Quality assurance and the simulation of fires

- A practical application for automated validation of user-generated input data for Fire Dynamics Simulator

Kristoffer Hermansson

Department of Fire Safety Engineering
Lund University, Sweden

Brandteknik
Lunds tekniska högskola
Lunds universitet

Report 5501, Lund 2015

Quality assurance and the simulation of fires - A practical application for automated validation of user-generated input data for Fire Dynamics Simulator

Kristoffer Hermansson

Lund 2015

Kvalitetssäkring vid simulering av brandförlopp - En praktisk tillämpning för automatiserad granskning av användargenererad indata till Fire Dynamics Simulator

Quality assurance and the simulation of fires - A practical application for automated validation of user-generated input data for Fire Dynamics Simulator

Kristoffer Hermansson

Report 5501
ISSN: 1402-3504
ISRN: LUTVDG/TVBB--5501--SE

Number of pages: 83
Illustrations: Kristoffer Hermansson

Keywords
FDS, Fire Dynamics Simulator, fire safety, fire simulation, quality assurance, automation.

Sökord
FDS, Fire Dynamics Simulator, brandsäkerhet, brandsimulering, kvalitetssäkring, automation.

Abstract
In order to systematize and rationalize the quality control process for FDS (Fire Dynamics Simulator) input data a software application for this specific purpose has been developed. In order to study the occurrence of user related errors in FDS input data and the possibility of reducing such errors by using the developed application, a case study on archived FDS input data from actual building projects has also been performed. Controls, which should be part of the quality control process, were identified by performing a literature review—covering building code requirements as well as recommendations from different stakeholders—and implemented in the application. FDS input data was collected in cooperation with Swedish consultants and evaluated using the application. Results have been presented in the form of checklists containing controls which should be performed as part of the quality control process, in the form of the actual application, and in the form of a summary of the frequencies with which different errors occurred in the studied FDS input data. The most frequently occurring errors were either of the kind where an intended parameter value was not correctly specified or closely related to the model mesh. It has been concluded that the application is a useful tool in discovering errors in FDS input data, thereby providing a means of reducing their occurrence.

Brandteknik
Lunds tekniska högskola
Lunds universitet
Box 118
221 00 Lund

http://www.brand.lth.se

Telefon: 046 - 222 73 60

Department of Fire Safety Engineering
Faculty of Engineering
Lund University
P.O. Box 118
SE-221 00 Lund
Sweden

http://www.brand.lth.se

Telephone: +46 46 222 73 60

Acknowledgements

For assisting me in the process of completing this project, I would like to thank:

Everyone participating by providing me with actual data, without which this project would not have been possible.

My supervisor, *Nils Johansson*, for providing valuable feedback.

Kristoffer Hermansson
Lund, 2 June 2015

Sammanfattning

FDS (Fire Dynamics Simulator) är en modell för simulering av brandförlopp, som ofta används inom ramarna för olika sorters riskbedömningar. En vanlig tillämpning är för verifiering av att myndighetskrav uppfylls i samband med projektering av byggnaders brandskydd. Simuleringar av brandförlopp med FDS utgår från användargenererad indata vars betydelse är avgörande för kvaliteten hos de resultat som sedan erhålls. Av denna anledning utgör oupptäckta felaktigheter i indata ett problem, eftersom de riskerar leda till att de riskbedömningar som simuleringarna utgör underlag för görs på felaktiga grunder.

I syfte att uppnå bättre underlag för riskbedömningar baserade på resultat från FDS-simuleringar har processen för kvalitetssäkring av användargenererad FDS-indata här utvecklats. De huvudsakliga målsättningarna i projektet har varit att fastslå i vilken utsträckning kvalitetssäkringsprocessen för FDS-indata kan effektiviseras och systematiseras genom att använda ett automatiserat mjukvaruverktyg och att fastslå huruvida förekomsten av användarrelaterade fel i FDS-indata kan reduceras genom att använda ett sådant verktyg. För att uppnå dessa mål har ett faktiskt mjukvaruverktyg för att delvis automatisera kvalitetsgranskningen utvecklats och utvärderats genom att studera förekomsten av användarrelaterade fel i arkiverad FDS-indata från faktiska brandskyddsprojekteringar.

Projektet har utförts i tre huvudsteg. Först genomfördes en litteraturstudie för att fastslå vilka kontroller som bör ingå vid kvalitetskontroll av användargenererad FDS-indata. Sedan utvärderades respektive kontroll för att fastslå huruvida den kunde automatiseras och genomföras med hjälp av ett mjukvaruverktyg och där så bedömdes möjligt implementerades kontrollerna i ett sådant verktyg. Slutligen genomfördes en fallstudie där det utvecklade verktyget användes för att utvärdera arkiverad FDS-indata från faktiska projekteringar. Avsikten med detta var att fastslå huruvida verktyget kan användas för att reducera antalet fel i indata samt vilka fel som är vanligast förekommande.

Resultat har redovisats i form av två checklistor omfattande de kontroller som bör ingå vid kvalitetsgranskning av användargenererad FDS-indata; en kvalitativ checklista avsedd att kontrolleras manuellt av användaren och en kvantitativ checklista avsedd att kontrolleras med hjälp av det utvecklade verktyget. Vidare har ett faktiskt verktyg presenterats, utvärderats genom en fallstudie och publicerats. Noterbar är att verktyget har visats kapabelt att upptäcka fel i samtliga av de utvärderade indatafilerna. Den frekvens med vilken respektive fel förekom i fallstudien har också redovisats. De vanligast förekommande felen i studien var antingen sådana där ett avsett parametervärde inte specificerats korrekt eller relaterade till meshen. De fel som ingick i den första gruppen var att utdata, tillväxthastighet, andel bränslemassa omvandlad till kolmonoxid eller koldioxid, förbränningsvärme och effektutveckling inte angivits som avsett eller, för BBRAD-scenarier, som kravställt. De fel som ingick i den andra gruppen var att $D^* / \mathrm{d}x$ understiger 10, mesh-grupper inte anges i ordning från finast till grövst och att mesh-celler inte är kubiska. På grund av studiens begränsade omfattning har inga generella slutsatser avseende vilka användarrelaterade fel som är vanligast i FDS-indata dragits. Trots att resultaten kanske inte är representativa, bör de dock kunna tjäna som en indikation på vilka områden som bör behandlas med störst omsorg vid utformningen av FDS-indata.

Gällande de huvudsakliga målsättningarna har slutsatserna dragits att det utvecklade verktyget utgör ett användbart verktyg för att upptäcka fel i FDS-indata och därigenom även för att minska deras förekomst.

Summary

FDS (Fire Dynamics Simulator) is a model for simulating fires, which is often used in the context of performing different sorts of risk assessments. One common application is in the verification of compliance with governmental requirements in the design of fire protection for buildings. Simulations of fires in FDS are based on user-generated input data in the form of text files, which are crucial to the quality of the results obtained by running the simulations. For this reason, undiscovered errors in input data present a problem in that the risk assessments, for which they constitute the foundation, may be based on incorrect information.

In this project, for the purpose of providing better foundations for risk assessments where the results of FDS simulations constitute the decision basis, the process for performing quality controls of user-generated FDS input data has been further developed. The main objectives of the project have been to determine to which extent the quality control process for FDS input data can be systematized and rationalized by using an automated software application and to determine whether the occurrence of user related errors in FDS input data can be reduced by using such an application. To reach these objectives, an actual software application designed to automate parts of the quality control process has been developed and evaluated by studying the occurrence of user related errors in archived FDS input data from actual building projects.

The project was performed in three main steps. First, a literature review was performed to determine which checks should be part of the quality control process for user-generated FDS input data. Second, an evaluation of whether each respective identified check could be automated and performed using a software application was made and, in case deemed possible, implemented in such an application. Third, a case study where the developed application was used to evaluate archived FDS input data from actual building projects was performed, in order to determine whether the application could be used to reduce the number of errors in input data and to determine which errors are most commonly

occurring.

Results have been presented in the form of two checklists covering controls which should be part of the quality control process for user-generated FDS input data; one qualitative checklist intended to be checked manually by the user and one quantitative checklist intended to be checked using the developed software application. Further, an actual application covering the quantitative checklist has been produced, evaluated in a case study, and published. Notably, the application was capable of identifying at least one error in all of the evaluated input files. The actual frequencies with which different errors occurred in the case study have also been presented. The ones most frequently occurring in the study were either of the kind where an intended parameter value was not correctly specified or closely related to the mesh. The errors concerned in the first group were the output data, the growth rate, the fractions of fuel mass converted to carbon monoxide or carbon dioxide, the heat of combustion, and the heat release rate not being specified as intended or, in the case of BBRAD scenarios, as required. The errors concerned in the second group were $D^* / \mathrm{d}x$ being less than 10, mesh groups not being entered from finest to coarsest, and mesh cells not being cubic. Due to the limited scope of the study, no general conclusions have been drawn in regard to which user related errors are most commonly occurring in FDS input data. However, while these results might not be generally representative, they could serve as an indication of certain areas which should be handled with care in producing actual FDS input data.

Finally, in regard to the main objectives, it has been concluded that the developed application indeed is a useful tool in discovering such errors, thereby also providing a means of reducing their occurrence.

Contents

Chapter 1

Introduction

1.1 Background

Today, CFD (Computational Fluid Dynamics) models constitute a common tool for simulating fires in the context of performing different sorts of risk assessments. For example, in Sweden such tools are applied to verify compliance with governmental requirements in the design of fire protection for buildings. A model which is often applied in these cases is FDS (Fire Dynamics Simulator) published by the American NIST (National Institute of Technology). Simulations of fires in FDS are based on user-generated input data in the form of text files which are crucial to the quality of the results obtained by running the simulations. However, generating such input data is a complex task where different types of errors may be made, placing high requirements on the skills of the user.

Currently, it is neither known to which extent user related errors in input data actually occur nor which types of errors are most commonly occurring. An apparent sign that errors do occur and that these are considered both hard and important to discover, is the fact that a number of documents and guidelines on the use of FDS have been published by a number of different stakeholders. Examples are BIV (2013), Briab (2012), Jakobsen et al. (2009), McGrattan, McDermott, Hostikka, and Floyd (2010b), and Nystedt and Frantzich (2011). To varying extents, the mentioned examples all provide recommendations on which input parameters to check in order to minimize the probability of errors and on which values should be applied for important input parameters. To name three examples, the quality control process should include checking that the rate of soot production complies to any governmental requirements or other recommendations, that the correct heat release rate ($\dot{Q}$) has been specified, and that the ratio $D^*/\,\mathrm{d}x$ falls within a recommended interval. Soot production is set explicitly in the input data and can be checked by locating and reading the correct parameter in the input file (McGrattan et al., 2010b). The heat release rate, however, is not set explicitly. In order to accurately check this parameter the fire area has to be calculated, based on its specified position in the input file, and multiplied by the specified heat release rate per unit area or some other defining metric (McGrattan et al., 2010b). To check $D^*/\,\mathrm{d}x$ the equation $D^* = (\frac{\dot{Q}}{\rho_\infty c_p T_\infty \sqrt{g}})^{2/5}$ has to be calculated (BIV, 2013; Nystedt & Frantzich, 2011). Due to the dependence on ρ_∞, c_p, and T_∞ the temperature specified in the input file has to be read and suitable values for the temperature dependent parameters be determined. Before the final calculation of the ratio can be performed, the value of $\mathrm{d}x$ has to be calculated based on coordinates specified in the input file.

In addition to the examples above, a number of other parameters should be checked according to the documents and guidelines. Hence, checking the input data is a comprehensive process requiring a non-negligible amount of time of the reviewer. Due to the required amount of time needed to perform these checks, there is a risk of such checks not always being performed—especially not for each simulated scenario in the same project—and that errors in input data may not be discovered. Even if a check in accordance with one of the guidelines is performed there is also a risk of mistakes being made in the control process, for example by miscalculating a parameter value or simply failing to perform a check of an important parameter value. Such undiscovered errors in input data present a problem in regard to the risk assessments for which they constitute the foundation, in that the assessments may be based on incorrect information. For example, while not related to the use of FDS, Lauridsen et al. (2001) have shown that noteworthy variations in the final results of different risk analysts exist and that the judgement of the analyst significantly affects the results of a risk assessment. Similar to what has been discussed here, uncertainties related to the detailed characteristics of an applied model as well as to constants and parameters were identified as one of a number of significant sources for uncertainty in the study.

Based on the above, an idea of further developing the method for performing quality controls of user-generated FDS input data and studying the occurrence of errors in such data was born. For these purposes a software application to automate parts of the quality control process could be developed, applied, and evaluated.

