afa FÖRSÄKRING

SLUTRAPPORT

Projekttitel	Dnr
Utveckling av ett numeriskt beräkningsverktyg med en öppen källkod för att skapa en säkrare arbetsmiljö inom svensk industri avseende dammexplosioner	180028

Projektledare
Chen Huang at RISE Research Institutes of Sweden AB

Innehåll:

Development of a numerical tool using an open-source code for creating a safer working environment for the Swedish industries regarding dust explosions

Final report (from 2019-02-01 until 2022-01-31)

Project members

Chen Huang	RISE
Andrei Lipatnikov	Chalmers
Anders Lönnermark	RISE

Reference Group Members (list sorted by last names)

Thomas Andersson	Lantmännen
Nijaz Basic	Fagerberg
Ricardo Capitao	Fagerberg
Patrik Carlryd	AVS (retired)
Chris Cloney	Dust-Ex Research (Canada)
Ingemar Klaesson	Scandbio
Henrik Larsson	Göteborg Energi
Mats Lindgren	Intressentföreningen för Processäkerhet
Vladimir Molkov	Ulster University (UK)
Ken Nessvi	Brandskyddslaget (former PS Group)
Håkan Nilson	Chalmers
Mats Perslid	IEP Technologies
Marcus Runefors	Lund University
Kristoffer Tannerfeldt	Firefly

Sammanfattning

Dammexplosioner är ett konstant hot mot den fysiska arbetsmiljön inom de svenska processindustrier som hanterar material eller utför processer som skapar brännbart damm. Exempel på sådana industrier är pellets-, pappers-, livsmedels-, metall-, läkemedels- och tillsatsindustri. Det aktuella projektet syftar till att (i) utveckla fysikbaserade och välvaliderade numeriska modeller som kan ta hänsyn till de viktigaste förbränningsfenomenen i dammeplosioner, (ii) utveckla ett välverifierat och effektivt numeriskt verktyg baserat på en öppen källkod OpenFOAM för att uppskatta konsekvenser av dammexplosioner, och (iii) beräkna storskaliga dammexplosioner i processindustrier. Projektresultatet förbättrar förståelsen för dammexplosioner, och tillför processindustrin ett beräkningsverktyg för att skapa säkrare processanläggningar avseende dammexplosioner.

Utvecklingen av modellen och koden genomfördes stegvis. I ett första steg, implanterades den så kallade Flame Speed Closure (FSC)-modellen för förblandade turbulenta flammor i OpenFOAM. Implementeringen verifierades mot analytiska lösningar för 1-dimensionella plana och 3-dimensionella sfäriska förblandade turbulenta flammor. I ett andra steg, validerades den utvecklade koden inklusive modellen, dvs. *FSCDustFoam*, mot småskaliga experimentella sfäriska flammor vid majsstärkelsedammexplosioner i ett fläktomrört förbränningskärl under välkontrollerade laboratorieförhållandena. I ett tredje steg, adapterades FSC-modellen genom att använda de välkände experimentella upptäckterna om självlikheten i flamacceleration för att kunna beräkna storskaliga dammexplosioner i industrier. En överlägsen överensstämmelse mellan experimentella data från tryckavlastade majsstärkelsedammexplosioner i ett 11.5 m^3 tryckkärl och beräkningar med den adapterade FSC-modellen har erhållits.

Trots den framgångsrika utvecklingen av *FSCDustFoam*, finns det fortsatta utmaningar. Konkret kan koden *FSCDustFoam* i dagsläget inte ta hänsyn till effekten av dammexplosioner med tryckavlastningsluckor i olika former. *FSCDustFoam* är likväl ett lovande beräkningsverktyg som kan appliceras och vidareutvecklas för att lösa utmaningar avseende dammexplosioner i den komplicerade verkligheten hos svenska processindustrier.

Abstract

Dust explosion has been a constant threat to the physical working environment of the Swedish process industries which deal with combustible powders. Examples of such industries are pellets, paper, metal processing, food and feed, pharmaceuticals, and additive industries. This project aims at (i) development of physics-based and well-validated models which address the important combustion phenomena in dust explosions, (ii) development of a well-verified and an efficient numerical tool based on an open-source toolbox OpenFOAM for predicting consequences of dust explosions and (iii) simulation of large-scale dust explosions in the process industries. The project result improves the understanding of dust explosions, and it provides the process industries with a numerical tool for designing safer process plant regarding dust explosions.

The model and code development were carried out in a step-by-step fashion. First, the so-called Flame Speed Closure (FSC) model for premixed turbulent combustion, was implemented into OpenFOAM. The implementation was verified against analytical solutions for 1-dimensional planar and 3-dimensional spherical turbulent flames. Second, the developed code including the model, i.e., *FSCDustFoam*, was validated against experimental data on corn starch dust explosion in a fan-stirred explosion vessel under well-controlled laboratory conditions. Third, the FSC model was extended by adapting the well-known experimental observations of the self-similarity of the flame acceleration to address large-scale industrial dust explosions. An excellent agreement between measurements of vented corn starch dust explosions in an 11.5 m^3 vessel and the simulations using the extended the FSC model was obtained.

In spite of the successful development of *FSCDustFoam*, challenges remain. Specifically, the current version of *FSCDustFoam* cannot address the effect of different shapes of vent openings on dust explosions. Nevertheless, *FSCDustFoam* is a promising tool to be applied and further developed to resolve the challenging reality regarding dust explosions in the Swedish process industries.

