

The Challenge of Maintaining FDS and CFAST

Kevin McGrattan

National Institute of Standards and Technology
Gaithersburg, Maryland, USA

Abstract. The paper discusses the day to day challenges of maintaining and developing complex open source software.

1. Introduction

The Fire Dynamics Simulator (FDS) has been in the public domain for 16 years, and the zone fire model CFAST (Consolidated Fire and Smoke Transport) has been around, in one form or another, for nearly twice that long. While both are widely used and well regarded within the fire protection community, most end users do not fully appreciate the challenge of their maintenance and upkeep. This is true of most of the free and very useful publicly available open source software that we have come to rely on. Indeed, I am composing this paper using a new document preparation software tool named Madoko that I just discovered yesterday. I had a similar experience 26 years ago when I wrote my doctoral thesis using TeX, even before LaTeX became popular outside the mathematics community.

I must confess that I do not know much about the organizations or people who create the wonderful free software tools that I use everyday for both work and home life. I assume that there is usually some financial motivation, like selling advertising space or user data or a not-free “premium” version of the software. This isn’t a new idea – we’ve been watching and listening to “free” TV and radio programs for a century for the price of being subjected to advertisements.

But this is not the case with FDS and CFAST. Their development is an integral part of our fire research program at NIST and collaborating organizations. These computer models are the primary means of transferring basic research results in fire into practice. Because of their role in “technology transfer”, most of the developers, past and present, are not fire protection engineers nor combustion researchers because software development requires a considerably different skill set. In fact, most developers are essentially applied mathematicians and computer scientists, whether or not our college degrees actually state it. Our job is to translate fundamental combustion research into differential equations whose solution is of value to practicing fire protection engineers. Thus, the model developers interface two very different communities – basic combustion research and fire protection engineering. These two groups live in very different spheres – they have different degrees, work for different organizations, and attend different kinds of meetings and conferences. The models try to bridge the gap.

This presents two fundamental challenges. First, researchers have historically viewed archival journals as the primary means of communicating their findings. Second, end users often regard these models as black boxes that spit out results and require little in the way of understanding basic fire phenomena. These two challenges have forced the model developers to do far more than just numerically solve the ordinary and partial differential equations that describe basic fire behavior. The long term viability of software like FDS and CFAST will require a considerable change in attitude of both researchers and end users because it is becoming increasingly difficult for the model developers to play all three roles – researcher, software developer/custodian, and end user. No doubt, we cannot avoid completely all three, but if the perception in the community that NIST or similar organization can do it all, then the effort will certainly fail.

2. A Brief History of FDS

I joined the staff of the Fire Research Division of the National Institute of Standards and Technology in 1992, after working for a year as a post-doctoral fellow in the Computing and Applied Mathematics Laboratory of NIST. The year before, I had graduated from the Courant Institute of New York University with a doctorate in mathematics. I assumed that I would eventually make my way to a teaching position at a university. However, teaching jobs were not easy to come by, and when offered a job in the Fire Research Division, I jumped at the chance. I knew nothing about fire, but it sounded interesting. At that age, everything sounds interesting. For the next few years, I worked three very different projects: (1) CFD fire calculations with Ron Rehm and Howard Baum, (2) microgravity flame spread calculations with Takashi Kashiwagi, and (3) mesoscale oil fire plume dispersion with Dave Evans and Doug Walton. On any given day, I worked on three separate simulation programs, with length scales ranging from millimeters to kilometers.

I soon discovered that my supervisors lived in completely different worlds. Ron, Howard and Takashi would go to Combustion Institute meetings and interact with chemists, physicists and applied mathematicians in a very academic environment. Evans and Walton would go to meetings held by their sponsors at the U.S. Minerals Management Service and U.S. Coast Guard. These meetings were anything but academic. Within a few years, I grew fairly comfortable with both sets of “stakeholders”, but it was clear that these groups just did not interact. Through the 1990s, I continued to work in these very different worlds, but by the end of the decade, I started to wonder about my future in fire. I was also starting to wonder about the impact of our work. The work on the large oil fires led to the development of a program called ALOFT (A Large Outdoor Fire plume Trajectory model), and the establishment of air quality guidelines for controlled burns of spilled crude oil. My more academic research, on the other hand, really hadn’t led to anything but a lot of papers and conferences. Seeing how these CFD calculations could actually be useful led me towards the development of FDS.

FDS was really just an amalgamation of those codes that I worked with throughout the 1990s. It became very tedious maintaining all of them, and so I combined the best of each and started working on just one code base. This seems perfectly obvious now, but at the time, it was not. Fire models proliferated throughout the 1980s and 1990s. In fact, according to a survey done by Combustion Science and Engineering, over 50 zone fire models and 10 CFD fire models were developed during this time period. Most died as soon as the funding did, and only a handful are left today. There are several reasons for this, but the simplest is that software maintenance is a thankless task. It's a lot of fun to write a new program, but not so much fun to keep it working reliably year after year.