The automated parts would have to be quantifiable, in the sense that their fulfillment can be evaluated using a computer—i.e. by being expressed as criteria suitable for evaluation by applying arithmetical or logical operations. While this may be suitable for some requirements—e.g. the ones mentioned above—other ones may not so easily be expressed in such terms. For example, while FDS can be used to model most any fire scenario and predict almost any quantity of interest, the prediction might not be accurate due to limitations in the description of the fire physics or because of limited information about things such as the fuel and geometry (McGrattan, McDermott, Hostikka, & Floyd, 2010a). In actually determining whether the use of FDS is appropriate for a given application one should consider things such as the scenarios of interest, the predicted quantities, and the desired level of accuracy (McGrattan et al., 2010a). However, since these aspects are not easily expressed in terms other than qualitative, they are not well suited for the intended automated part of the quality control process. Instead, the control process would have to be separated into two parts. A manual qualitative part covering the aspects not deemed suitable, or possible, for automated control and a quantitative part performed by supplying input data to a software application.

While limited in scope, by applying the suggested application on a set of input files the relative occurrences of different errors could be evaluated and based on this information it would be possible to determine which parameters should be handled with most care in producing the input data to avoid errors arising. Such an application could also contribute to the further systematization and rationalization of the quality control process, with the expected effect of resulting in an increased number of discovered errors and an increased application of quality controls. Presuming that the application is correctly constructed, its application would be expected to result in a decreased probability of undiscovered errors in the quality control process. In a larger perspective, a better foundation for risk assessments—for which results of FDS simulations constitute the decision basis—would be expected.

1.2 Objectives and research questions

The project had two main objectives. First, to determine to which extent the quality control process for FDS input data could be systematized and rationalized by developing an automated software application, and second, to study the occurrence of user related errors in FDS input data and evaluate whether it could be reduced by applying the developed application.

To reach these objectives the following research questions have been formulated and studied:

1. Which checks should be part of the quality control process for user-generated FDS input data?
2. Which of the identified checks can be automated and performed using a software application?
3. Could the application of the developed software application result in a reduced number of errors in FDS input data?
4. Which user related errors are most commonly occurring in FDS input data?

1.3 Scope

Depending on the type of application, different recommendations for how to set the parameter values of the FDS input data and for which checks to include in the quality control process can apply. The checks and recommendations covered by the scope of this project have been limited to those which are suitable for the design of fire protection in buildings. Additionally, the scope has been limited to a Swedish perspective in regard to building code requirements and other recommendations.

Due to many of the available recommendations thus far only being adapted for FDS version 5 and due to differences in the structure of FDS input files between version 5 and version 6 only the former version has been included in the scope of the project.

In order to automate the checks included in the quality control process they have to be quantifiable. Identified checks have been translated to quantifiable and verifiable terms where possible. Checks for which this was not deemed possible have not been included in the software application.

All input data has been assumed correctly generated in terms of it being sufficient for starting an actual FDS simulation. Hence, no checks will be included for aspects of the input data which would prevent an FDS simulation from starting.

Both research question 3 and 4 relate to the occurrence of errors in FDS input data. The word error may be defined in a number of ways, each more or less appropriate for a given context. To avoid any confusion, in the context of this project, an error has been defined as not fulfilling the acceptance criteria in one of the checks identified in the answering of research question 1—i.e. checks which should be part of the quality control process for user-generated FDS input data.

The topic covered in this project concerns the particulars of how to generate input data for FDS. Because of this, any reader of this report has been assumed familiar with the FDS application and its related process for generating such input data. Anyone not familiar with these topics should study the McGrattan et al. (2010b) before continuing with the material presented here. To clearly indicate commands and parameters which constitute FDS input data, these will be set in a different font: `OBST`, `VENT`, etc.

1.4 Limitations

Some of the research questions are dependent on the amount of input data analyzed. Since such data could not simply be generated, but had to be collected from participating parties, the collected amount of data has been a limiting factor of the project.

Due to the sensitive nature of the results no participating companies and individuals have been named and the actual input data obtained has not been published. Instead, different users have only been referred to as A, B, C, etc. and no employer information has been presented.

Chapter 2

Method

2.1 Overview

The project was performed in three main steps:

1. To answer research question 1—i.e. determining which checks should be part of the quality control process for user-generated FDS input data—a literature review was performed.
2. To answer research question 2—i.e. determining which of the identified checks could be automated and performed using a software application—all items in the reproduced and compiled checklists presented in the literature review were evaluated one by one. All items deemed suitable for automated control were then implemented in a software application.
3. To answer research questions 3 and 4—i.e. determining if the application of the developed software application could result in a reduced number of errors in FDS input data and determining which user related errors are most commonly occurring in FDS input data—a case study was performed.

The applied methods for each step have been described in the subsections below.

2.2 Literature review

To identify relevant literature, three main categories were studied:

- Building code requirements.

- Recommendations from the FDS developer.
- Recommendations from other stakeholders.

Some of the identified sources provided complete checklists for what to include in the quality control process; these have been reproduced in section 3.1. For the sources which did not provide complete checklists such lists were compiled by summarizing all relevant recommendations found in each respective document. These have also been presented in section 3.1.

2.3 Evaluation of checklist items

For each item it was determined whether to develop either a full implementation, a limited implementation, or no implementation in the software application. An item has been considered fully implemented if the automated check covers all aspects of the item. If only parts of the item are covered it has instead been considered a limited implementation. The decision was based on the types of qualities covered by the check and wether a suitable arithmetical or logical operation for performing it could be constructed, but a full implementation was strived for in all cases. The type of implementation—along with clarifying remarks detailing the decisions, where deemed necessary—have been presented along with the checklists in section 3.1.

Two additional checklists were then constructed; one quantitative and one qualitative. The quantitative checklist contains all checklist items from the reviewed sources which have been implemented in the application, i.e. all items with a full implementation or the parts of an item which are covered by a limited implementation. The qualitative checklist contains all remaining checklist items, i.e. all items with no implementation and the parts of all items which are excluded in the limited implementations.

The actual application was developed using version 2.7 of the Python programming language (Python Software Foundation, 2015b). Python is a high-level general-purpose programming language—incorporating concepts such as modules, exceptions, dynamic typing, very high level dynamic data types, and classes—that can be applied to many different classes of problems (Python Software Foundation, 2015a). It was chosen for this particular application in large parts due to its comprehensive standard library, which for example provides tools for string processing, unit testing, and file system operations.

Two measures were taken to verify that correct results were provided by the application: manual evaluation of identified errors and unit tests. The manual evaluation consisted of double checking any identified errors by manually performing the arithmetical and/or logical operations necessary to perform the check. If no differences were identified the implementation was considered correct. Unit testing refers to a method for testing whether units of code are fit for actual use, which is common in software development. The actual tests are constructed by supplying a code unit with some input for which the correct output is known. When running the test, if the actual output differs from the correct output, the program will report this as an error.

2.4 Case study

The case study was performed by applying the developed application on archived FDS input data from actual projects, obtained by cooperating with consultants operating on the Swedish market. A number of users known by the author were contacted and asked to participate and input data was then gathered by collecting archived input files from projects which had been documented in some form of report. By studying the reports, information on the intended parameter values, such as peak heat release rate, growth rate, and soot production, were also gathered. Finally, the input files were evaluated by applying the developed application.

Due to the limited sample size it has not been possible to draw any general conclusions regarding the entire population of FDS users. This has been discussed at length in chapter 4.

Chapter 3

Results

3.1 Overview of recommendations and regulations

A literature review has been performed to identify which aspects of the FDS input data should be checked as part of the quality control process. A number of sources have been identified, which are presented in the following subsections. For each source, a table covering recommendations and/or requirements is presented along with information regarding the implementation of each item in the software application for automating the quality control process.

3.1.1 Building code requirements

In accordance with the Swedish building code BBR (Boverkets byggregler), the fire protection of buildings shall be designed, arranged, and verified by either simplified or analytical design (Boverket, 2014). Simplified design means fulfilling the detailed prescriptive requirements of the code and analytical design means fulfilling the performance based requirements. When applying analytical design the additional requirements of BBRAD (Boverkets allmänna råd om analytisk dimensionering av byggnaders brandskydd) should be fulfilled. BBRAD does not explicitly mention FDS, but requirements regarding the values of certain parameters used in fire calculations as well as acceptable levels in regard to the safety of evacuating persons are presented. Hence, if a simulation is intended to fulfill the requirements of BBRAD, a number of required parameter values can be derived from just a few basic conditions regarding the building and its intended use. These are:

- The fire scenario.
- The type of occupation.
- The presence of a sprinkler system.
- In case a sprinkler system is present, the activation time of the system.

Based of these parameters, the code specifies minimum values for the growth rate, heat release rate, heat of combustion, soot production, carbon monoxide production, and carbon dioxide production. Further, to actually determine whether a fire scenario is acceptable after simulating it, the layer height, visibility, heat flux, temperature, carbon monoxide concentration, carbon dioxide concentration, and oxygen concentration must all be evaluated. Due to these requirements, an important step in the quality control process should be to check whether a FDS input file is compliant with the building code. The requirements from BBRAD have been presented in table 3.1.

3.1.2 Developer recommendations

Fire Dynamics Simulator (Version 5) – User's Guide

The user's guide published by NIST (McGrattan et al., 2010b) does not provide any guidance on how to perform a quality control of input data or which checks should be included in such a control. However, a thorough description of how to write an input file is given. Further, a complete record of all available parameters is presented and for certain parameters instructions regarding suitable parameter values. The user's guide has been studied in its entirety and all relevant recommendations have been collected in table 3.2.

3.1.3 Other recommendations

Kvalitetsmanual för brandtekniska analyser vid svenska kärntekniska anläggningar

This manual by Nystedt and Frantzich (2011) is intended to be used in the context of CFD simulations in Swedish nuclear facilities. It contains a checklist separated into two parts, one to be performed before a simulation and the other to be performed after a simulation. These checks are of a general nature and have thus been included in this project. The checks recommended to be performed before running a simulation have been reproduced in table 3.3.

Table 3.1: *Checks to be performed by the user before running a simulation, in order to achieve compliance with Boverket (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None.*

No.	Instructions	Impl.			Remarks
		F	L	N	
1	The correct growth rate has been specified.	x			The correct value is dependent on the type of occupancy, as defined in BBRAD.
2	The correct heat release rate has been specified.	x			The correct value is dependent on the type of occupancy, as defined in BBRAD, and the activation time of the sprinkler system, if present.
3	The correct heat of combustion has been specified.	x			The correct value is dependent on the type of occupancy, as defined in BBRAD.
4	The correct soot production has been specified.	x			The correct value is dependent on the required fire scenario, as defined in BBRAD, and wether a working sprinkler system is present.
5	The correct carbon monoxide production has been specified.	x			The correct value is dependent on the required fire scenario, as defined in BBRAD, and wether a working sprinkler system is present.
6	The correct carbon dioxide production has been specified.	x			The correct value is dependent on the required fire scenario, as defined in BBRAD, and wether a working sprinkler system is present.

Table 3.1: *Checks to be performed by the user before running a simulation, in order to achieve compliance with Boverket (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
7	The necessary output data has been specified.	x			The necessary output data is specified in BBRAD.

Table 3.2: *Checks to be performed by the user before running a simulation, in accordance with McGrattan et al. (2010b). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None.*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
1	Mesh cells should resemble cubes—i.e. have equal length, width, and height.	x			
2	To avoid unnecessarily slow calculations the y and z dimensions of each mesh—i.e. the `JK` part in the `IJK` group of the `MESH` namelist—should be of the form $2^l3^m5^n$, where l,m, and n are integers.	x			
3	When using multiple meshes they should be entered from finest to coarsest in the input file since FDS gives precedence to a preceding mesh if meshes overlap.	x			
4	Mesh boundaries should not be put where critical action is expected, especially not near a fire, since the exchange of information across mesh boundaries is not as accurate as cell to cell exchanges within one mesh.		x		Not fully quantifiable. Critical action can be expected at or above the fire surface and mesh boundaries crossing fire surfaces are therefore deemed inappropriate. This has been implemented in the application, but all other aspects of this item must be checked manually.
5	Meshes should not overlap, since information only is exchanged at exterior boundaries.	x			

Table 3.2: *Checks to be performed by the user before running a simulation, in accordance with McGrattan et al. (2010b). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
6	It should be checked that each mesh can"see" a planar obstruction placed close to abutting meshes. If this is not the case, information will not be transferred properly between the meshes.			x	Whether an obstruction is visible or not in all intended meshes has to be checked in Smokeview after running a simulation. Since the item cannot be expressed in quantitative terms, it has not been implemented.
7	A sufficiently small grid spacing should be used.		x		No recommendation is given other than that the parameter should be determined by gradually refining the mesh until no appreciable differences can be seen in the results. The suggested method is not quantifiable and must be performed manually. However, as an indication of whether a sufficiently small grid spacing is used, checking whether $D^* / \mathrm{d}x$ is greater than 10 has been implemented.
8	Thin sheet obstructions work fine as flow barriers, but other features—expecially in terms of burning and blowing gas—are fragile. Hence, for full functionality, obstructions should be at least one mesh cell thick.			x	Whether an obstruction is a flow barrier or not cannot be determined numerically. Therefore, the item has not been implemented.