Contents

1. Project aim and background

1.1 Dust explosion incidents

A smouldering fire was detected when emptying a silo with approximately 200-ton sunflower seeds at a food factory in Lidköping on October 21st, 2021. The smouldering fire was then developed into a big fire and a suspected dust explosion. Five fire-fighting forces were present to put out the fire [1]. Similar fire and explosion incidents occurred in the same factory in 2020 [2]. Dust explosion is a constant threat to the process industries worldwide especially among countries and regions with high industrial output. Fire and explosion incidents have continuously been reported globally, e.g., see the latest five-year incidents statistics in the US in Figure 1. Also, in Sweden dust explosion represents a continuous threat to the working environment of industries which deal with combustible powders e.g., woodworking, metal processing, food and feed, pharmaceuticals and additive industries. There is at least one dust explosion accident reported to Arbetsmiljöverket per month [3], and it is highly possible that there are many more unreported accidents. Some examples during the recent years are:

- several dust explosions occurred in a food factory in Lidköping on 21st October, 2021 [1],
- a metal dust explosion occurred in the steel industry in Falköping on 3rd November, 2020 [4],
- a dust explosion occurred during the handling of metal dust at SSAB in 2019 [5],
- an explosion of a truck loaded a 20-meter diameter silo with pellets in Gothenburg on 7th March 2017 [6],
- a metal dust explosion occurred in an aluminium pigment company in Huskvarna on 28th March 2017 [7],
- seven people were injured due to a severe dust explosion during cleaning of a dust extractor in Trångsund on 15th August 2017 [8],
- a dust explosion in a 40-meter-high grain silo in Tråvad on 29th August 2017 [9].

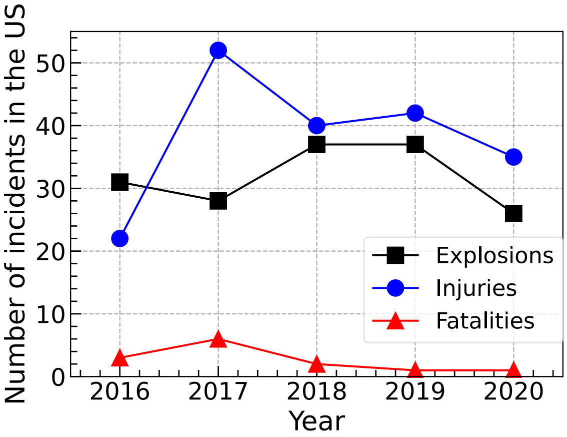


Figure 1. Explosions, injuries and fatalities incidents data related to combustible dust during 2016 and 2020 in the US [10].

According to the Afa Försäkring's statistical analysis, the number of workers who received severe work-related injuries due to fire, explosion, welding, electricity or cold has decreased during 2015 and 2019 (see Figure 2) [11-14]. Unfortunately, the percentage of medical invalidity has increased among those workers during 2015 and 2019 (see Figure 3). Moreover, there is one to two fatalities each year among those workers during this time period [11-14]. A severe work-related injury is defined as an incident which leads to sick leave for more than 30 days or a medical invalidity [11-14]. A medical invalidity is a measure of a permanent physical disability as a consequence of an injury [11-14]. It is worth noting that dust explosion incidents are only a part of the statistics in Figure 2 and Figure 3. More precise data on injuries related to dust explosions at workplaces can be obtained by analysing the Afa Försäkring's statistics in detail. Nevertheless, male workers in metal working and other industries are more common in the dust explosion incidents.

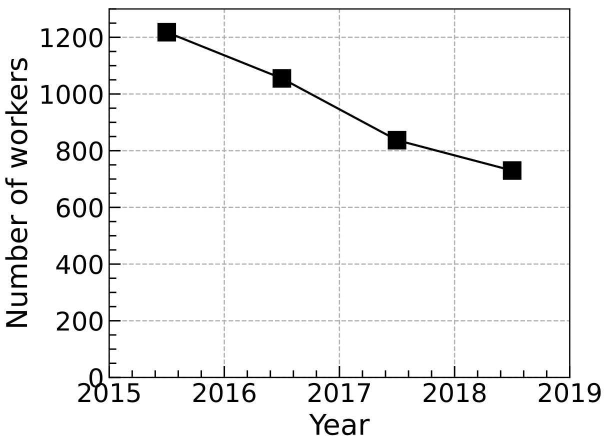


Figure 2. Number of workers who received severe work-related injuries due to fire, explosion, welding, electricity or cold, during 2015 and 2019 in Sweden [11-14].

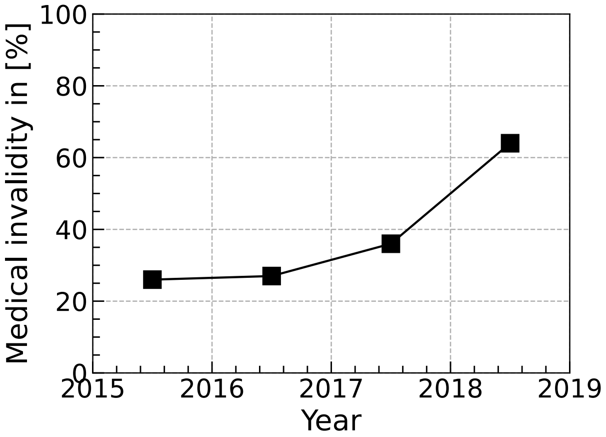


Figure 3. Percentage of medical invalidity of workers who received severe work-related injuries due to fire, explosion, welding, electricity or cold, during 2015 and 2019 in Sweden [11-14].