This discussion rarely arises in academic settings because it is assumed by most researchers that their job is to do research, not development, and certainly not maintenance. Universities and research labs are supposed to do “fundamental” research and publish the findings in archival journals. This basic research then forms the backbone of commercial products and services. But for the field of fire protection engineering, and I suspect similar niche disciplines, there's a problem – and, of course, the problem has to do with money. There is just not a big enough customer base to support fire models like FDS and CFAST. There might be a market for modeling and computing services, and handy tools such as PyroSim, but there is not a large enough market, in my opinion, to support the research behind the models.

And so I was confronted with two very difficult challenges – the very large gap between academic fire research and the practicing engineers, and the thankless task of software maintenance. A solution to both problems, or so I thought, was to release FDS as an open source application that would draw academics to do their research, and engineers to design their sprinkler systems. It just seemed to make so much sense, until I remembered back in college when I took a course in Marxist economics. It all seems reasonable unless you actually have to make a living.

3. The Lesson of CFAST

FDS and other CFD models of fire did not naturally evolve within fire science – zone models did. If you flip through a text book on fire science, you'll see a steady progression of empirical correlations leading towards the two-zone compartment model. If you have a degree in fire protection engineering, you have undoubtedly sat through many lectures where the professor draws a little dog house on the chalk board, with just one door on one side, and sketches the fire, the plume, the ceiling jet, the neutral plane, the Bernoulli flow in and out of the door. This is “classic” fire science, with the so familiar \dot{m} , \dot{Q}^* , and so on. I've seen this lecture a hundred times. It was a great advancement, and according to a [survey performed by the firm, Combustion Science and Engineering](#), it led to the development of roughly 50 programs to solve the set of ordinary differential equations for the layer temperatures, height, and compartment pressure. Anyone with some background in numerical methods could write a very basic zone model

in a few hours, and if you add in multiple compartments and a host of other bells and whistles, maybe a few weeks. But writing the computer program is the easy part. What about verification and validation? What about all those uncertainties in the assumptions that underly each and every subroutine? How do you explain to the authority having jurisdiction that the calculation is valid? The vast majority of the now defunct zone models on the CSE web site were written with none of this in mind. Most were developed by a few students, who wrote a few papers, graduated, and left the program to rot on some old computer at the back of the fire lab.

So the lesson to be learned from CFAST is that it's fairly easy to develop a fire model, of one sort or another, but it's quite another to make it usable, verified, validated, etc., and keep it that way year after year. Rick Peacock, the primary caretaker of CFAST, has announced his retirement in 2017. There are no immediate plans to replace him. The reason is that it's not an appealing job for a young researcher, or so it would seem. However, I spent some time last year working with Rick to streamline the core solver, eliminate routines that were not being used or that were never verified and validated, simplify the documentation and graphical user interface, and update its maintenance process to conform to that which we use for FDS. In doing this, I found that there are a wealth of interesting problems, unique to zone models, that have never really been solved satisfactorily. Yes, a lot of work was done in the past looking at buoyant flows through ceiling vents and spill plumes and so on, but somehow much of the basic research never made it into CFAST, and it lies scattered throughout the literature. Many of the routines that were included in CFAST were never formally verified and validated. CFAST is an ideal topic of study because it is much more aligned with the curricula of fire protection engineering programs than FDS is. To really work with FDS, you need a graduate level understanding of partial differential equations and CFD. To work with CFAST, the bar is not so high, and there are plenty of interesting topics for masters degree students. The challenge for us, however, is convincing these students and their advisors to work with us to get that basic research into usable form and help to keep it that way.

4. The Problem with Papers

A major problem in maintaining models like CFAST and FDS is that the work is seen by many as completely separate from research, the goal of which is to publish archival journal papers. For centuries, progress in science and engineering has been promulgated by peer-reviewed publications in archival journals. Any young aspiring researcher knows that the pathway to success in academia is a long list of papers. And, indeed, many of the numerical techniques in our models were obtained from the general purpose CFD literature, and sometimes more specialized journals in combustion, fluid flow and meteorology. These papers illustrate basic finite differencing techniques, much like a textbook would. However, archival journals are not a particularly effective means of describing models like FDS because it is impossible to include all of the relevant details within a 10