Table 3.2: *Checks to be performed by the user before running a simulation, in accordance with McGrattan et al. (2010b). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
9	To ensure that holes fully penetrate an obstruction they should extend outside the obstruction.			x	Determining which obstructions a hole is intended to penetrate cannot be done numerically. Asking the user to specify all holes and which obstructions they are intended to penetrate would be a complicated task and probably not add much in terms of rationalization compared to just checking out the model in Smokeview. This item has therefore not been implemented.
10	Only one `VENT` group may be specified for any given wall cell. If overlapping `VENT` groups exist only the one listed first in the input file will be applied. However, the second `VENT` is not rejected entirely—only where there is overlap. The presence of overlapping `VENT` groups will be noted by FDS when processing the input file.	x			
11	The `MIRROR` boundary condition should not be used along the centerline of a turbulent fire plume.	x			In the context of building fires, all fire plumes have been assumed turbulent.
12	If `BURN_AWAY` is prescribed, the `SURF` should be applied to the entire object, not just a face of the object.	x			

Table 3.2: *Checks to be performed by the user before running a simulation, in accordance with McGrattan et al. (2010b). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
13	On `SURF` lines where `HRRPUA` or solid phase reaction parameters are specified, no velocity should be prescribed. The combustible gases are ejected at a velocity computed by FDS.	x			
14	Finite-rate reactions should only be invoked when FDS is running in DNS (Direct Numerical Simulation) mode.	x			
15	Beam detectors cannot span more than one mesh. Instead, the beam detector path should be broken into multiple `DEVC` lines, one for each mesh that the path crosses				
16	`VENT` groups with the `SURF_ID` parameter set to `'MIRROR'` or `'OPEN'` should not be activated or deactivated during the simulation. Instead, the preferred method of controlling the flow in these cases is to create obstructions—which can be activated or deactivated—in front of them.	x			
17	Any output data that is to be saved must be specified in the input file before the simulation is started.		x		Only quantifiable when the intended output data is known. A limited implementation has been achieved by providing the option to specify any of the output quantities required by BBRAD as intended output data.

Table 3.2: *Checks to be performed by the user before running a simulation, in accordance with McGrattan et al. (2010b). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
18	Non-pointwise devices—i.e. planar or volumetric—should not cross mesh boundaries.	x			
19	The `MASS_EXTINCTION_COEFFICIENT` parameter of the `REAC` namelist group is should be set to 8 700 m^2/kg ± 1 100 m^2/kg for most flaming fuels.	x			
20	When measuring visibility levels, the `VISIBILITY_FACTOR` parameter of the `REAC` namelist group is should be set to 3 for light-reflecting signs and 8 for light-emitting signs.		x		Only quantifiable if the type of signs are known. Conservatively, a value of 3 will be applied unless otherwise is stated.
21	Do not specify both `HRRPUA` and `MLRPUA` in the same `SURF` group.	x			

Table 3.3: *Checks to be performed by the user before running a simulation, in accordance with Nystedt and Frantzich (2011). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None.*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
1	Check that Q^* is in the magnitude of 0.3 to 2.5.	x			
2	Check that the resolution $D^* / \mathrm{d}x$ of the calculation is in the magnitude of 10 to 20. Initial simulations can be made using a ratio of about 5 to roughly examine certain variables.	x			
3	Check that meshes are connected in an appropriate way and that objects spanning more than one mesh are visible in them all. Check that there are no mesh boundaries in areas with much flow.		x		Not fully quantifiable. Meshes should not overlap and be input from finest to coarsest, which has been implemented in the application. All other aspects of this item must be checked manually.
4	Check combustion and material properties.		x		Only quantifiable when the intended values are known. The implementation has been limited to the growth rate, heat release rate, heat of combustion, soot production, carbon monoxide production, and carbon dioxide production. All other aspects of this item must be checked manually.
5	Check that the radiation model is used and that enclosing surfaces have correctly defined properties for heat transfer.		x		Not fully quantifiable. Use of the radiation model has been implemented, but the other aspects must be checked manually.

CFD-beräkningar med FDS

BIV, the Swedish chapter of the SFPE (Society of Fire Protection Engineers), has published guidelines for how to perform CFD calculations using FDS (BIV, 2013). The guidelines cover version 5.5.3 of the model and are intended to provide support when applying it to analytical design in accordance with the Swedish building code, specifically BBRAD 1 (Boverket, 2011). The stated reason for publishing the guidelines is that while analytical design is permitted by the building code, no guidance on how to correctly perform the underlying analyses is provided. Hence, the overall purpose of the document is to act as support in applying CFD analyses to perform smoke filling calculations and to achieve an acceptable level of quality in the calculations. The guidelines are limited to CFD simulations of the early stages of fire and which are intended to evaluate the possibility of safe egress.

Of most interest in this context is appendix A of the guideline, which covers quality assurance and consists of two comprehensive checklists. Appendix A.1 presents a checklist intended for the person who has produced the FDS input data and appendix A.2 provides a checklist intended for any other person performing a review of the FDS input data. Each checklist is divided into three subsections: items regarding the overall model, items to check before running a simulation, and items to check after running a simulation. Of relevance here are the checklist items intended to be checked before running a simulation, which are basically identical in the two checklists. These items have been numbered and translated and are presented in table 3.4.

Vägledning - brandgasfyllnad

This document, published by Briab Brand & Riskingenjörerna AB in 2012 (Briab, 2012), predates the guidelines described in section 3.1.3. However, the more recent document by BIV is practically identical with this one, which for this reason will not be presented as a separate checklist. Instead, see table 3.4.

CFD Best Practice

The Danish best practice document (Jakobsen et al., 2009) predates all the other documents in the literature review. While it does not contain a checklist, all relevant recommendations have been collected in table 3.5.

Table 3.4: *Checks to be performed by the user before running a simulation, in accordance with BIV (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None.*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
1	Check that the purpose and objective of the analysis are clearly formulated and that the computational model is constructed based on this.			x	Not quantifiable, must be checked manually.
2	Check that the chosen fire scenarios are representative for the objective of the analysis (if applicable, check against the required fire scenarios of BBRAD 1).			x	Not quantifiable, must be checked manually.
3	Present any simplifications in FDS that might have a significant effect on the result and whether these are acceptable or not based on the chosen fire scenarios.			x	Not quantifiable, must be checked manually.
4	Check that the grid resolution $D^* / \mathrm{d}x$ is between 10 and 20.	x			
5	Check that the mesh division has been performed appropriately and that mesh boundaries are not placed in areas with a high flow velocity. At mesh boundaries with different grid sizes, it is important for flow to be mainly perpendicular to the mesh boundary.		x		Not fully quantifiable. Mesh boundaries crossing fire surfaces is deemed inappropriate and has been implemented in the application. All other aspects of this item must be checked manually.
6	Check that cells are cubic (ratio 1:1:1).	x			

Table 3.4: *Checks to be performed by the user before running a simulation, in accordance with BIV (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	Instructions	Impl.			Remarks
		F	L	N	
7	Check that Q^* is between 0.3 and 2.5.	x			
8	Check that the placement of the fire is appropriate based on the activities performed in the building.			x	Not quantifiable, must be checked manually.
9	Check that the properties of the fire, materials, and surfaces have been defined as intended.		x		Only quantifiable when the intended values are known. The implementation has been limited to the growth rate, heat release rate, heat of combustion, soot production, carbon monoxide production, and carbon dioxide production. All other aspects of this item must be checked manually.
10	Check that the fire is placed on an elevation.			x	Whether an object is elevated is relative and not quantifiable unless an intended base level is specified. The item has not been implemented and must be checked manually.
11	If `SPREAD_RATE` is used, check that the velocity has been calculated based on the correct `HRRPUA`.	x			Implicitly, it is understood that a t-squared fire growth is intended.
12	If `SPREAD_RATE` is used, check that `XYZ` for `SPREAD_RATE` is correct.	x			It will be assumed that being correct means specifying the center point of the fire surface.

Table 3.4: *Checks to be performed by the user before running a simulation, in accordance with BIV (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
13	Check that the correct soot production, HRR, and heat of combustion have been specified for the fire based on the current scenario.	x			Only quantifiable when the intended values are known.
14	Check that material properties for walls and ceilings have been specified correctly. If no information is available, perform a parameter study using inert and adiabatic surfaces.			x	Only quantifiable when the intended values are known. Means of providing such intentions have not been implemented and the item must be checked manually.
15	Check that the building dimensions are correct and that the correct drawing scale has been used.			x	Not quantifiable, must be checked manually.
16	Check that the volume of the building has been preserved.			x	Possibly quantifiable but deemed too complex for this context, must be checked manually.
17	Evaluate whether geometrical simplifications (the exclusion of stairs, openings, etc.) in the model are reasonable and do not affect the results in such a way as to affect the conclusions of the analysis.			x	Not quantifiable, must be checked manually.

Table 3.4: *Checks to be performed by the user before running a simulation, in accordance with BIV (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
18	Evaluate if natural leakages have to be observed for the scenario. If that is the case, check that leakages have been specified in a correct way and with a reasonable area.			x	Not quantifiable, must be checked manually.
19	Check that openings for pressure relief have been placed at mesh boundaries and that the shortest distance between openings in the building and the mesh boundary is 0.5 x the hydraulic diameter of the largest opening of the calculation model. (Hydraulic diameter = 4 x The area of the opening / The circumference of the opening)		x		The implementation has been limited to checking that at least one open vent has been placed on the mesh boundary. All other aspects must be checked manually.
20	Perform a general review of the model in Smokeview.			x	Not quantifiable, must be checked manually.
21	Check that smoke ventilation, if used, is placed appropriately (size of hatches/distance between hatches/placement).			x	Not quantifiable, must be checked manually.
22	Check that all control functions have been defined and connected correctly.			x	Not quantifiable, must be checked manually.
23	Check that sufficient output data has been specified to achieve the objective of the simulation.		x		Only quantifiable when the intended output data is known.

Table 3.4: *Checks to be performed by the user before running a simulation, in accordance with BIV (2013). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None (cont.).*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
24	Check the placement of point-wise measurements and that “devices” (`DEVC`) on solid surfaces “point” in the correct direction.			x	The correct direction cannot has to be indicated by the user and is not fully quantifiable. The item has not been implented.
25	Evaluate whether output data needs to be logged more often than pre-defined in FDS.			x	Not quantifiable, must be checked manually.
26	Review the FDS code in the input file. Especially check the fire, active systems, control functions, and any commands that might have unintentionally been included in the code.			x	Not quantifiable, must be checked manually.

Table 3.5: *Checks to be performed by the user before running a simulation, in accordance with Jakobsen et al. (2009). F, L, and N indicate the level of implementation in the developed software application, i.e. Full, Limited, or None.*

No.	**Instructions**	**Impl.**			**Remarks**
		F	L	N	
1	The Smagorinsky constant should equal 0.2.	x			
2	Q^* should be between 0.3 and 2.5.	x			
3	$D^*/\,\mathrm{d}x$ should be between 4 and 16, but is recommended to be greater than 10.	x			
4	The `HRRPUA` parameter should be between 500 kW/m^2 and 2500 kW/m^2.			x	Though quantification is possible, this check is not considered appropriate. Many types of fires may exceed 2500 kW/m^2, see (Karlsson & Quintiere, 2000, table 3.6), and this check may also contradict the evaluation of Q^*, which has been implemented.
5	The `VISIBILITY_FACTOR` should be set to 3.	x			

3.2 Quality assurance checklist

For each checklist item in the different checklists presented in section 3.1, an assessment of the possibility of quantifying the check has also been presented. All items which were deemed possible to quantify, and hence possible to evaluate by using an automated application, have been collected in a *quantitative* checklist and all remaining items have been collected in a *qualitative* checklist. For each item in the unified checklists, references to the source items in the old checklists have been included. All checklist items in the quantitative checklist have been integrated into the application, but for a complete check of any input file—i.e. a check matching all the evaluated checklists in section 3.1—both checklists have to be evaluated.

The quantitative checklist is presented in table 3.6 and the qualitative checklist is presented in table 3.7.