After going through the most recent dust explosion incidents in Sweden, the question arises: why are there so many dust explosions? We believe that the main reason is the lack of knowledge in the complicated combustion process during a dust explosion and the lack of numerical tools for designing explosion protection systems.

Dust explosion is a complicated physical and chemical process, when very fine combustible particles well mixed with air in a confined equipment are ignited, resulting in a violent and explosive burning. Once the dust explosion occurs, the high-pressure waves, hot flames and extremely radiative heat may cause loss of lives and huge economic consequences.

1.2 The open-source code OpenFOAM

The next question will be: how could we reduce the consequences of dust explosions? One important solution would be having access to high-fidelity and well-validated models and an efficient numerical tool. Specifically, the numerical tool can be used to design explosion protection system with complicated geometries at the process plants where the standards are not applicable. Moreover, the numerical tool can be used to investigate the incidents which have already happened to avoid similar incidents.

A suitable platform for developing dust explosion models is the code OpenFOAM (Open Field Operation and Manipulation). It is a free, open-source general-purpose Computational Fluid Dynamics (CFD) software package mainly for simulating thermodynamics, fluid dynamics, and chemical reactions. The code solves various partial differential equations using finite volume method on unstructured mesh. OpenFOAM is licensed under GNU General Public License (GPL) v3 which means users have the greatest freedom in adapting the code. More specifically, the users are free to adopt the software for any purpose (commercial or non-commercial); they are free to make change to the software; they are free to share the software with the changes they have made.

OpenFOAM has attracted growing interests from both industries and academies since its release in 2004. Researchers are strongly interested in access to source codes in order to develop and to implement new models and to easily exchange knowledge and experience with each other. Moreover, OpenFOAM has a very attractive feature; it is written in object-oriented language C++. Accordingly, solvers, written using the OpenFOAM classes, closely resemble the corresponding partial differential equations. Furthermore, OpenFOAM has inbuilt parallelization. That means parallelization is integrated at a low level, e.g., new application does not need parallel-specific coding since it runs in parallel by default. At the same time, OpenFOAM has mainly been used by the academia due to the challenges such as (i) limited or no documentation, (ii) limited or no code and model verification or validation, (iii) no user-interface, (iv) requirement of solid programming skill, (v) limited or no support, etc.

The overall OpenFOAM structure is shown in Figure 4. The workflow of using OpenFOAM is similar to that of a conventional CFD program, and it includes pre-processing, solving and post-processing. First, OpenFOAM has its own mesh generation utilities, such as blockMesh, snappyHexMesh and cfmesh. Such meshing utilities have limitations of either too simple geometry or less user-friendly, which limits the usage of OpenFOAM in industrial applications with complicated geometry and moving mesh. However, OpenFOAM mesh is compatible with the most of the common commercial CFD programs. After the computational mesh is ready, there are various kinds of solvers designed to study specific computational continuum mechanics. OpenFOAM offers a set of libraries which are dynamically linked to the solvers, and the libraries serve as the source code of physical models. Finally, post-processing of the computed results especially for data visualization can be achieved using both an open-source program ParaView, and commercial programs, e.g., EnSight and Tecplot. Moreover, there are utilities for data acquisition as well.

OpenFOAM is available for the Linux, Mac and Windows operating systems. Currently the owner of OpenFOAM - ESI releases both source code and pre-compiled binaries for those system, and users can freely download the source code from the Internet. Usually, compilation of OpenFOAM from source code requires knowledge of Linux operation system and capability to work in terminals using commands. Moreover, the installation of ParaView, which is an open-source data visualization application released together with OpenFOAM, is available for both Linux and Windows operation systems.

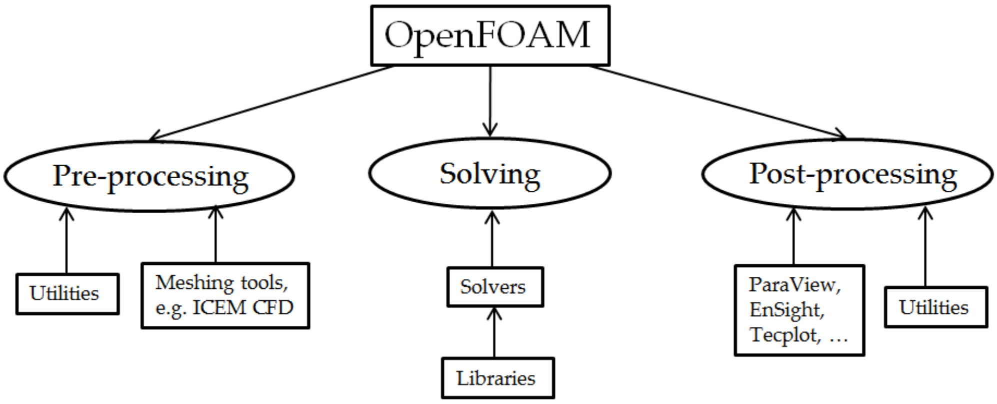


Figure 4 Overview of OpenFOAM structure.