or 20 page paper. Some CFD modelers publish a great many papers in various journals, creating a form of documentation via a list of references. This may be an effective way of advertising one's research work, but it is a terrible way of documenting a CFD model. I learned this lesson the hard way after releasing the first version of FDS in 2000. I assumed that my and others' papers in the journals would supplement the fairly sparse description of the model that I included in the manuals. What resulted was chaos – random snapshots of different versions of the code applied to various fire scenarios by dozens of students who were just learning about fire and CFD. The quality of these papers was, in general, poor, and they led to misconceptions about the basic model that still remain today. It was not until about 2007 when Bryan Klein put FDS, followed by CFAST, under version control (Sourceforge, then GoogleCode, now GitHub). At the same time, our collaboration with the U.S. Nuclear Regulatory Commission led to the publication of our verification and validation guides, which now collectively document thousands of test cases that we run on a regular basis to ensure that the code runs consistently and accurately. Our continuous integration process, developed by Bryan and then extended by Kris Overholt, is something we do on a daily basis, as described in the FDS Configuration Management Plan. Much of this work is laborious and somewhat tedious, but it is an essential part of our model development strategy because when dealing with hundreds of thousands of lines of source code, mistakes are inevitably made which can be caught and corrected within a day. Contrast this with journal publications on a roughly three year cycle commensurate with the duration of a typical graduate student's research work.

This is only one drawback of archival journals. Another is that talented researchers and aspiring students who want to help us are by necessity pressured to write papers to continue their studies and attract funding. On virtually a daily basis, we receive a request from a graduate student to join the FDS Discussion Group, and we are very gratified to have this potential pool of talent to help us. But very rarely does the work of the student affect any change in the model, whether it be the source code itself, its documentation, or its experimental validation test suite. In fact, except for a question or two, we rarely hear from the student again until, maybe, a paper is sent to one of us developers for review for a journal. Invariably, the paper discusses an old version of the code, or perhaps a bug we've already fixed, or sometimes a potential new feature that has no way of being implemented once the student has graduated and started working a real job. This has happened so many times that it's become an almost daily frustration among the handful of us who maintain FDS and CFAST. Given the incredibly powerful tools that we have at our disposal for doing collaborative software development, we revert back again and again to a time-honored system of scholarship that is simply out of date.

To be fair, I suspect that students and their advisors worry that they could not possibly make a significant contribution to these fairly complex programs that have evolved for decades. This may well be true – it's unrealistic to expect a two year masters student to make that paradigm-shattering change in a science

that’s been around since the Stone Age or computer programs that pre-date the Internet. But that’s not what we’re looking for – the promise of open-source software development is to harness the power of hundreds of contributors, however small their contributions might be. How small? Even a bug report that leads to a minor code improvement is a contribution. An experimental dataset. A clarification of the manuals. Almost anything, so long as the contents of the program’s on-line repository is changed. Even by one byte – a byte is one character, which might be a change in an empirical coefficient from 0.2 to 0.3. The FDS and CFAST repositories are roughly 1 GB in size, which means there are a lot of bytes that might need changing.

5. A Path Forward

For those of you who are interested in fire research, and you want to improve these fire modeling programs, here are a few steps to take:

1. Identify a problem in the existing models. There are already some listed in the FDS Road Map, but you need only simulate some new fire scenario to discover that there’s undoubtedly numerical or physical assumptions that can be improved. Each and every time I try FDS on a new fire scenario, I compile a list of very interesting masters or PhD thesis topics. At the moment, and this of course changes month to month, we are interested in wildland fire spread, pyrolysis, parallel processing (in particular MPI, message passing interface), validation experiments that expand the current parameter space of existing experiments, and immersed boundary methods (IBM) that will move FDS beyond rectilinear geometries and “stair stepping”.
2. Contact one of the core developers to make sure that your idea fits in with our current development plans. Do not wait until your paper or thesis is written and then present it to us as a finished work. We simply do not have enough hands to make the changes to the code or documents ourselves.
3. Be open to learning Git, LaTeX, Matlab, Fortran, and whatever other software we might use to maintain the codes and supplemental materials. You need not be an expert in any of these, but you will benefit tremendously from the investment, even if your career path veers away from fire, which in most cases it inevitably will. These are the tools that we use in our virtual laboratory, and much like a real lab, you simply cannot work without knowing how to use the tools. After working with these tools for a while, you will find that the basic Microsoft Office Suite is not particularly useful for doing serious technical work.
4. If you want to publish your work in a journal, work with us first to add your contribution to the FDS or CFAST GitHub repositories to document your contribution. Consider this activity to be of mutual benefit – you can share your work with a larger audience through the journal, and have a direct impact on the model users and developers. Much of the resistance that we get from professors is that the students must write a thesis and follow-up paper(s), and that they see documenting the work in the FDS or CFAST

repository as extra work. In fact, it need not be extra work at all, save for perhaps a bit of cutting and pasting from the model documents into the thesis.

Beyond the academic arena, for those of you who use our models for day to day fire protection engineering, be a smart user. When something doesn't seem quite right in one of your calculations, spend some time and put together a simple test case for us to debug. Yes, this will cost you a bit of time, but it will lead to better understanding of the model and help you in the long run. Consider it an investment that is well worth your time, and a small cost to pay to keep these models in the public domain.