Table 3.6: *Quantitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5.*

No.	**Instructions**	**Source**				
		A	B	C	D	E
1	Mesh cells are cubic.		1		6	
2	The dimensions in the y- and z-directions can be written on the form $2^l 3^m 5^n$.		2			
3	Mesh groups are entered from finest to coarsest.		3	3		
4	Mesh boundaries do not cross a fire.		4		5	
5	Meshes do not overlap.		5	3		
6	$D^* / \mathrm{d}x$ is greater than 10.		7	2	4	3
7	The `MASS_EXTINCTION_COEFFICIENT` parameter of the `REAC` group should be set to between $7600\,\mathrm{m^2\,kg^{-1}}$ to $9800\,\mathrm{m^2\,kg^{-1}}$ for most flaming fuels.		19	4		
8	The `VISIBILITY_FACTOR` parameter of the `REAC` group should be set to 3 when modeling light-reflecting signs and to 8 when modeling light-emitting signs.		20	4		5
9	$\dot{Q}^*$ is between 0.3 and 2.5.			1	7	2

Table 3.6: *Quantitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5 (cont.).*

No.	**Instructions**	**Source**				
		A	B	C	D	E
10	The heat release rate has been specified as intended.	2		4	9, 13	
11	The growth rate has been specified as intended.	1		4	9, 11, 13	
12	The heat of combustion has been specified as intended.	3		4	9, 13	
13	The fraction of fuel mass converted to soot has been specified as intended.	4		4	9, 13	
14	The fraction of fuel mass converted to carbon monoxide has been specified as intended.	5		4	9, 13	
15	The fraction of fuel mass converted to carbon dioxide has been specified as intended.	6		4	9, 13	
16	`SURF` groups with the `BURN_AWAY` parameter set to `.TRUE.` cover all surfaces of any `OBST` group on which they are applied.		12			
17	No velocity is specified for `SURF` groups where the `HRRPUA` or `MLRPUA` parameters are specified or where solid phase reaction parameters are specified.		13			
18	Finite-rate reactions have not been specified, unless DNS mode is used.		14	4		
19	If `SPREAD_RATE` is specified in any `SURF` or `VENT` groups, the `XYZ` parameter is centered on the fire surface.				12	
20	Both `HRRPUA` and `MLRPUA` are not specified in the same `SURF` group.		21			
21	Output data has been specified.				23	
22	The necessary output data has been specified.	7	17		23	
23	`VENT` groups do not overlap.		10			

Table 3.6: *Quantitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5 (cont.).*

No.	**Instructions**	**Source**				
		A	B	C	D	E
24	The `MIRROR` boundary condition is not used along the centerline of a fire obstruction or fire vent.		11			
25	`VENT` groups with the parameter `SURF_ID` set to `'MIRROR'` or `'OPEN'` are not activated or deactivated during the simulation.		16			
26	Non-pointwise devices do not cross mesh boundaries.		15, 18			
27	At least one `VENT` group with the `SURF_ID` parameter set to `'OPEN'` has been placed on the mesh boundary.				19	
28	The radiation model is used.			5		
29	The Smagorinsky constant equals 0.20.					1

Table 3.7: *Qualitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5.*

No.	**Instructions**	**Source**				
		A	B	C	D	E
1	The purpose and objective of the analysis have been clearly formulated and the computational model has been constructed with these in mind.				1	
2	The chosen fire scenarios are representative for the objective of the analysis (if applicable, check against the required fire scenarios of BBRAD 1).				2	
3	Simplifications in FDS that might have a significant effect on the result have been presented and are acceptable based on the chosen fire scenarios.				3	
4	Properties of the fire, materials, and surfaces have been specified as intended.			4	9	

Table 3.7: *Qualitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5 (cont.).*

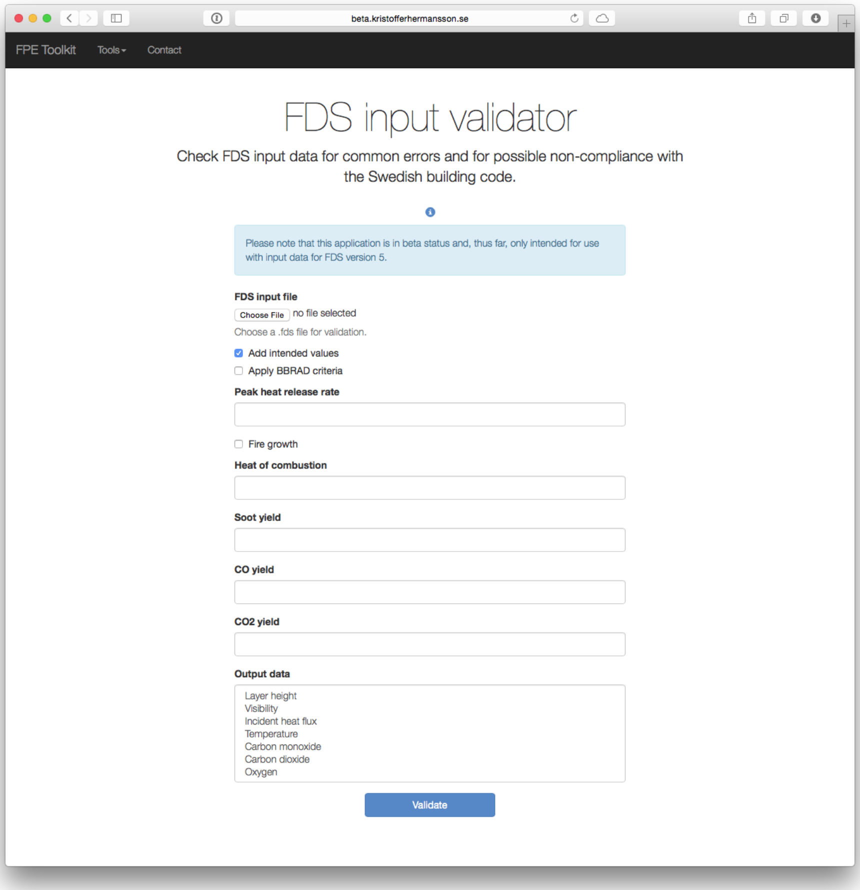
No.	**Instructions**	**Source**				
		A	B	C	D	E
5	The building dimensions are correct and the correct drawing scale has been used.				15	
6	The volume of the building has been preserved.				16	
7	Geometrical simplifications (the exclusion of stairs, openings, etc.) in the model are reasonable and not expected to affect the results in such a way as to affect the conclusions of the analysis.				17	
8	Where necessary, natural leakages have been observed and have been specified in a correct way and with a reasonable area.				18	
9	The shortest distance between openings in the building and the mesh boundary is 0.5 x the hydraulic diameter of the largest opening of the calculation model.				19	
10	A general review of the model in Smokeview has been performed.				20	
11	When present, smoke ventilation installations have been placed appropriately (size of hatches/distance between hatches/placement).				21	
12	All control functions have been defined and connected correctly.				22	
13	Mesh divisions have been made appropriately.		4	3	5	
14	Mesh boundaries are not placed in areas where high flow velocities are expected.		4	3	5	
15	At boundaries between meshes with different cell sizes, the flow is mainly expected to be perpendicular to the mesh boundary.				5	
16	The placement of the fire is appropriate based on the activities performed in the building.				8	
17	The fire is placed on an elevation.				10	

Table 3.7: *Qualitative checks to be performed by the user before running a simulation. The sources and their numbering refer to the checklists presented in section 3.1, A: table 3.1, B: table 3.2, C: table 3.3, D: table 3.4, E: table 3.5 (cont.).*

No.	**Instructions**	**Source**				
		A	B	C	D	E
18	Sufficient output data, in order to achieve the objective of the simulation, has been specified.				23	
19	Point-wise measurement devices have been placed as intended.				24	
20	Whether output data needs to be logged more often than pre-defined in FDS has been evaluated.				25	
21	Objects spanning more than one mesh are visible in all the affected meshes.		6	3		
22	Heat conduction properties have been specified correctly for enclosing surfaces.			5	14	
23	Obstructions are at least one mesh cell thick.		8			
24	Holes extend outside the obstructions they are supposed to penetrate.		9			
25	Point-wise measurement devices have been placed correctly and devices on solid surfaces point in the correct direction.				24	
26	Check that material properties for walls and ceilings have been specified correctly. If no information is available, perform a parameter study using inert and adiabatic surfaces.				14	
27	Review the FDS code in the input file. Especially check the fire, active systems, control functions, and any commands that might have unintentionally been included in the code.				26	

3.3 Software application

The developed application is available at http://beta.kristofferhermansson.se/fds-validator. Detailed descriptions of how the items of the quantitative checklist have been calculated and checked have been presented in appendix A. One example of how the checks have been implemented in code is presented in appendix A.1 but the full source code, which consists of over 4000 lines of code and would add an additional 80 pages to this document, has not been included. As noted in table 3.6 in section 3.2, several of the checklist items require an intended value to be known, in order to perform an automated check. To achieve this, the application accepts two inputs, an FDS input file and an optional intent specification. In the web version mentioned above, an FDS file from the local computer can be chosen and the intent specified using a simple form, see fig. 3.1. After submitting the data, it is evaluated and presented in the form of a checklist where all valid checklist items have been checked off and any errors are reported, see fig. 3.2.



FDS input validator

Check FDS input data for common errors and for possible non-compliance with the Swedish building code.

Please note that this application is in beta status and, thus far, only intended for use with input data for FDS version 5.

FDS input file
no file selected
Choose a .fds file for validation.

☑ Add intended values
☐ Apply BBRAD criteria

Peak heat release rate

☐ Fire growth

Heat of combustion

Soot yield

CO yield

CO2 yield

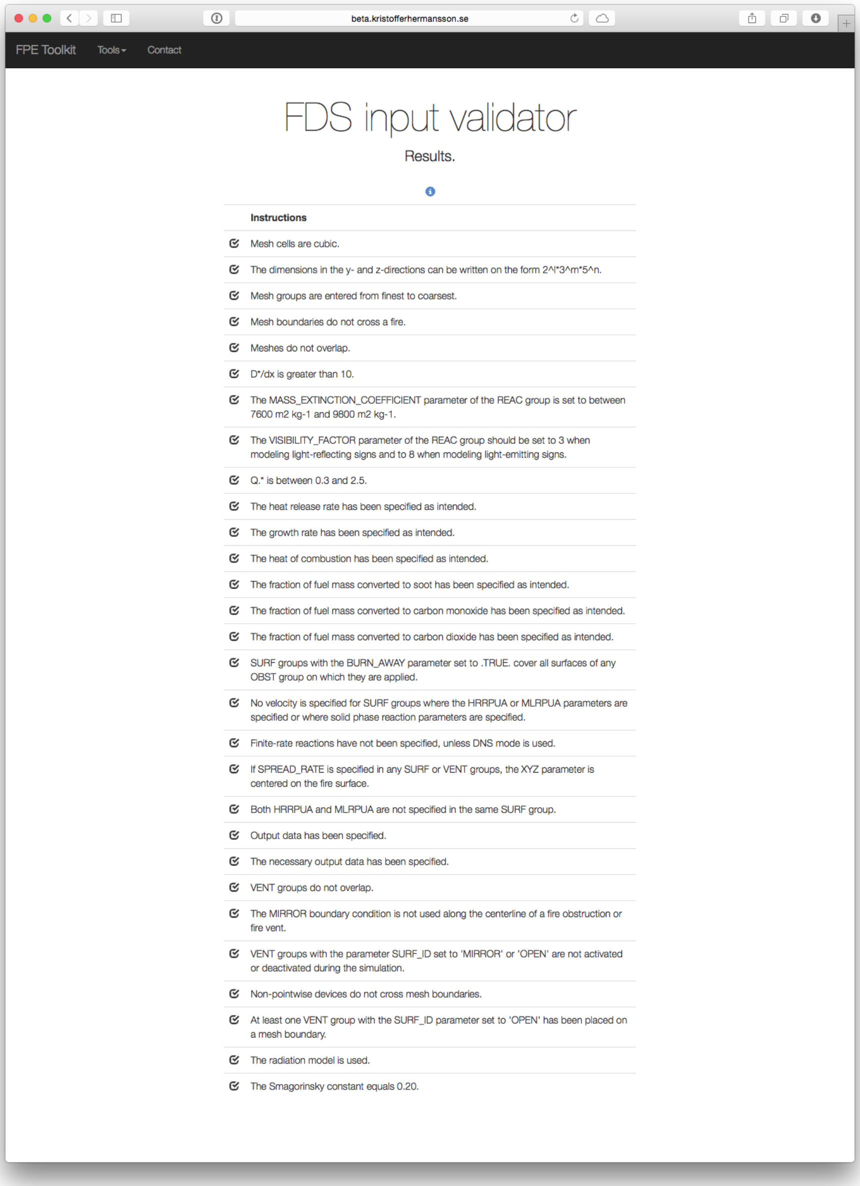
Output data
Layer height
Visibility
Incident heat flux
Temperature
Carbon monoxide
Carbon dioxide
Oxygen

Figure 3.1: *Screenshot of the application form.*

3.4 Evaluation of input data

A total of eighty-four files from ten different users have been evaluated using the developed application and a total of two hundred seventy-seven errors were detected. Errors were detected in all files, with the total number of errors in each file ranging from one to eight. A complete summary of all evaluated input files and the detected errors has been presented in appendix B.

The total number of occurrences for each checklist item has been summarized in table 3.8. Some errors were not identified in any of the evaluated input files, these have not been included in the table.



FDS input validator

Results.

Instructions

☑ Mesh cells are cubic.

☑ The dimensions in the y- and z-directions can be written on the form 2^l*3^m*5^n.

☑ Mesh groups are entered from finest to coarsest.

☑ Mesh boundaries do not cross a fire.

☑ Meshes do not overlap.

☑ D*/dx is greater than 10.

☑ The MASS_EXTINCTION_COEFFICIENT parameter of the REAC group is set to between 7600 m2 kg-1 and 9800 m2 kg-1.

☑ The VISIBILITY_FACTOR parameter of the REAC group should be set to 3 when modeling light-reflecting signs and to 8 when modeling light-emitting signs.

☑ Q.* is between 0.3 and 2.5.