1.3 Project aim

For the above reasons, this project aims at (i) development of physics-based and well-validated models which address the important combustion phenomena such as flame expansion, turbulence generation in the flame and flame acceleration in dust explosions, (ii) development of a well-verified and an efficient numerical tool based on OpenFOAM for predicting consequences of dust explosions and (iii) simulation of large-scale dust explosions in the process industries.

2. Project implementation

The project focuses on the development of an open-source code for quantitatively predicting the dust explosion consequences in the process industry. The project has been carried out in a step-by-step fashion, (see the four Work Packages (WP) in Figure 5). Note that all the WPs have been jointly performed by RISE and Chalmers with valuable and continuous support from the reference group members.

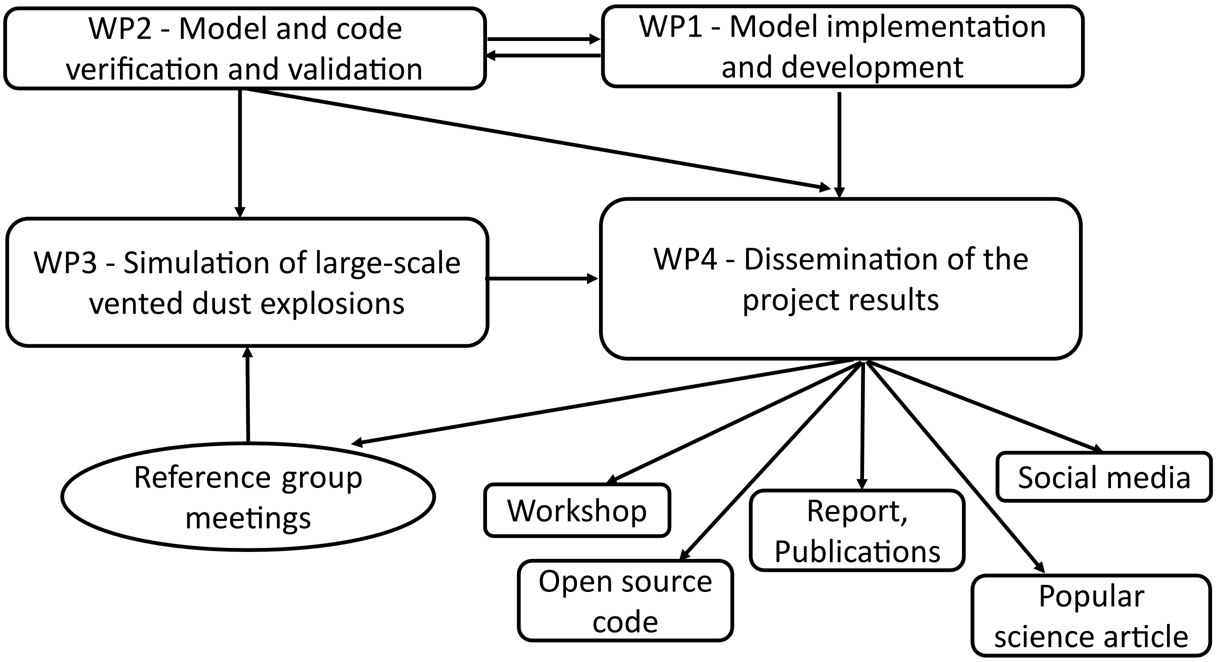

WP2 - Model and code verification and validation
WP1 - Model implementation and development
WP3 - Simulation of large-scale vented dust explosions
WP4 - Dissemination of the project results
Reference group meetings
Workshop
Report, Publications
Social media
Open source code
Popular science article

Figure 5. A schematic illustration of the project structure involving four work packages.

2.1 WP1 – Model implementation and development

In this work package, a so-called Flame Speed Closure (FSC) model has been implemented into the OpenFOAM package. The FSC model, developed at Chalmers addresses the influence of turbulence on combustion of premixed gases [15, 16]. There are two major reasons for choosing this method. First, a dust explosion resembles the turbulent burning of a gas cloud for very fine organic dust particles with high volatile content [17-19]. Second, the FSC model has been quantitatively validated against a representative set of experimental data obtained by various research groups from different premixed turbulent flames under a wide range of substantially different conditions (various fuels, equivalence ratios, temperatures, pressures, and turbulence characteristics) [15, 16]. Specifically, using **the same value of a single model constant**, the FSC model has been shown to quantitatively predict [20, 21]:

- dependencies of the mean radius of a statistically spherical flame kernel on time for various rms turbulent velocities and for various compositions of CH_4/air or C_3H_8/air mixtures,
- an increase in the observed turbulent flame speed as the flame kernel grows in the stoichiometric *iso*-C_8H_{18}/air mixture under elevated temperatures and pressures,
- mean structure of a statistically stationary, oblique, confined, lean CH_4-air turbulent flame stabilized by a hot jet in intense turbulence,
- mean shapes of open lean CH_4-air turbulent V-shaped flames characterized by different equivalence ratios,
- influence of bulk flow velocity, turbulence generation method, and pressure on the mean axial length and the mean axial thickness of confined preheated lean CH_4-air turbulent flames

stabilized due to abrupt expansion of a channel at elevated pressures + an axial profile of the Reynolds-averaged combustion progress variable in such a flame,

- mean structure of confined preheated lean C_3H_8-air flames stabilized by a bluff body,
- mean axial heights of lean CH_4-air weakly turbulent Bunsen flames at various equivalence ratios or inlet mass flow rates,
- axial profiles of a mean combustion progress variable obtained from axisymmetric weakly turbulent CH_4/CO_2/air Bunsen flames stabilized within a high-pressure combustion chamber using a pilot flame,
- burning velocities of H_2/CO/air turbulent premixed flames for different equivalence ratios, volume fraction of H_2, pressures, and turbulence characteristics,
- pressure growth due to stratified turbulent combustion in an optical Direct Injection Spark Ignition gasoline engine.

