health and safety are exercised and verified. Benchmarking should always be done using the same version of the computer computer source code and the same platform as will be used in the actual analysis. Certain designs will severely test the limits of a particular program, and benchmarking should probe these limits to enhance understanding of program behavior. The aspects of "benchmarking" related to computation speed are not directly related to the objectives of this chapter.

Special Circumstances

1. ***Computer Source Codes Translation Between Different Platforms -*** Since different computers can have different architecture and methods of doing internal arithmetic, it is important to compare and understand the differences between the same calculation performed on different platforms. It is suggested

that this can be accomplished by *benchmarking* the results using a new platform against those obtained using an older platform in order to understand the significance of any differences which may be encountered and, if possible, their origin. This kind of benchmarking will likely be needed only once, when the computer source code is translated to a new platform.

2. ***Updated and Improved* Computer Source Codes -** The user of a code should follow such changes and updates, and if necessary, the codes should be updated to the current version. Care should be taken, especially when "centralized" computer source codes are used, to assure that the input data specifications, which are typically retained for indefinite periods of time remain compatible with the updated code. In some environments, multiple users access computer source codes in association with their own applications. In these circumstances, it is especially important to use one or more of the techniques of this chapter to verify the validity of the results in order to assure compatibility among the various applications used.

3. ***Standardization* -** In situations where the source code is available to many users, it is common that several customized "offspring" are produced by the different users. This can make comparative checks and reviews exceedingly difficult. Such source codes should be standardized whenever possible and individual users should document the modifications that they have made. Any fundamental changes to the central version of the source code should be approved, if possible, by the author of the computer source code. There should be mechanisms for obtaining the input of the user community and at the very least mechanism for informing the users of the changes.

4. ***Error Reports for Commercial Computer Source Codes* -** Quite often, vendor-supported commercial programs maintain a record of the errors known to exist in their code, and release error reports to their customers. The individual performing calculations subject to this chapter should review these error reports to ensure that his/her results is not within the regime known to have produced incorrect results for some users. If critical errors of this type are discovered, measures taken to assure a correct understanding of the results and their possible impact on environment, safety, and health. Management should be notified as appropriate if significant environment, safety, or health consequences are identified.