☑ The heat release rate has been specified as intended.

☑ The growth rate has been specified as intended.

☑ The heat of combustion has been specified as intended.

☑ The fraction of fuel mass converted to soot has been specified as intended.

☑ The fraction of fuel mass converted to carbon monoxide has been specified as intended.

☑ The fraction of fuel mass converted to carbon dioxide has been specified as intended.

☑ SURF groups with the BURN_AWAY parameter set to .TRUE. cover all surfaces of any OBST group on which they are applied.

☑ No velocity is specified for SURF groups where the HRRPUA or MLRPUA parameters are specified or where solid phase reaction parameters are specified.

☑ Finite-rate reactions have not been specified, unless DNS mode is used.

☑ If SPREAD_RATE is specified in any SURF or VENT groups, the XYZ parameter is centered on the fire surface.

☑ Both HRRPUA and MLRPUA are not specified in the same SURF group.

☑ Output data has been specified.

☑ The necessary output data has been specified.

☑ VENT groups do not overlap.

☑ The MIRROR boundary condition is not used along the centerline of a fire obstruction or fire vent.

☑ VENT groups with the parameter SURF_ID set to 'MIRROR' or 'OPEN' are not activated or deactivated during the simulation.

☑ Non-pointwise devices do not cross mesh boundaries.

☑ At least one VENT group with the SURF_ID parameter set to 'OPEN' has been placed on a mesh boundary.

☑ The radiation model is used.

☑ The Smagorinsky constant equals 0.20.

Figure 3.2: *Screenshot of the application results.*

Table 3.8: *Identified occurrences of each error in the input data, the frequency of each error, the fraction of users committing each error, the corresponding item number in table 3.6, and the actual instructions of the item.*

Occurrences	**Frequency**	**Frac. users**	**Item no.**	**Instructions**
70	0.83	0.8	22	The necessary output data has been specified.
47	0.56	0.8	6	$D^* / \mathrm{d}x$ is greater than 10.
30	0.36	0.6	3	Mesh groups are entered from finest to coarsest.
21	0.25	0.7	11	The growth rate has been specified as intended.
21	0.25	0.4	14	The fraction of fuel mass converted to carbon monoxide has been specified as intended.
19	0.23	0.3	15	The fraction of fuel mass converted to carbon dioxide has been specified as intended.
18	0.21	0.4	12	The heat of combustion has been specified as intended.
18	0.21	0.3	1	Mesh cells are cubic.
12	0.14	0.4	10	The heat release rate has been specified as intended.
5	0.06	0.2	27	At least one `VENT` group with the `SURF_ID` parameter set to `'OPEN'` has been placed on the mesh boundary.
5	0.06	0.1	19	If `SPREAD_RATE` is specified in any `SURF` or `VENT` groups, the `XYZ` parameter is centered on the fire surface.
5	0.06	0.2	9	$\dot{Q}^*$ is between 0.3 and 2.5.
3	0.04	0.2	2	The dimensions in the y- and z-directions can be written on the form $2^l 3^m 5^n$.
3	0.04	0.1	23	`VENT` groups do not overlap.

Chapter 4

Discussion

4.1 General

First and foremost, it should be noted that while this project has mainly focused on the quantitative parts of the quality assurance process, the more qualitative aspects must not be forgotten. In many cases it could rather be argued that these aspects—e.g. whether applying FDS to a given problem is actually appropriate or whether the geometrical simplifications in a model are reasonable—are even more important than whether some parameter value is off by a decimal point. This means that even when applying the developed application, the quality control cannot be considered complete without manually controlling the items of the accompanying qualitative checklist presented in table 3.7 in section 3.2.

It should also be noted that some of the materials covered by the literature review, and which the presented checklists are based upon, were written in Swedish. When cited or reproduced in this report, these source materials have been interpreted and translated. This process might have introduced deviations from the original statements, e.g. regarding nuance and tone in the language. In order to avoid any interpretation or translation errors outside of this project, the source materials should be studied directly.

4.2 Method and results

4.2.1 Research question 1

Which checks should be part of the quality control process for user-generated FDS input data?

The answer to this question has been provided in the form of two checklists presented in section 3.2. The checklists were derived from a number of documents relevant to the context of this project and are believed represent the current state of the art in the Swedish FDS community. However, the possible existence of some other recommendations which perhaps should have been considered cannot be discarded. For example, some recommendations present in the guidelines which were included in the literature review might have been missed and the existence of other relevant guidelines might have been unnoticed.

In performing the literature review, any recommendations applicable to the design of fire protection in buildings were included in the compiled or reproduced checklists. However, no further consideration was given to the appropriateness of the actual recommendations. In performing the case study, further discussed in section 4.2.3 and section 4.2.4 below, the actual occurrence of the different errors covered by the developed application was studied. As evident by the results, see section 3.4, a number of the recommendations were frequently not adhered to in the user generated input files. This begs the question, are these errors an indication of quality problems associated with the use of FDS or are the suggested recommendations perhaps not suitable for the intended application? Were the latter to be true, these checks should, in fact, not be part of the quality control process. In regard to this, a few items were considered notable and have been given special consideration below.

The single most frequent error—necessary output data not being specified—was present in 83 % of the evaluated input files and for the evaluated files which were supposed to comply with one of the BBRAD scenarios, all of the necessary output quantities was not specified in a single case. What constitutes necessary output data for a given simulation depends on the purpose of the simulation and in order for this item to be checked, the user had to supply an intention containing the necessary output data parameters. This was done either explicitly, by specifying the parameters, or implicitly, by specifying a BBRAD scenario. For all BBRAD scenarios, the implementation of this particular item checks that either the `LAYER HEIGHT` or `VISIBILITY` quantities have been specified and that the `INCIDENT HEAT FLUX`, `TEMPERATURE`, `CARBON MONOXIDE`, `CARBON DIOXIDE`, and `OXYGEN` quantities have been specified. The reason for checking the presence of these specific parameters was that BBRAD requires evacuation to be finished before any of them reach certain specified levels. Hence, if the quantities are not included in the output data, it would not be possible to determine if all of the requirements of BBRAD are fulfilled. A possible explanation as to why this particular error was so frequent, based purely on personal experience, might be that out of all the stated quantities, the levels for temperature and visibility in practice appear to always reach the levels specified in BBRAD first. True or not, the possibility of this conception being widely held could explain why the error was so frequent in the input data. Still, its inclusion in the quality control process has been considered purposeful.

The second most frequent error—$D^* / \mathrm{d}x$ being less than 10—was present in 56 % of the evaluated input files. Due to the ratio being so closely related to the grid resolution and due to the importance of that particular parameter, the frequent occurrence of this error might be cause for concern. However, before jumping to any conclusions the actual implication of applying a ratio less than 10 must also be considered. If a value of 10 is deemed fully acceptable, does a value of 9 really justify calling a simulation erroneous, or should more weight possibly be given to the judgement of the individual user? In regard to grid spacing, the FDS user's guide (McGrattan et al., 2010b) states that determining a suitable size is dependent on the purpose of a given simulation and not easy to do. The suggested method of accomplishing it is to start with a coarse mesh and gradually refine it until no appreciable differences in the results can be seen. While $D^* / \mathrm{d}x$ is mentioned as a measure of how well the flow-field of a buoyant plume is resolved and might be a simple way of evaluating the resolution, it should not be considered the sole criterion for determining whether a sufficient resolution has been applied. That said, its inclusion in the quality control process has still been considered purposeful and the high frequency of this error is worrisome.

The third most frequent error—mesh groups not being entered from finest to coarsest—was present in 36 % of the evaluated input files. The recommendation was derived from the following:

> *If more than one mesh is used, there should be a `MESH` line for each. The order in which these lines are entered in the input file matters. In general, the meshes should be entered from finest to coarsest. FDS assumes that a mesh listed first in the input file has precedence over a mesh listed second if the two meshes overlap. (McGrattan et al., 2010b, p. 30)*

The recommendation is vague and thus hard to interpret, since no further information than it being applicable *in general* is given. While it is stated that FDS will give precedence to a mesh listed before another, no information of what this actually means or if this is the only time where actual problems may occur is given. So, while this error was frequent in the evaluated data the actual implication on the quality of the results is not well known. On a similar note, another frequent error for which the actual motivation and the implication on the results is not well known, is mesh cells not being cubic. While the actual effect of both these errors is somewhat unclear, their inclusion in the quality control process has still been considered purposeful due to them being clearly stated by the FDS developer (McGrattan et al., 2010b).

On a final note, while the chosen criteria in some cases may seem overly strict in the context of the evaluation performed in this project, do note that the intended use of the application is to systematize and rationalize the control process in actual risk assessments. In such cases the reporting of errors which could be too strict for a given application will not cause any problems in the design process. While possibly not relevant, they will serve as a reminder in the odd case where the opposite is true and otherwise be easily discarded by the user. For example, while the user may have gradually refined the mesh and determined that a given cell size is acceptable, the application will still produce an error if $D^* / \mathrm{d}x$ is less than 10. Actually using such a value is should then not be considered erroneous, and the fact that the application will report it as such can simply be discarded.

4.2.2 Research question 2

Which of the identified checks can be automated and performed using a software application?

The answer to this question has been presented in the form of a quantitative checklist and the accompanying software application in which it has been implemented, see section 3.2 and section 3.3. However, while the stated question implies a notion of certainty no such claim has been made regarding the provided answer. As stated in section 2.3, the decisions regarding which checks to implement were based on the types of qualities covered by each check and wether a suitable arithmetical or logical operation for performing it could be constructed. While certainly strived for, e.g. by including the motivations for discarding certain checklist items in section 3.1 and presenting the steps of the quantification process in appendix A, it cannot be claimed that the applied method was completely objective. In other terms, it cannot be said that no checks other than the ones implemented would ever be possible to implement and that the actual implementations could not be constructed in some other way than the one used here. Rather, the implemented checks must be seen, in some ways at least, as a limited by the capability and judgement of the author.

4.2.3 Research question 3

Could the application of the developed software application result in a reduced number of errors in FDS input data?

Even though the number of users participating in the case study was rather limited, the answer to this question would appear to be a simple yes. Errors were found in all of the evaluated files, and while the studied population perhaps might not be representative for the general population of FDS users on the Swedish market, it does represent a subsection of it. Hence, any errors present in the evaluated data that could have been discovered using the application would constitute a reduction as stated in the research question. The fact that such errors were found in all evaluated input files would then only serve to further prove the potential usefulness of the actual application. However, to complicate matters a bit, some of the evaluated input files were generated before some of the recommendations were published. Hence, the fact that errors in these files were detected using the application does not mean that these would have been undetected if the recommendations had been available at the time of production.

In answering the question, the quality of the actual application also becomes an important factor. To assure the validity of the application two main measures have been taken. First, during development unit tests were programmed where for complex functions. These unit tests were used to check that the application functions worked as intended for a defined set of input data. For each check where tests were applied, one or more different sets of input data were supplied to the check algorithm and the correct output defined. If the actual output did not match that defined as correct, an error was presented. All these different tests were then collected in test source code files. When the application source code was later modified, the full test suite could easily be run to assure that the modification did not lead to any of the functions breaking and returning invalid output. Second, all errors detected by the application in the case study have been manually verified. This, of course, only assures that the application is valid in regard to the errors actually detected in the case study. Undetected errors which should have been detected by the application may still be present.

While the focus of this project has been fire protection in buildings, and the application adapted for this purpose, the same method of evaluating FDS input data could be useful for other applications as well. In order to achieve this, different sets of checklists for different types of applications would have to be developed and implemented, much in the same way as has here been done for the simulation of fires in buildings. The same can also be said in regard to other versions of FDS, slight modifications to the source code and added implementations adapted would enable the application on inout data generated for this newer version.

4.2.4 Research question 4

Which user related errors are most commonly occurring in FDS input data?

The frequency of the different errors which are covered by the application, in the studied input data, has been presented in table 3.8 in section 3.4. This means that the checks included in the application—i.e. the qualitative checks listed in table 3.7 in section 3.2—have not been considered. Further, due to the limited sample size and to the fact that no randomized selection was made, the actual frequencies presented might not reflect those of the general population of FDS users. The reason for not performing such a study here, but instead a more limited case study, boils down to scope. The main objectives, as stated in section 1.2, were to systematize and rationalize the quality control process for FDS input data by developing the automated software application and to evaluate whether its application could be expected to lead to a reduction in user related errors in such data. In that context, this final research question was limited to a case study, which, even though not general, could give some indication as to what is the correct answer.