Furthermore, within the framework of the present project, the FSC model has been extended [22] to address self-acceleration of a large flame kernel due to the influence of combustion-induced thermal expansion on the upstream flow. A detailed description of the FSC model and its extended version is reported in Appendix I.

2.2 WP2 – Model and code verification and validation

In this work package, the code and the model have been verified by comparing the simulation results with the benchmark analytical solutions for statistically 1-dimensional planar flame and 3-dimensional spherical flame in “frozen” turbulence (see Appendix II for further details). Close agreement between simulation results and the analytical solutions shows that the model has been implemented correctly. Next, The FSC model and the so-extended code were validated against experimental data on corn starch dust explosions obtained in a small-scale fan-stirred vessel. This equipment installed in the Leeds University is well known in the combustion community [18]. This validation step ensures that the model reflects the physics of a dust explosion (see Appendix III for further details).

2.3 WP3 – Simulation of large-scale vented dust explosions

In this work package, the developed model and code is applied to simulate the vented corn starch dust explosions in an 11.5 m^3 vessel. The experiments were carried out at the REMBE® Research + Technology Center in Brilon, Germany within a research project in IND EX® (Intercontinental Association of Experts for Industrial Explosion Protection) [23, 24]. As mentioned in WP1, the dust explosion model has been further extended by including the effect of self-acceleration of a large-scale flame kernel. Comparison between the simulations and the experiments shows excellent agreement, which indicates that the developed model and code are promising and deserves further study in simulations of other large-scale dust or gaseous explosions.

2.4 WP4 – Dissemination of the project result

The project results are disseminated through various channels, including project kick-off meeting, reference group meetings, final workshop, journal and conference publications, presentations, reports, and social media. A detailed description of the spreading of the project results can be found in Section 5.

3. Achieved results

The three major project results are as follows:

- Validation of the FSC combustion model for small-scale dust explosions and extension of the model for large-scale dust explosions [22, 25].
- Development of an open-source CFD tool *FSCDustFoam* for unsteady 3-D simulations of small-scale and large-scale dust explosions [26, 27].
- Validation of the developed CFD tool *FSCDustFoam* in simulations of small-scale [25] and large-scale [22] experiments with dust explosions. The validation yielded excellent agreement with the data measured in Leeds University [18] and Rambe in an IND EX® project [24], respectively.

In the present section, only the most important results are briefly presented. The reader is referred to Appendices II – IV for detailed discussion of the project results regarding model and code verification, validation, and comparison with experiments. Furthermore, detailed information regarding simulation case setups, mesh generation, model implementation, as well as tips and tricks in using the *FSCDustFoam* can be found in Appendices V-XI.

3.1 Model and code validation

First, the model was validated by simulating small-scale Leeds experiments [18], which were performed under well-defined conditions. In the cited work [18], turbulent burning velocities of corn starch flames were measured using high-speed Schlieren technique. A more detailed description of the experimental and numerical setups can be found in Appendix III and in our recent publication [25]. Figures 6 and 7 summarize the most important findings of the model validation. The numerical results show that the model well predicts (i) an increase in the apparent turbulent flame speed by the rms turbulent velocity at moderate turbulence and (ii) a slow increase in the flame speed with growing mean flame radius. These results indicate that the adapted dust explosion model is an appropriate building block for developing an advanced numerical tool for CFD research into explosions of fine dust particles with high volatile contents. Moreover, the project results summarized in Figures 6 and 7 validate the developed CFD tool *FSCDustFoam.*

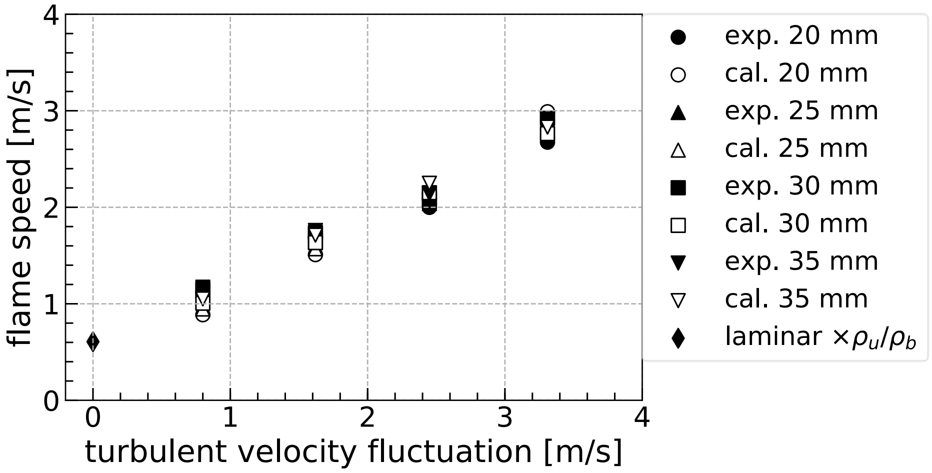


Figure 6 Comparison of computed (open symbols) and measured (filled symbols) mean flame speeds. The diamond symbol represents the laminar flame speed multiplied with the density ratio.

