Consulting Fire Engineers

34 Satara Crescent
Khandallah
Wellington 6035
New Zealand

Discussion Document

Verification of Fire Dynamics Simulator

Prepared by: T.G. O'Brien, Consulting fire Engineer

Dated: 10 November 2016

Revision: R01

Table of Contents

1 Executive Summary

Fire Dynamics Simulator (FDS) is arguably the program of choice for Computational Fluid Dynamics (CFD) fire modelling.

The FDS technical documentation identifies that evaluation of the FDS model through validation and verification is critical.

Evaluation of FDS for a particular use is the responsibility of the user or the regulator.

The FDS program developer, NIST, provides current and comprehensive validation information for every FDS major and minor release that can readily assessed by the user or regulator.

However, verification is computer-platform dependant and for reasons of practicality can only be completed by the user.

The majority of FDS installations are not subjected to verification.

This discussion document considers the need for FDS verification, particularly in the context of the program's use in the design of life safety systems. It presents causes of verification numerical error, provides a rationale for completing a verification process, and suggests methods by which verification issues might be resolved.

While conclusions are left to the reader, a reference to an example verification report is included that may provide a useful template for the verification process to meet regulators' requirements.

2 Introduction

Fire Dynamics Simulator[1] (FDS) is arguably the program of choice for Computational Fluid Dynamics (CFD) solutions for certain fire engineering problems. This is because: the program is free; validation information is available for a wide range of fire phenomena; it is well supported by NIST[2]; it is subject to ongoing improvements, refinements and revisions; and it has wide community support.

Fire engineers use FDS for developing alternative and specific fire engineering designs, as well as for research. However, many FDS installations (including those of regulators and statutory authorities) are not subject to verification (defined below).

The usual FDS installation process is to download the pre-compiled binary files from NIST, run the installation file, and immediately apply the program to engineering problems.

However, the FDS installation process can be somewhat more complicated than this - often dictated by the hardware platform and operating system. Installation may require compiling OpenMPI and FDS for a particular operating system with additional compilation of software for specific hardware such as Infiniband.

If user-testing of the FDS installation is carried out at all it is usually cursory, running the couch.fds model[3] is perhaps typical. Such testing would look for fatal errors that might prevent the simulation from completing, and perhaps extend to a qualitative review of SmokeView output of smoke, HRR history, HRRPUA and, in the case of couch.fds, consumed obstructions.

It would then be reasonable to ask:

> Is cursory testing of an FDS program installation sufficient for software used in the design of life safety systems?

3 Definitions

Before answering the question posed above, we need to be clear about the terminology being used. It is appropriate in the context of FDS to use definitions contained in the FDS technical documentation[1, Vol. 2, Preface]:

> *Model evaluation consists of two main components: verification and validation.*
>
> *Verification is a process to check the correctness of the solution of the governing equations.*
>
> *Validation is a process to determine the appropriateness of the governing equations as a mathematical model of the physical phenomena of interest.*

Note that these definitions are not used consistently in associated literature and terminology has changed over time. Terms such as benchmarking are often used, and the distinction between validation and verification become blurred, or even contrary to those above.

4 The Need for FDS Evaluation

Whether a cursory test is sufficient for software to be used in the design of life safety systems is addressed succinctly in the preface to Volume 2 of FDS Technical Reference Manual[1, Vol. 2]:

> *Evaluation is critical to establish both the acceptable uses and limitations of a model.* (my emphasis)

FDS practitioners will be familiar with the program's extensive documentation and may understand the significance of warnings like '*fragile* and *critical'* which, my opinion, are often understated.

The FDS Users Manual[4] also provides guidance on the need for installation testing. Section 2.4 states:

> *If you are running FDS under a quality assurance plan that requires installation testing, a test procedure is provided in Appendix B of the FDS Verification Guide [3]. This guide can be obtained from the FDS-SM.*

This statement is certainly not as compelling as the Technical Reference Manual's dire warning. However, fire engineers designing for building code compliance may be subject to a quality assurance plan to conform to regulator accreditation requirements.

4.1 Accreditation

In New Zealand the regulator for building code compliance is the relevant Building Consent Authority (BCA). BCA's are independently accredited to carry out this function.

A fundamental tenet of quality systems is that subcontractors and suppliers to an accredited organisation should also be appropriately accredited. So fire engineers should hold relevant accreditation for fire engineering designs submitted to BCA's for building code compliance. The very least that a BCA might expect is that fire engineers have an acceptable quality assurance regime. If this does not hold then the accredited organisation (the BCA) must apply its own quality procedures to ensure an appropriate level of quality for submitted fire engineering designs.