That aside, of particular note in the results of the case study is the fact that several of the more frequently occurring errors are of the kind where an intended parameter value is not correctly specified. In fact, out of the nine errors which are each present in more than 10 % of files, six are of this particular kind. The errors concerned are the output data, the growth rate, the fractions of fuel mass converted to carbon monoxide or carbon dioxide, the heat of combustion, and even the heat release rate not being specified as intended or, in the case of BBRAD scenarios, even as required. For example, the necessary output data was not specified in 83 % of the evaluated input files, making this the most frequent of all errors and one being committed by 80 % of the users. Even worse, for the evaluated files which were supposed to comply with one of the BBRAD scenarios, all of the necessary output quantities were not specified in a single case. The presence of each of the other errors of this kind ranges from 14 % to 25 % and they are committed by between 30 % and 70 % of the users. Considering the fact that these errors arise due to the user's stated intention for the simulation not being reflected by the actual user-generated input data, this is disturbing. The implication of the frequency of these errors on the quality of the obtained results is primarily that these often may not fully reflect the user's stated intention with the simulation. While no further inquiry as to why this is so frequent has been made, two possible explanations seem reasonable. The errors could be either intentional or unintentional, neither of which would be very flattering for the actual users committing them. Intentional errors could for example be explained by users masking deficiencies in the input data, in order to appear compliant with the building code in cases where that is not the case. Unintentional errors could for example be explained by users lacking the sufficient knowledge to properly achieve what they intend to do.

The remaining three errors, out of the nine which are each present in more than 10 % of files, are $D^* / \mathrm{d}x$ being less than 10, mesh groups not being entered from finest to coarsest, and mesh cells not being cubic. All of these are closely related to the mesh, and thereby potentially significant to the turbulence resolution and the flow-field obtained in the simulations. The implication of the frequency of these errors on the quality of the obtained results is that it often might be questionable. However, as discussed in section 4.2.1, other—and perhaps more suitable—ways of asserting a sufficient quality in regard to grid resolution do exist and might explain the frequency of at least the first of these errors.

4.3 Future work

The developed application could be further applied to perform evaluations of large sets of input data. For example, if the results of the case study resemble that of the general population, a problem in regard to the quality of risk assessments based on FDS calculations may exist. It would therefore be of interest to perform a more thorough investigation into the practices of the general population of FDS. It would also be of interest to study the occurrence and frequency of different errors over time, e.g. what effect has the publications of the different recommendations had on the quality of FDS input data?

To increase the usefulness of the developed application, further refining the quantification of the checklist items and possibly expanding it to cover more of the items in the qualitative checklist would be of interest. For example, as implemented, the heat release rate is calculated simply as specified by the user. However, FDS adapts the obstructions and vents to match the grid, which might affect the actual values achieved in a simulation. This, and other similar aspects, could also be implemented.

The results indicate that certain recommendations in the different guidelines often are not followed. It would be of interest to further study the cause of this. Such a study could perhaps be performed by interviewing FDS users and covering aspects such as whether the guidelines are considered too strict, erroneous, or simply not applicable.

Last but not least, since FDS version 5 has been succeeded by version 6, it would be of interest to update the application for this newer version.

4.4 Conclusions

In regard to the more general objectives of the project—i.e. *determining to which extent the quality control process for FDS input data could be systematized and rationalized by developing an automated software application* and *whether the occurrence of user related errors in FDS input data could be reduced by applying the application*—it has been concluded that the developed application indeed is a useful tool in discovering such errors, thereby also providing a means of reducing their occurrence.

More specifically, the answers to research questions one and two—i.e. *which checks should be part of the quality control process for user-generated FDS input data* and *which of the identified checks can be automated and performed using a software application*—have been presented in the form of two checklists in section 3.2; one qualitative intended to be checked manually by the user and one quantitative intended to be checked using the developed application.

As to research question three—i.e. *whether the application of the developed software application could result in a reduced number of errors in FDS input data*—since the application was capable of identifying errors in all evaluated input files, the answer is yes.

No general conclusions in regard to research question four—i.e. *which user related errors are most commonly occurring in FDS input data*—have been drawn, due to the limited scope of the performed case study. However, the actual frequencies with which different errors occurred in the case study have been presented in section 3.4. The most frequently occurring errors are either of the kind where an intended parameter value is not correctly specified or closely related to the mesh. The errors concerned in the first group are the output data, the growth rate, the fractions of fuel mass converted to carbon monoxide or carbon dioxide, the heat of combustion, and even the heat release rate not being specified as intended or, in the case of BBRAD scenarios, even as required. The errors concerned in the second group are $D^* / \mathrm{d}x$ being less than 10, mesh groups not being entered from finest to coarsest, and mesh cells not being cubic. While these results might not be generally representative, they could serve as an indication of certain areas which should be handled with care in producing actual FDS input data.

References

BIV. (2013). *CFD-beräkningar med FDS - BIV:s tillämpningsdokument 2013-2 utgåva 1.* Föreningen för brandteknisk ingenjörsvetenskap (BIV).

Boverket. (2011). *Boverkets allmänna råd om analytisk dimensionering av byggnaders brandskydd, BBRAD, BFS 2011:27.* Author.

Boverket. (2013). *Boverkets allmänna råd om analytisk dimensionering av byggnaders brandskydd, BBRAD, BFS 2011:27 med ändringar t.o.m. BFS 2013:12.* Author.

Boverket. (2014). *Boverkets byggregler, BBR, BFS 2011:6 med ändringar t.o.m BFS 2014:3.* Author.

Briab. (2012). *Vägledning - brandgasfyllnad.* Briab Brand & Riskingenjörerna AB.

Jakobsen, A., Valkvist, M. B. S., Bennetsen, J. C., Carstensen, R. E., Madsen, G. L., Sommerlund-Thorsen, K., … Sørensen, L. S. (2009). *CFD Best Practice.* Best Practice gruppen.

Karlsson, B., & Quintiere, J. G. (2000). *Enclosure fire dynamics.* CRC Press.

Lauridsen, K., Christou, M., Amendola, A., Markert, F., Kozine, I., & Fiori, M. (2001, September). Assessing the uncertainties in the process of risk analysis of chemical establishments: Part II. In E. Zio, M. Demichela, & N. Piccinini (Eds.), *Towards a safer world - proceedings of the ESREL conference* (Vol. 1, p. 599-606). Turin, Italy.

McGrattan, K., McDermott, R., Hostikka, S., & Floyd, J. (2010a). *Fire Dynamics Simulator (Version 5) - Technical Reference Guide - Volume 3: Validation.* National Institute of Standards and Technology, U.S. Department of Commerce.

McGrattan, K., McDermott, R., Hostikka, S., & Floyd, J. (2010b). *Fire Dynamics Simulator (Version 5) - User's Guide.* National Institute of Standards and Technology, U.S. Department of Commerce.

Nystedt, F., & Frantzich, H. (2011). *Kvalitetsmanual för brandtekniska anlyser vid svenska kärntekniska anläggningar* (Tech. Rep.). Brandteknik och riskhantering, Lunds tekniska högskola, Lunds universitet.

Python Software Foundation. (2015a, 04). *General Python FAQ.* Retrieved 2015-04-26, from https://docs.python.org/2/faq/general.html

Python Software Foundation. (2015b, 04). *Python 2.7.10rc0 documentation.* Retrieved 2015-04-16, from https://docs.python.org/2.7/

Appendix A

Quantification of checklist items

This appendix contains a description of how the items of the quantitative checklist have been calculated and checked. Hence, the item numbers refer to the ones presented in table 3.6 in section 3.2. The methods described here have been implemented in the software application used to automate the quality control process. By supplying the application with a valid FDS input file and a valid intent specification and actually running it, a series of functions are run in steps. In general terms these steps can be described as:

1. Structuring raw input data.
2. Discarding any irrelevant data.
3. Converting raw data to the correct data types.
4. Adding default values for required, but not specified, parameters.
5. Evaluating the data and generating a checklist.

An input file containing the lines:

```
&GROUPA A=0.1 B='EXAMPLE'/

&GROUPB C=1/

&GROUPB C=2/
```

would be processed as follows:

1. The text is converted to a Python dictionary containing the group names as keys, whose values are lists of dictionaries, one for each object of the group. In this case:

```
    {“GROUPA”: [{“A”: ‘0.1’, “B”: “‘EXAMPLE’”}], “GROUPB”:
    [{“C”: “1”}, {“C”: “2”}]}
    ```

2. Any parameters which are not used in the control process are discarded. For example, if `″A″` were to be such a parameter, after this step the remaining data would be:

    ```
    {“GROUPA”: [{“B”: “‘EXAMPLE’”}], “GROUPB”: [{“C”: “1”},
    {“C”: “2”}]}
    ```

3. All parameter values are converted to the correct Python data types. For example, numbers stored as strings will be converted to the `Decimal` type—which is used to avoid any precision issues in performing arithmetic operations—and any strings will be stripped of leading and trailing apostrophes. In this case:

    ```
    {“GROUPA”: [{“B”: “EXAMPLE”}], “GROUPB”: [{“C”: Decimal(“1”)},
    {“C”: Decimal(“2”)}]}
    ```

4. Any parameters required to perform the quality control, but which have not been specified by the user, will be added and set to their default values, as specified in the FDS user's guide (McGrattan et al., 2010b). If `D` were to be a required parameter of `GROUPA`, with a default value of `Decimal(“5”)`, the resulting data would be:

    ```
    {“GROUPA”: [{“B”: “EXAMPLE”, “D”: Decimal(“5”)}], “GROUPB”:
    [{“C”: Decimal(“1”)}, {“C”: Decimal(“2”)}]}
    ```

5. When reaching step five, the individual checks described in the subsections below are performed on the processed data. As an example, the actual code used to check item 1 has also been included.

A.1 Item 1

Mesh cells are cubic.

For each `MESH` group, this item is checked by:

1. Reading the `IJK` and `XB` parameters.
2. Calculating the x-, y-, and z-lengths.
3. Checking if $x = y$ and $x = z$.

The group `&MESH ... IJK=10,10,5 XB=0,10,20,30,40,50/` will be checked as follows:

1. $IJK = 10, 10, 5$
   $XB = 0, 10, 20, 30, 40, 50$
2. $x = (XB_1 - XB_0)/IJK_0 = (10 - 0)/10 = 1$
   $y = (XB_3 - XB_2)/IJK_1 = (30 - 20)/10 = 1$
   $z = (XB_5 - XB_4)/IJK_2 = (50 - 40)/5 = 2$
3. Since $1 \neq 2 \Rightarrow x \neq z$, the check will fail.

The necessary data to perform the check is all `MESH` groups and their `IJK` and `XB` parameters.

The actual code used to perform the check is presented below. First, the function `additional_mesh_data()` will be run to, among others, determine whether the flag `cubic_cells` should be set to `True` or `False`, i.e. whether the cells are cubic or not. When the actual check later is performed the `check()` method of an object of class `CheckCubic` is run. This method sets the `valid` property of the class to `False` if the cells are not cubic or to `True` if they are.

```
def additional_mesh_data(data):
    mesh_num = 0
    if hasattr(data.get('MESH'), '__iter__'):
        data['mesh'] = {}
        for mesh in data['MESH']:
            mesh_num += 1
            valid_numbers = True
            cubic_cells = False
            jk = mesh['IJK'][1:]
            for i in jk:
                if len(set(factor(i)) - set((1, 2, 3, 5))) != 0:
                    valid_numbers = False

            x_length = (mesh['XB'][1]-mesh['XB'][0])/mesh['IJK'][0]
            y_length = (mesh['XB'][3]-mesh['XB'][2])/mesh['IJK'][1]
            z_length = (mesh['XB'][5]-mesh['XB'][4])/mesh['IJK'][2]

            # Check if cells are cubic.
            if x_length == y_length and x_length == z_length:
                cubic_cells = True

            mesh['cubic_cells'] = cubic_cells
            mesh['mesh_num'] = mesh_num
            mesh['valid_numbers'] = valid_numbers
```

```
            mesh['xyz_lengths'] = (x_length, y_length, z_length)
            data['mesh'][mesh_num] = mesh
    return data

class CheckCubic(Check):
    instructions = u"Mesh cells are cubic."
    error_message_template = Template(u"Mesh cells are not cubic in mesh "\
            u"$meshes.")

    def check(self, data, intent):
        """Check if any mesh contains non-cubic cells.