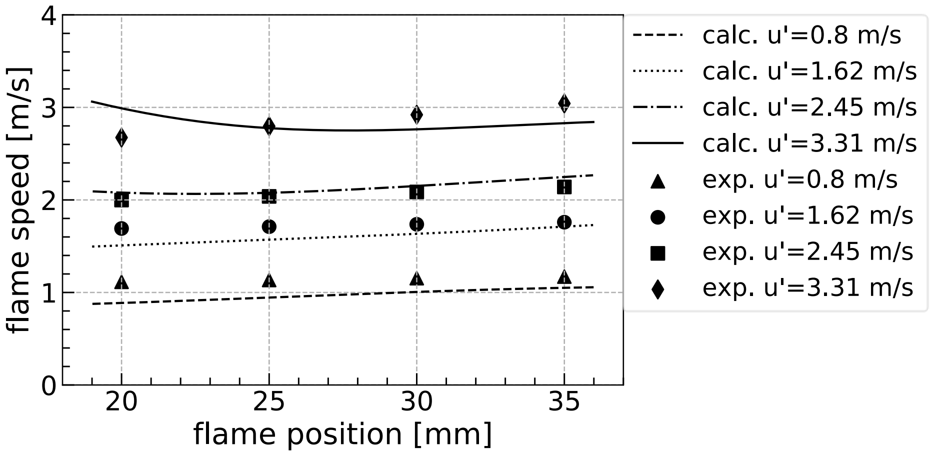


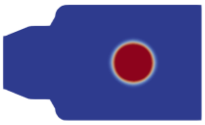
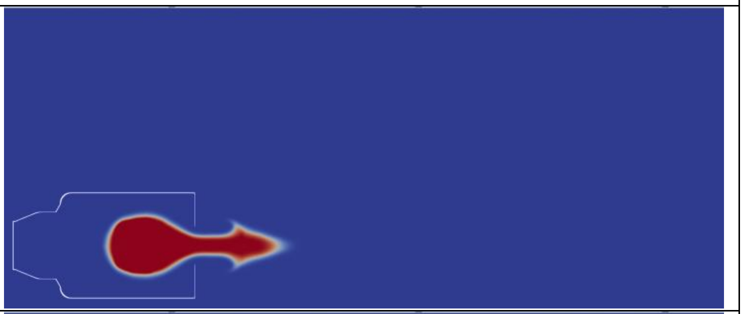
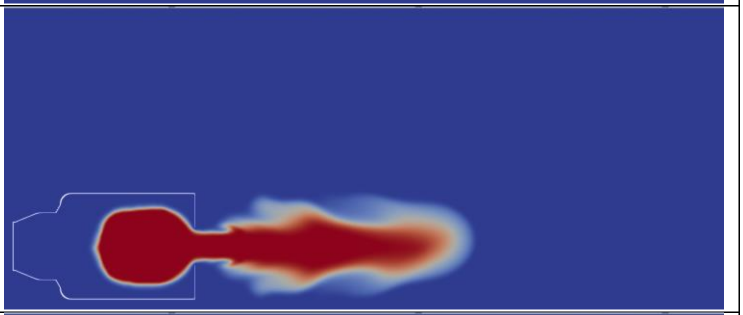
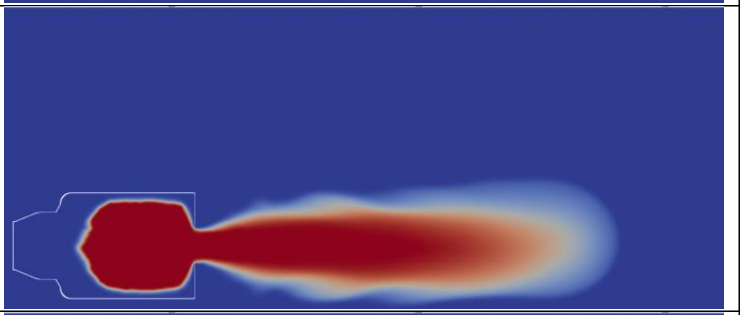
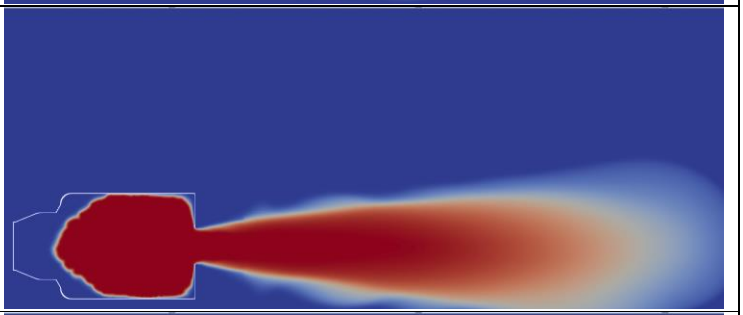
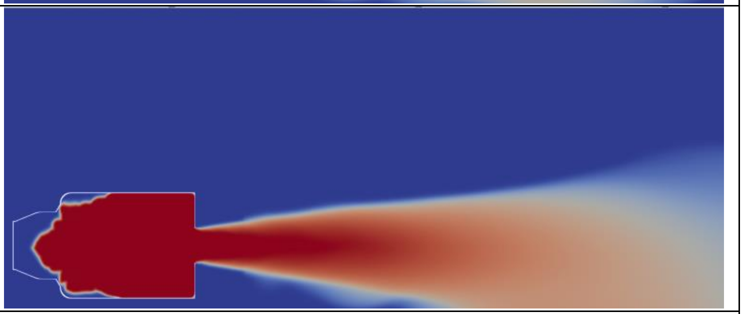
Figure 7 Computed (lines) and measured (symbols) mean flame speeds vs. mean flame position.