FDS validation and verification are somewhat problematic for regulators. While validation may be completed by the regulator through the Fire Engineering Brief stage of an IFEG[5] design process, verification requires access to a particular FDS installation, and in either case regulators may not be adequately resourced to undertake these functions.

In research applications there is no regulatory incentive for evaluation of FDS however in keeping with scientific methodology experiments should be repeatable, designed to minimise error, and findings are subjected to quality control through the peer review process. While validation may not be relevant to research into new FDS applications, verification is both relevant and necessary for robust results.

5 Evaluation through Validation and Verification

5.1 Validation

Validation is not computer-platform dependant and every major or minor release of FDS is subjected to a validation process by NIST. NIST provides a current and comprehensive body of evidence that should allow an end-user or regulator to determine if FDS is appropriate for a particular use through a relatively straight forward review process. A proviso is that users do not tamper with the source code (something that NIST do not recommend).

Although NIST's validation process is comprehensive and well documented[1, Vol.3] the following points should be noted: it is impossible to test every aspect of the code; FDS has limitations such as the low Mach number approximation; and FDS has aspects of fragility (many of which are described in the program's documentation).

The acceptability of NIST's validation process is arguably the responsibility of the end-user or the regulator who must determine if FDS is capable of adequately predicting metrics of interest for a specific design. The paper Fire Model Validation – Eight Lessons Learned[6] advises:

> *A common misconception about model validation is that it is the responsibility of the model developers. Actually, it is the responsibility of the end users or regulatory authority (AHJ) acting on their behalf. After all, to say that the model has been verified and validated means that it has been deemed acceptable for a particular use by the end user or AHJ. The model developers might contribute examples demonstrating the model's reliability and accuracy, but they cannot make the decision as to whether the model is appropriate or not.*
>
> *All this being said, it is unrealistic to believe that end users and/or regulatory authorities will have the resources to thoroughly evaluate all aspects of a model, in particular a CFD model which has such a wide range of potential applications. Thus, the model developers do a considerable amount of work that others might review to make their assessment as to whether the model is appropriate or not. Model developers can assist in the process by organizing and documenting case studies that can be periodically updated as new versions of the model are developed. Anyone using the model should be able to examine the experimental reports, input files, assumptions, and so on, to determine if the model is appropriate.*

5.2 Verification

Verification is computer-platform dependant and therefore cannot reasonably be completed by the program developer (NIST) or the regulator.

The output of a given FDS model on a particular hardware platform, with a particular allocation of computational resources, should produce identical results. The same model, run on different computer platforms or with different computational resource allocations (for example different nodes in a computer cluster, or a change in the number of OpenMP threads, or OpenMPI processes), can produce variations in the model outputs. These two statements are borne out in practice.

It is worth noting that for a large computer cluster with different node-build specifications it is impractical to test all node combinations through the verification process.

The reasons for output variation are cumulative effects of computational numerical precision and the hardware and software implementation of mathematical functions.

Verification looks specifically at the output of the iterative numerical calculation performed by FDS on a particular computer platform. It compares the output of predefined models that exercise various aspects of the FDS program against analytical metrics and tolerances provided by NIST.

While NIST cannot complete the verification process, except on their own computer platforms, they have provided a verification procedure which may satisfy the evaluation criteria of end-users and regulators. The NIST minimal installation testing suite of FDS models is listed in Appendix B of Volume 2 of FDS Technical Reference Manual[1, Vol. 2]. It comprises 15 FDS models that look at fundamental aspects of FDS functionality.

The NIST minimum installation verification models can be run manually in about an hour, even on a modest computer platform. NIST also provides script (Linux) and batch (Windows) files to automate the verification process. Thus the FDS verification process cannot be considered to be a significant burden for the end-user unless verification fails, in which case there are likely to be issues with the FDS installation that need to be addressed.

6 Numerical Variability

How significant are the numerical variations in the verification process? The answer to this question is entirely computer-platform dependant. This is one reason why the verification process is important. If we don't attempt to measure the variations then we will have no idea of whether or not they are significant.

The limits[1, Vol. 2, Appendix B] for the minimal verification suite of models established by NIST are either absolute or relative, generally with an error tolerance of 1%, although tolerances are quite specific to the metrics being evaluated.

There appears to be no reported instances of FDS verification failure, but given that FDS is seldom actually verified this is not surprising. I have completed FDS verification of a number of computer platforms and some have failed to achieve all of NIST's verification criteria. Even NIST's own verification process results[1, Vol. 2, Appendix A] show a predicted radiation metric from the minimal verification suite exceedingly close to the established pass/fail criteria
(error: 9.93E-3, error tolerance 1.00E-2).