        Store such mesh nubers in 'self.non_cubic_meshes' and set the error
        message to name each invalid mesh.
        """
        non_cubic_meshes = []
        if hasattr(data.get('MESH'), '__iter__'):
            for mesh in data['MESH']:
                if mesh.get('cubic_cells') == False:
                    non_cubic_meshes.append(mesh.get('mesh_num'))
        if non_cubic_meshes != []:
            self.valid = False
            self.error_message_parameters =\
                    {'meshes': self.and_join(non_cubic_meshes)}
        else:
            self.valid = True
```

A.2 Item 2

The item states that:

> *The dimensions in the y- and z-directions can be written on the form* $2^l 3^m 5^n$.

For each `MESH` group the item, is checked by:

1. Reading two last values of the `IJK` parameter.
2. Performing an integer factorization for each of the values and checking that no factors other than 2, 3, and 5 are present.

The group `&MESH ... IJK=10,10,22/` will be checked as follows:

1. $IJK_2 = 10$
   $IJK_3 = 22$
2. While $IJK_2 = 2 \times 5$ is a valid factorization, $IJK_3 = 2 \times 11$ is invalid and the check will fail.

The necessary data is all `MESH` groups and their `IJK` parameters.

A.3 Item 3

The item states that:

> *Mesh groups are entered from finest to coarsest.*

This recommendation is given in the FDS user's guide (McGrattan et al., 2010b). However, no instructions on how to determine the actual "fineness" of a `MESH` group is given. It has therefore been assumed that one `MESH` group is finer than another if its cell volume—i.e. the respective volume of each grid cell, not the total volume of all cells—is smaller than that of the other. In order for FDS to run when applying the `MPI_PROCESS` parameter, the meshes have to be entered by increasing process number—i.e. all `MESH` groups belonging to process 0 must be entered before those belonging to process 1, and so on. In these cases, it has been assumed that the `MESH` groups should be entered from finest to coarsest in each process group, and that the groups should be ordered from finest to coarsest based on the smallest cell volume in each group.

The item is checked by:

1. Calculating the x-, y-, and z-lengths for each `MESH` group by reading their `XB` and `IJK` parameters (in the same way as has been described in appendix A.1) and calculating the cell volume, $V_{cell} = xyz$.
2. Reading the `MPI_PROCESS` parameter of each `MESH` group.
3. Checking that no mesh group precedes another mesh group with a smaller cell volume for each `MPI_PROCESS` and that the processes are ordered so that no process group precedes another process group containing a `MESH` group with a smaller cell volume than the smallest volume of any `MESH` group in the process group.

The necessary input data is the relative order of all `MESH` groups in the input file and their `IJK`, `MPI_PROCESS`, and `XB` parameters.

A.4 Item 4

The item states that:

> *Mesh boundaries do not cross a fire.*

The item requires determining which objects in the model are fires, which was achieved by checking which `OBST` and `VENT` groups reference a `SURF` group with either the `HRRPUA` or `MLRPUA` parameter set.

The item is checked by:

1. Determining which `SURF` groups constitute fire surfaces by checking if they have either the `HRRPUA` or `MLRPUA` parameter set and reading their `ID` parameter.
2. Identifying all fire objects by checking which `OBST` and `VENT` groups reference a the `ID` of a fire surface, by reading any `SURF_ID`, `SURF_IDS`, or `SURF_ID6` parameters.
3. Reading the `XB` parameter of each identified fire object.
4. Reading the `XB` parameter of each `MESH` group.
5. Checking that each mesh does not partially contain a fire object by checking that it does not cross a fire object.

The necessary input data is:

- All `MESH` groups and the values of their respective `XB` parameter.
- All `OBST` groups and the values of their respective `SURF_ID`, `SURF_IDS`, `SURF_ID6`, and`XB` parameters.
- All `SURF` groups and their respective `HRRPUA`, `ID`, and `MLRPUA` parameters.
- All `VENT` groups and the values of their respective `SURF_ID` and `XB` parameters.

A.5 Item 5

The item states that:

> *Meshes do not overlap.*

The item is checked by:

1. Reading the `XB` parameter of each `MESH` group.
2. For each combination (A, B) of two `MESH` groups, checking that their `XB` parameters do not overlap. An overlap would mean that both $XB_{i,A} \leq$

$XB_{i,B}$ and $XB_{i+1,A} > XB_{i,B}$ are true, or that both $XB_{i,B} \leq XB_{i,A}$ and $XB_{i+1,B} > XB_{i,A}$ are true, for $i = 1, 3, 5$.

The necessary data is all `MESH` groups and their `XB` parameters.

A.6 Item 6

The item states that:

> $D^*/\,\mathrm{d}x$ *is greater than or equal to 10.*

In accordance with McGrattan et al. (2010b), D^* is calculated by:

$$D^* = \left(\frac{\dot{Q}}{\rho_\infty c_p T_\infty \sqrt{g}}\right)^{2/5} \tag{A.1}$$

The dependence on ρ_∞, c_p, and T_∞ means that the specified ambient temperature needs to be read from the input data and suitable values for the temperature dependent parameters need to be determined. Before the final calculation of the ratio can be performed the value of $\mathrm{d}x$ also needs to be calculated based on the coordinates specified in the input data.

The steps for calculating D^* are:

- $\dot{Q}$ will be the peak heat release rate, $\dot{Q} = \dot{Q}_{max}$, and calculated by:
    - Fire objects will be identified as has been described in appendix A.4.
    - The sides on which any fire surfaces are applied will be identified by reading any `SURF_ID`, `SURF_IDS`, or `SURF_ID6` parameters and their areas calculated by reading the `XB` parameter.
    - If the `MLRPUA` parameter has been specified for the fire surface and the `HEAT_OF_COMBUSTION` parameter has been specified for the `REAC` group, $\dot{Q}_{max}$ is calculated by multiplying these parameter values and the fire area, i.e. `MLRPUA` × `HEAT_OF_COMBUSTION` × *area*. If no `HEAT_OF_COMBUSTION` parameter has been specified for the `REAC` group, this parameter will be calculated by multiplying the `EPUMO2` parameter of the `REAC` group with the oxygen yield. If the `HRRPUA` parameter instead has been specified for the fire surface, $\dot{Q}_{max}$ is calculated by multiplying this parameter value with the area, i.e. `HRRPUA` × *area*. If any `RAMP` group is referenced by the fire surface, the calculated peak heat release rate will be reduced by the peak value of the ramp.

- ρ_∞, calculated for the value of the `TAMB` parameter of the `MISC` group by applying the ideal gas law $\rho_\infty = \frac{p_\infty M_\infty}{RT_\infty}$.
  - p_∞, the pressure in Pa, read from the `P_INF` parameter of the `MISC` group.
  - M_∞, $28.97 \times 10^{-3}\,\mathrm{kg\,mol^{-1}}$.
  - R, $8.314\,\mathrm{m^3\,Pa\,K^{-1}\,mol^{-1}}$.
  - T_∞, K. Read from the `TMPA` parameter of the `MISC` group in Celsius and converted to kelvin by $T_K = 273.15 + T_C$.
- c_p, $1000\,\mathrm{kJ\,kg^{-1}\,K^{-1}}$.
- T_{inf}, see above.
- g, the gravitational constant, calculated by adding the values specified in the `GVEC` parameter of the `MISC` group.

The steps for calculating dx are:

- Depending on the orientation of the fire object, determining its x- and y-lengths, x- and z-lengths, or y- and z-lengths. The greatest of the two lengths will constitute the value of dx.

The necessary input data is:

- All `MESH` groups and the values of their respective `XB` parameter.
- The `MISC` group and the values of its `GVEC`, `P_INF`, and `TMPA` parameters.
- All `OBST` groups and the values of their respective `SURF_ID`, `SURF_IDS`, `SURF_ID6`, and`XB` parameters.
- All `SURF` groups and their respective `HRRPUA`, `ID`, `MLRPUA`, and `RAMP_Q` parameters.
- All `VENT` groups and the values of their respective `SURF_ID` and `XB` parameters.

A.7 Item 7

The item states that:

> *The MASS_EXTINCTION_COEFFICIENT parameter of the REAC group should be set between* $7600\,\mathrm{m^2\,kg^{-1}}$ *to* $9800\,\mathrm{m^2\,kg^{-1}}$ *for most flaming fuels.*

The item is checked by:

1. Reading the `MASS_EXTINCTION_COEFFICIENT` parameter of the `REAC` group.

2. Checking that the specified value is within the specified interval.

A.8 Item 8

The item states that:

> *The `VISIBILITY_FACTOR` parameter of the `REAC` group should be set to 3 when modeling light-reflecting signs and to 8 when modeling light-emitting signs.*

The item is checked by:

1. Reading the `VISIBILITY_FACTOR` parameter of the `REAC` group.
2. Checking that the specified value is equal to either 3 or 8.

A.9 Item 9

The item states that:

> $\dot{Q}^*$ *is between 0.3 and 2.5.*

In accordance with Nystedt and Frantzich (2011), $\dot{Q}^*$ is calculated by:

$$\dot{Q}^* = (\frac{\dot{Q}}{\rho_\infty c_p T_\infty \sqrt{gD} D^2})^{2/5} \tag{A.2}$$

The item is checked by:

1. Calculating $\dot{Q}$, ρ_∞, c_p, T_∞, and g for each fire object as described in appendix A.6.
2. Calculating D by assuming the fire area represents a circle, i.e. $D = 2\sqrt{\frac{A}{\pi}}$.
3. Checking that the value of $\dot{Q}^*$ for each fire object is within the specified bounds.

The necessary input data is:

- All `MESH` groups and the values of their respective `XB` parameter.
- The `MISC` group and the values of its `GVEC`, `P_INF`, and `TMPA` parameters.
- All `OBST` groups and the values of their respective `SURF_ID`, `SURF_IDS`, `SURF_ID6`, and`XB` parameters.

- All `SURF` groups and their respective `HRRPUA`, `ID`, `MLRPUA`, and `RAMP_Q` parameters.
- All `VENT` groups and the values of their respective `SURF_ID` and `XB` parameters.

A.10 Item 10

The item states that:

> *The heat release rate has been specified as intended.*

The item is checked by:

1. Calculating the peak heat release rate as described in appendix A.9.
2. Checking if it is equal to a supplied intended value.

The necessary input data is:

- All `OBST` groups and the values of their respective `SURF_ID`, `SURF_IDS`, `SURF_ID6`, and`XB` parameters.
- All `SURF` groups and their respective `HRRPUA`, `ID`, `MLRPUA`, and `RAMP_Q` parameters.
- All `VENT` groups and the values of their respective `SURF_ID` and `XB` parameters.

A.11 Item 11

The item states that:

> *The growth rate has been specified as intended.*

There are multiple ways of specifying a fire and how it is supposed to grow in FDS. However, in none of these the growth rate is specified explicitly. The closest way is to set `TAU_Q` on the `SURF` group of the specified fire surface, which specifies the time at which the maximum `HRRPUA` will be reached. By setting positive values the `HRRPUA` will ramp up like $tanh(t/\tau)$—where t is time and τ is the specified `TAU_Q` value—and by setting negative values the `HRRPUA` will ramp up like $(t/\tau)^2$ (McGrattan et al., 2010b).

The growth rate refers to the α parameter of the t-squared fire—i.e. the fire described on the form $\dot{Q} = \alpha t^2$. Using the terminology of the FDS user's guide (McGrattan et al., 2010b) the expression can be written as $\dot{Q} = \dot{Q}_{max}(t/\tau)^2$.

Combining these two expression gives the following expression for calculating the growth rate for the t-squared fire:

$$\alpha = \frac{\dot{Q}_{max}}{\tau^2}$$

The item is checked by:

1. Calculating $\dot{Q}_{max}$ as described in appendix A.6.
2. Reading the value of `TAU_Q` on any `SURF` groups where this parameter is set.
3. If the value is negative—i.e. if a t-squared fire is specified—calculating α as specified above.
4. Checking that the calculated value of α equals the intended value.

The necessary input data is:

- All `OBST` groups and the values of their respective `SURF_ID`, `SURF_IDS`, `SURF_ID6`, and`XB` parameters.
- All `SURF` groups and their respective `HRRPUA`, `ID`, `MLRPUA`, `RAMP_Q`, and `TAU_Q` parameters.
- All `VENT` groups and the values of their respective `SURF_ID` and `XB` parameters.

A.12 Item 12

The item states that:

> *The heat of combustion has been specified as intended.*

The item is checked by:

1. Reading the `HEAT_OF_COMBUSTION` parameter of the `REAC` group, if present.
2. Checking that the actual value equals the intended value.

A.13 Item 13

The item states that:

> *The fraction of fuel mass converted to soot has been specified as intended.*

The item is checked by:

1. Reading the `SOOT_YIELD` parameter of the `REAC` group.
2. Checking that the actual value equals the intended value.

A.14 Item 14

The item states that:

> *The fraction of fuel mass converted to carbon monoxide has been specified as intended.*

The item is checked by:

1. Reading the `CO_YIELD` parameter of the `REAC` group.
2. Checking that the actual value equals the intended value.

A.15 Item 15

The item states that:

> *The fraction of fuel mass converted to carbon dioxide has been specified as intended.*

The fraction of fuel mass converted to carbon dioxide cannot be set explicitly in the input file. Instead, it depends on a number of parameters and has to be calculated manually. There are two ways of calculating the parameter, depending on whether the one-step or the two-step reaction model is used. To determine the reaction model the `CO_PRODUCTION` parameter of the `MISC` group must be read. If the parameter is set to `.FALSE.` the one-step model is used and if it is set to `.TRUE.` the two-step model is used. Only the one-step model has been implemented.

For the one-step model the sought fraction (y_{CO_2}) can be calculated by:

$$y_{CO_2} = \frac{\nu_{CO2} W_{CO2}}{W_F} \tag{A.3}$$

where:

$$
\begin{aligned}
\nu_{CO_2} &= \text{stoichiometric coefficient for the produced carbon dioxide} \\
W_{CO_2} &= \text{the molecular weight of carbon monoxide} \\
W_F &= \text{the molecular weight of the fuel}
\end{aligned}
$$

The stoichiometric coefficient for the produced carbon dioxide is calculated by:

$$\nu_{CO_2} = x - \nu_{CO} - (1 - X_H)\nu_S \tag{A.4}$$

where:

$$
\begin{aligned}
x &= \text{number of carbon atoms in the fuel} \\
\nu_{CO} &= \text{stoichiometric coefficient for the produced CO} \\
X_H &= \text{atomic fraction of hydrogen in soot} \\
\nu_S &= \text{stoichiometric coefficient for the produced soot}
\end{aligned}
$$

The values of x and X_H can be read from the `C` and `SOOT_H_FRACTION` parameters of the `REAC` group. The values of ν_{CO} and ν_S have to be calculated manually by:

$$\nu_i = \frac{W_F}{W_i} y_i \tag{A.5}$$

where:

$$
\begin{aligned}
W_F &= \text{molecular weight of the fuel} \\
W_i &= \text{molecular weight of the substance} \\
y_i &= \text{yield of the substance}
\end{aligned}
$$

The yields y_{CO} and y_S are read from the `CO_YIELD` and `SOOT_YIELD` parameters of the `REAC` group. The molecular weight of carbon monoxide (W_{CO}) is known but the molecular weight of the fuel (W_F) and the soot (W_S) have to be calculated manually. The molecular weight of the fuel is calculated by:

$$W_F = W_C x + W_H y + W_O z + W_N a + W_{other} b \tag{A.6}$$

where:

$$\begin{aligned} W_i &= \text{molecular weight of substance i} \\ x &= \text{number of carbon atoms in fuel} \\ y &= \text{number of hydrogen atoms in fuel} \\ z &= \text{number of oxygen atoms in fuel} \\ a &= \text{number of nitrogen atoms in fuel} \\ b &= \text{number of other atoms in fuel} \end{aligned}$$

The values of x, y, z, a, and b are read from the `C`, `H`, `O`, `N`, and `OTHER` parameters of the `REAC` group. The molecular weights of carbon, hydrogen, oxygen, and nitrogen are known, but that of the "other" atoms in the fuel—i.e. the one weight specified for any atoms other than the ones previously mentioned—must be read from the `MW_OTHER` parameter of the `REAC` group.

The molecular weight of the soot is calculated by:

$$W_S = X_H W_H + (1 - X_H) W_C \tag{A.7}$$

where:

$$\begin{aligned} X_H &= \text{atomic fraction of hydrogen in the soot} \\ W_i &= \text{molecular weight of substance i} \end{aligned}$$

The atomic fraction of hydrogen in soot can be read from the `SOOT_H_FRACTION` parameter of the `REAC` group and the molecular weights of hydrogen and carbon are known.

The item is checked by:

1. Calculating eqs. (A.3) to (A.7).
2. Checking that the actual value of y_{CO_2} equals the intended value.

A.16 Item 16

The item states that:

> *`SURF` groups with the `BURN_AWAY` parameter set to `.TRUE.` cover all surfaces of any `OBST` group on which they are applied.*

Surfaces are applied to `OBST` groups using the `SURF_ID`, `SURF_IDS`, or `SURF_ID6` parameters. For all `OBST` groups referencing a `SURF` group with the `BURN_AWAY` parameter set to `.TRUE.`, check that the same `SURF` group is applied to all surfaces of the `OBST` group.

The item is checked by:

1. Checking which, if any, `SURF` groups have the `BURN_AWAY` parameter set to `.TRUE.`.
2. For each `OBST` group:
    1. Checking if its `SURF_ID`/`SURF_IDS`/`SURF_ID6` parameter references any identified `SURF` groups.
    2. If that is the case, checking that the `SURF` group is applied to all surfaces of the `OBST` group.

Valid examples:

```
&SURF ..., ID='burn' BURN_AWAY=.TRUE./
&OBST ..., SURF_ID='burn'/
&OBST ..., SURF_IDS='burn','burn','burn'/
&OBST ..., SURF_ID6='burn','burn','burn','burn','burn','burn'/
```

Invalid examples:

```
&SURF ..., ID='burn' BURN_AWAY=.TRUE./
&OBST ..., SURF_IDS='burn','burn','INERT'/
&OBST ..., SURF_ID6='burn','burn','burn','burn','burn','INERT'/
```

A.17 Item 17

The item states that:

> *No velocity is specified for `SURF` groups where the `HRRPUA` or `MLRPUA` parameters are specified or where solid phase reaction parameters are specified.*

The item is checked by:

1. For each `SURF` group where the `VEL` parameter value does not equal 0, reading the `HRRPUA`, `MLRPUA`, and `MATL_ID` parameters.
2. If either `HRRPUA` or `MLRPUA` does not equal 0 the check is invalid.
3. Reading the `N_REACTIONS` parameter of the referenced `MATL` group.
4. If `N_REACTIONS` does not equal 0, the check is invalid.

A.18 Item 18

The item states that:

> *Finite-rate reactions have not been specified, unless DNS mode is used.*

The item is checked by:

1. Reading the `E` parameter of the `REAC` groupand the `DNS` parameter of the `MISC` group.
2. If the `E` parameter has been set but the `DNS` parameter is not set to `.TRUE.`, the check is invalid.

A.19 Item 19

The item states that:

> *If `SPREAD_RATE` is specified in any `SURF` or `VENT` groups, the `XYZ` parameter is centered on the fire surface.*

The item is checked by:

1. Reading the `SPREAD_RATE` parameter of each `SURF` and `VENT` group where it has been specified.
2. If not equal to 0, check that the corresponding `XYZ` parameter value is in the same plane as and in the middle of the fire surface by reading its `XB` parameter.

A.20 Item 20

The item states that:

> *Both `HRRPUA` and `MLRPUA` are not specified in the same `SURF` group.*

The item is checked by:

1. Reading the `HRRPUA` and `MLRPUA` parameters of each `SURF` group.
2. If more than one of the parameters do not equal 0, the check is invalid.

A.21 Item 21

The item states that:

> *Output data has been specified.*

The item is checked by:

1. Checking that at least one `BNDF`, `ISOF`, `DEVC`, `PROF`, or `SLCF` group with the `QUANTITY` parameter set is specified.

A.22 Item 22

The item states that:

> *The necessary output data has been specified.*

Necessary output data can be provided as an intention, or as a requirement when applying the BBRAD checks.

The item is checked by:

1. For each quantity specified as necessary output data:
    1. Reading the `QUANTITY` parameter of each `BNDF`, `ISOF`, `DEVC`, `PROF`, or `SLCF` group.
    2. Checking if the necessary output quantity matches any of the `QUANTITY` parameters. If not, the check is invalid.

A.23 Item 23

The item states that:

> `VENT` *groups do not overlap.*

The item is checked by:

1. Reading the `XB` parameter of each `VENT` group.

2. For each `XB` parameter, checking that no other `XB` parameter overlaps in more than two dimensions.

A.24 Item 24

The item states that:

> *The `MIRROR` boundary condition is not used along the centerline of a fire obstruction or fire vent.*

The item is checked by:

1. Reading the `XB` parameter of each fire object.
2. Reading the `XB` parameter of each `VENT` group with the `SURF_ID` parameter set to `MIRROR`.
3. For each `VENT` group, checking that it does not intersect any fire object by:
    1. Checking in which direction the `VENT` is planar—i.e. which of $XB_1 = XB_2$, $XB_3 = XB_4$, and $XB_5 = XB_6$ is true.
    2. If $XB_{FIRE,a} \leq XB_{VENT,ab} \leq XB_{FIRE,b}$ is true, the check is invalid.

A.25 Item 25

The item states that:

> *`VENT` groups with the parameter `SURF_ID` set to `'MIRROR'` or `'OPEN'` are not activated or deactivated during the simulation.*

The item is checked by:

1. Reading the `SURF_ID` parameter of each `VENT` group.
2. If the parameter is set to `'MIRROR'` or `'OPEN'`, checking that the `DEVC_ID` parameter has not been set.

A.26 Item 26

The item states that:

> *Non-pointwise devices do not cross mesh boundaries.*

The item is checked by:

1. Reading the `XB` parameter of each `DEVC` group.
2. Reading the `XB` parameter of each `MESH` group.
3. Checking that the `XB` parameter of each `DEVC` is either completely contained in or outside of each `MESH`. If not, the check is invalid.

A.27 Item 27

The item states that:

> *At least one `VENT` group with the `SURF_ID` parameter set to `'OPEN'` has been placed on the mesh boundary.*

The item is checked by:

1. For each `VENT` group, reading its planar dimension ($XB_{VENT,ab}$).
2. For each `MESH` group, cehcking if $XB_{MESH,a} = XB_{VENT,ab}$ or $XB_{MESH,b} = XB_{VENT,ab}$.
3. If none of the criteria is valid for any of the meshes, the check is invalid.

A.28 Item 28

The item is checked by:

> *The radiation model is used.*

The item is checked by:

1. Reading the `RADIATION` parameter of the `MISC` group.
2. If the parameter is set to `.FALSE.`, the check is invalid.

A.29 Item 29

The item is checked by:

> *The Smagorinsky constant equals 0.20.*

The item is checked by:

1. Reading the `CSMAG` parameter of the `MISC` group.
2. If the parameter is not set to `0.20`, the check is invalid.

Appendix B

Identified errors in input data

A complete summary of all evaluated input files and the detected errors has been presented in table B.1.

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file.*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
1	A	7	Yes		x				x				x	x			x	x							x							
2	A	7	Yes		x				x				x	x			x	x							x							
3	A	5	Yes												x		x	x							x	x						
4	A	5	Yes												x		x	x							x	x						
5	A	6	Yes						x						x		x	x							x	x						
6	B	5	Yes						x						x		x	x							x							
7	B	5	Yes						x						x		x	x							x							
8	B	7	Yes						x				x	x	x		x	x							x							
9	B	6	Yes										x	x	x		x	x							x							
10	B	7	Yes						x				x	x	x		x	x							x							
11	B	2	No						x																x							
12	B	1	No																						x							
13	B	2	No						x																x							

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
14	B	2	No						x																x							
15	B	2	No						x																x							
16	B	2	No						x																x							
17	B	5	Yes						x						x		x	x							x							
18	B	2	No						x																x							
19	B	4	Yes												x		x	x							x							
20	B	4	Yes												x		x	x							x							
21	B	7	Yes			x			x					x	x		x	x							x							
22	B	8	Yes			x			x				x	x	x		x	x							x							
23	C	2	Yes			x																			x							
24	C	2	Yes												x										x							
25	D	1	No						x																							
26	E	2	Yes						x																x							

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
27	E	1	Yes																						x							
28	E	1	Yes																						x							
29	E	1	Yes																						x							
30	E	3	Yes						x					x											x							
31	E	3	Yes						x					x											x							
32	E	3	Yes						x					x											x							
33	E	3	Yes						x					x											x							
34	E	1	Yes																						x							
35	E	2	Yes						x																x							
36	E	2	Yes						x																x							
37	E	1	Yes																						x							
38	E	2	Yes						x																x							
39	E	2	Yes						x																x							

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
40	E	2	Yes						x																x							
41	E	2	Yes			x																			x							
42	F	4	No	x					x				x	x																		
43	F	4	No	x					x				x	x																		
44	F	4	No	x					x				x	x																		
45	F	3	No	x					x				x																			
46	F	3	Yes	x													x								x							
47	F	2	Yes	x																					x							
48	F	3	Yes	x													x								x							
49	G	3	No						x			x													x							
50	G	4	No						x				x	x											x							
51	G	2	No						x																x							
52	G	3	No						x			x													x							

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
53	G	4	No						x			x	x												x							
54	H	8	Yes	x		x			x						x		x	x							x					x		
55	H	7	Yes	x		x									x		x	x							x					x		
56	H	8	Yes	x		x			x						x		x	x							x					x		
57	H	8	Yes	x		x			x						x		x	x							x					x		
58	I	3	No	x		x			x																							
59	I	3	No	x		x			x																							
60	I	4	No	x		x			x					x																		
61	I	4	No	x		x			x					x																		
62	I	4	No	x		x			x					x																		
63	I	4	No	x		x			x					x																		
64	I	4	No	x		x			x					x																		
65	I	2	No						x			x																				

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
66	I	1	No									x																				
67	J	2	Yes																			x			x							
68	J	3	Yes			x																x			x							
69	J	1	Yes																						x							
70	J	1	Yes																						x							
71	J	2	Yes			x																			x							
72	J	2	Yes			x																			x							
73	J	2	Yes			x																			x							
74	J	3	Yes			x																x			x							
75	J	4	Yes		x	x																x			x							
76	J	4	Yes			x								x								x			x							
77	J	2	Yes			x																			x							
78	J	2	Yes			x																			x							

Table B.1: *File number, user id, total number of detected errors, whether the file was supposed to comply with a BBRAD scenario, and failed checklist items—marked with an x—for each evaluated input file (cont.).*

No.	**User**	**Errors**	**BBRAD**	**Failed checklist items**																												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
79	J	2	Yes			x																			x							
80	J	2	Yes			x																			x							
81	J	2	Yes			x																			x							
82	J	2	Yes			x																			x							
83	J	3	Yes			x																			x					x		
84	J	2	Yes			x																			x							