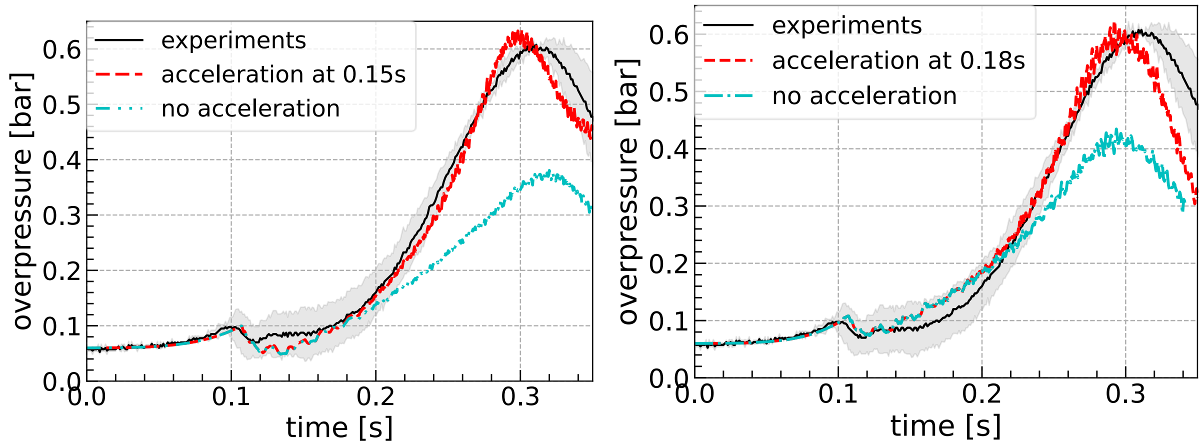
3.2 Simulation of vented dust explosions in an 11.5 m^3 vessel

The FSC model and the developed CFD tool were further validated by simulating large-scale industrial dust explosion experiments [22]. The model input parameters, the experimental and numerical setups are reported in detail in Appendix IV. Only the most important results are briefly presented here.

Table 1 shows evolution of the computed fields of the Reynolds-averaged combustion progress variable at different time instants. The explosion overpressure-time traces computed using the extended FSC model and both k-omega-SST [28-30] and realizable k-epsilon [31, 32] turbulence models, which are implemented in the standard version of OpenFOAM, agree reasonably well with the experimental data (c.f. curves shown in red dashed and black solid lines in Figure 8). The experimental data was obtained by averaging results of two dust explosion tests. Accordingly, the grey areas in Figure 8 cover mean values ± the standard derivations based on the two tests data. The use of the conventional FSC model without the newly introduced acceleration factor T_2 in Eq. (Appendix I.18) yields significantly underpredicted overpressure when compared to the experimental data, cf. cyan dash-dotted lines with black solid lines in Figure 8. The use of other turbulence models implemented into OpenFOAM did not allow us to predict the experimental results either, see Appendix IV.VI. All in all, the project results summarized in Figure 8 indicate the most appropriate turbulence models, validate the extended FSC model, and validate the developed CFD tool *FSCDustFoam* in the challenging case of a large-scale dust explosion. Moreover, these results show importance of the extension of the FSC model, developed within the framework of the present project.

Table 1 Mean combustion progress variable fields computed at different time instants.

Time [s]	Reynolds-Averaged combustion progress variable [-]
0.1	
0.15	
0.2	
0.25	
0.3	
0.35	
scale	0 1



(a), k-omega-SST, u'=0.75 m/s *(b), realizable k-epsilon, u'=0.50 m/s*

Figure 8 Comparison between measured (mean value in the black solid line and the standard derivation in the grey area) and simulated (with and without flame accelerations) explosion overpressures.

4. Implemented efforts to enable the practical usage of project results

4.1 Reference group meetings

Totally nine reference group meetings with three meetings per year have been organized throughout the three-year period of this project. The reference group members represent the whole value chain including industries such as Scandbio, Göteborg Energi and Lantmännen, experienced safety consultants and safety solution suppliers such as Brandskyddslaget, Fagerberg, AVS, IEP Technologies, Firefly, organizations such as Intressentföreningen för Processäkerhet (IPS), the global combustible dust safety knowledge platform DustEx Research, and academy such as Lund University. During these meetings, the project results were presented, and valuable inputs from reference group members were considered and implemented in the project.

4.2 Open access publications and source code with detailed documentation

Project results were published as gold open access in two peer-reviewed journals [22, 25] to enable a free access. The developed code *FSCDustFoam* with a tutorial of vented dust explosion has been released to a public platform Zenodo for sharing open science to allow for free downloading and usage of the project results [27]. The *FSCDustFoam* code available to the public on December 21st, 2021, has received 129 views and 87 downloads (until February 24th, 2022). Detailed documentations of the model and code development, case setups are also described in Appendices in this report.