We can look at several aspects of numerical precision in order to understand how FDS variations in numerical output might arise (refer to Appendix A) . A useful primer on computational numerical error can be found in Numerical Recipies[7]. This topic is fundamental to computer science and there are many other texts that could be cited.

A single numerical error near the level of machine precision (epsilon) is unlikely to be significant in an FDS simulation. However over the course of an FDS simulation, perhaps involving billions (10E9) of iterations, errors may cancel or compound.

Many computational numerical functions are hardware-dependant (determined by physical structures in the silicon of the CPU or mathematics co-processor). However the accuracy of numerical functions may also be subject to the code generated by the compiler. We can therefore reasonably expect different results for an FDS simulation in different hardware and software environments.

It is worth reiterating that any given model run multiple times on a particular FDS installation, on a particular computational platform and with a particular allocation of computational resources, should produce identical results (independent of the relative correctness of the result).

7 Why Bother with Verification?

There are many reasons why you would want to complete the FDS verification process.

- Because the FDS Technical Manual states that evaluation is critical, verification is part of the evaluation process, and verification is computer-platform dependant.
- Verification will give confidence that FDS output is not subject to errors associated with the computational platform.
- If you run a quality management system then you may have an obligation to your accreditation body to verify that your equipment is calibrated.
- On some fire engineering work your client or the regulatory authority may require you to verify your FDS installation.
- Verification is not a significant overhead, particularly when compared to the time that might be required to compile and install FDS on a multi-node network, or run a simulation.
- While verification provides a quantitative measurement of system performance, it also provides a defined initial model processing regimen that will test for fatal errors that might prevent a simulation running to completion.
- Verification can provide a marketing advantage.
- In the litigation of fire loss of a building that was designed using FDS a question that might be asked is ‘was FDS verified?’ It is unlikely that the original FDS installation can be reconfigured exactly. While it may be possible to re-run the simulations on a verified platform and confirm that the results were sound, the damage to credibility will have been done.

8 Resolving Verification Problems

A final question that warrants consideration is: ‘What can I do if verification fails?’ Identifying a problem without a means of resolution is clearly undesirable. Here are a few suggestions for proceeding should this occur.

In the first instance you might assess the type and magnitude of the verification failure, document it, and consider if this will have a detrimental effect on your specific modelling requirements. Few simulations use all of FDS’s features and many (for example radiation from a small fire in a large compartment or HVAC) may not be relevant to a particular model. The converse is also true, so the minimal installation verification suite may not adequately consider aspects of the model that are of interest in a design. This may warrant delving into the comprehensive list of verification models[1, Vol. 2, Appendix A] to complete a more appropriate verification for a particular project.

Reviewing the verification model and the associated FDS documentation[1] may assist in understanding the cause of a verification problem.

Adjusting FDS model parameters may resolve verification issues. For example increasing the number of radiation solid angles from the default value of 100 will improve the computational accuracy towards the analytical result (although at the expense of additional simulation time).

You might consider the type and version of your operating system and compiler, and the version of FDS. In general you should be running the most recent version of FDS, although updating may not be practical mid-way through a CFD modelling project. Your compiler flavour and optimizations can have a significant effect on numerical performance, stability and processing efficiency. Interestingly, reduced compiler optimization does not always lead to improved stability.

You might also consider the build-state of your computer and networking hardware, particularly on computer clusters. I am aware of at least one High Performance Computer (HPC) cluster where network hardware caused FDS instability under OpenMPI. This was identified through the verification process and resolved by replacing network hardware components.

Finally, you should consider contacting NIST and reporting verification issues. NIST specifically invite feedback on these matters, but be sure to include all of the information they require for investigation[4]. NIST (and the wider community) may be able to provide advice on possible solutions and assist in determining if the problem that you are experiencing is specific to your computer platform, or if it has wider implications for other FDS users. My experience has been that NIST are extremely responsive and exceedingly helpful.

9 Final Words

While some practitioners may regard FDS verification as a pointless exercise and a potential compliance barrier, others see it as a useful risk management process that enhances quality assurance.

As this document is intended for discussion it is left to the reader to draw their own conclusions regarding the need for FDS verification.

If you are interested in the verification process you can download an example FDS verification report[8] based on the NIST minimum verification suite[4, Appendix B] through the ‘FDS Verification Report’ download link at:

http://fire.aquacoustics.biz/html/publications.html

10 References

[1] McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., & Overholt, K., Fire Dynamics Simulator, Technical Reference Guide. National Institute of Standards and Technology, Gaithersburg, Maryland, USA and VTT Technical Research Centre of Finland, Espoo, Finland. 6th ed., , Vol. 1: Mathematical Model; Vol. 2: Verification Guide; Vol. 3: Validation Guide; Vol. 4: Configuration Management Plan, FDS Version 6.5.2, Revision Git-r21-0-g4e9103f, 24 August 2016

[2] National Institute of Standards and Technology, Gaithersburg, Maryland, USA, US Department of Commerce

[3] couch.fds FDS Model distributed with FDS 6.5.2[1] under Examples/Fires

[4] McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., & Overholt, K., Fire Dynamics Simulator, User's Guide, Chapter 4, National Institute of Standards and Technology, Gaithersburg, Maryland, USA and VTT Technical Research Centre of Finland, Espoo, Finland. 6th ed., FDS Version 6.5.2, Revision Git-r21-0-g4e9103f, 24 August 2016

[5] NRC, ICC, DBH, ABCA, International Fire Engineering Guidelines, 2005 Ed., Australian Building Codes Board, 2005

[6] McGrattan, K., Peacock, R., & Overholt, K., Fire Model Validation – Eight Lessons Learned, International Association for Fire Safety Science, Fire Safety Science - Proceedings Of The Eleventh International Symposium, 2014, pp. 958-968

[7] Press, W., Flannery B., Teukolsky, S., & Vetterling, W., Numerical Recipies: The Art of Scientific Computing, Cambridge University Press, 1987

[8] FireNZE, Fire Dynamics Simulator (FDS) Verification Report on the FireNZE Linux Cluster, Rev. R02, 24 September 2016, http://fire.aquacoustics.biz/html/publications.html

Appendix A Numerical Error in FDS

The representation of real numbers in the computational domain is discrete with a resolution determined by the number of bits used to represent numbers. This is not necessarily the native bit length of the processor being used (typically 32 or 64 bit) but there are computational efficiencies in speed and storage when using native bit length.

When represented in exponential format (with a signed significand, signed exponent, and perhaps check bits) the resolution of relatively large positive or negative real numbers is necessarily lower than relatively small numbers. So the computational number line is not only discrete but the gaps between numbers vary with magnitude.

The nature of the computational number line leads to a number of potential sources of numerical error in mathematical operations which can be generally classified as:

maximum and minimum limitations,

precision and round-off,

repeatend binary numbers,

operations with near zero results, and

implementation of transcendental functions.

The last four of these error classifications can be expected to be relevant at some point in an FDS simulation.

If you have read this far then you might like to experiment with some Microsoft Excel examples at https://support.microsoft.com/en-nz/kb/78113. These show how simple addition (a fundamental operation for algebraic functions) can lead to numerical errors. Subject to the operands it is possible that a single binary floating point addition can lead to a result without a single correct binary digit!

While integers have an exact representation in binary arithmetic (within maximum and minimum bounds) decimal fractions can result in repeatend binary numbers. The binary representation of 0.1 (decimal - base 10) is 0.000110011..., a repeatend with no exact representation in binary.

In the numerical analysis of fire simulations we do not usually anticipate instantaneous transients in fire phenomena. Changes occur gradually over time (certainly subsonic) and, perhaps with the exception of radiation and HVAC, transport throughout the physical computational mesh occurs relatively slowly.

In fact transient behaviour is often the cause of FDS program failure usually attributable to modelling error. Modelling techniques, such as ramping constant heat sources from zero at the beginning of a simulation to their constant value over 10 to 15 seconds of simulated time, are recommended to reduce transient behaviour and associated numerical instability.

As a consequence of relatively slow changes between simulation iterations we can reasonably expect adjacent domain cells to have similar states (with the obvious exception of obstruction boundaries). It is the difference between adjacent cells and applied forces that leads to flow fields in the domain. These are calculated as finite differences between adjacent cells resulting in algebraic operations on similar magnitude operands with near zero results.

Radiation transport involves the computation of trigonometric (transcendental) functions which are often computed as the sum of a number of terms from an infinite polynomial series (for example, the Maclarum series, a special case of the Taylor's series where the argument x is assumed to be near zero). They are often calculated with irrational arguments (typically a product of the identity pi). Theoretically the precision of the calculation can be increased by incorporating additional polynomial terms, but practically this is limited by the rapid accumulation of rounding error. It is not surprising that programs such as Microsoft Excel do not calculate the analytical result of zero for sin(n * pi), where n is a positive integer. The numerical solution is different for each value of n, although still within the double precision limits of IEEE 754.