4.3 The practical usages of the project results

The well-verified and well-validated numerical tool is useful for developing better experiments, learning more from the test results, and applying the test results to more scenarios. The *FSCDustFoam* is especially useful for performing parametric studies to reduce the huge costs associated with large-scale dust explosion tests. Furthermore, the *FSCDustFoam* code is especially useful in scenarios which are not covered in the dust explosion venting standards, i.e., EN 14491:2012 [33] and NFPA 68:2018 [34]. Two parametric studies are reported in the following sections, including the effect of vent size and the ignition location on the dust explosion process. A more detailed discussion of such effects can be found in a conference paper [35].

4.3.1 Effect of vent size

In this section, the developed numerical tool *FSCDustFoam* is applied to study the effect of vent size on the maximum reduced explosion overpressure, p_{red}. The detailed experimental and numerical setups can be found in the Appendix IV and a conference paper [35]. Both the standards (i.e., EN 14491:2012 [33] for Europe and NFPA 68:2018 [34] for the US) and the simulations show that the maximum reduced explosion overpressure decrease with an increase of vent area (see Figure 9). Furthermore, both the estimations done using the standards and the simulations closely follow each other, with the simulation results being slightly closer to the estimations for the NFPA 68:2018 standard. For the current case, the calculations using the NFPA 68:2018 standard are slightly less conservative for a vent area larger than 0.35 m^2 as compared to the results yielded by the EN 14491:2012. Similar finding was reported by Tascon et al. [36]. However, for a vent area smaller than 0.35 m^2, EN 14491:2012 standard yields a less conservative result than that by NFPA 68:2018 standard.

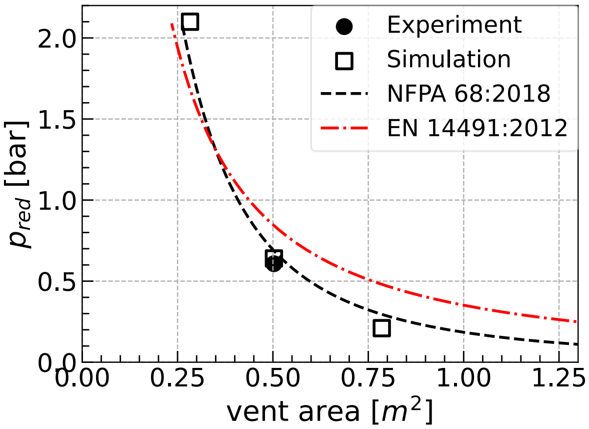


Figure 9 Comparison of the maximum reduced explosion overpressure versus vent area among simulations, standards, and 11.5 m^3 experiments. NFPA 68:2018 invokes vessel length-to-diameter ratio L/D = 1.86, and EN 14491:2012 invokes L/D = 1.41.

4.3.2 Effect of ignition location

The developed model and tool can be used not only for parametric studies covered in the standards (see the previous example), but can also be used for parametric studies that are partly addressed or not addressed in the standards. Ignition location is a factor which can affect vented explosions.

In this study, four ignition positions were studied, i.e., back, center, upper and lower, as illustrated in Figure 10. All the ignition points are chosen in a vertical plane cutting through the center of the vessel. The back ignition point is located at 0.5 m from the back wall of the vessel. The upper and lower ignition positions are located at 0.6 m away from the horizontal center line of the vessel (see Fig. 6). Figure 7 compares the computed explosion overpressure for four ignition locations using the original FSC model, i.e., Eq. Appendix I.18 without the term T_2 or the extended model, which allows for the acceleration effect, see Eq. (Appendix I.18). Due to the lack of experimental data obtained for the ignition locations other than the center ignition, the flame acceleration timing is chosen by assuming that the flame position characterized by $\bar{c} = 0.5$ reaches a distance of 1.85 m away from the vent opening in all cases. Accordingly, the t_{flacc} for back, center, upper and lower ignition positions is equal to 0.193 s, 0.15 s, 0.143 s and 0.143 s, respectively.

The simulations clearly show that the back ignition position yields the highest explosion overpressure, whereas the upper and lower ignition positions yield the lowest overpressure. This is also reflected in the calculations using the EN 14491:2012 standard (c.f. red dotted line and cyan dash dotted line in Fig. 11). Note that the effect of ignition location can be considered in the EN 14491:2012 standard by varying *L/D*, in which *L* is the flame length inside the vessel depending on the ignition location. The explosion overpressure simulated using the extended FSC model with the flame acceleration term T_2 yields substantially higher values than those computed using the original FSC model for the center, upper and lower ignition positions, but not for the back ignition position (c.f. slash filled bars and backslash filled bars in Fig. 11).

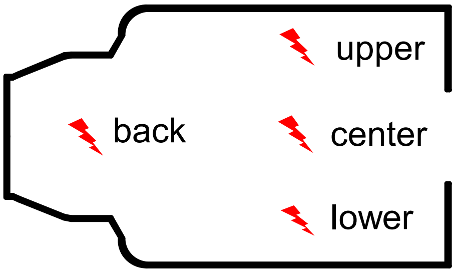


Figure 10 Illustration of the ignition locations.

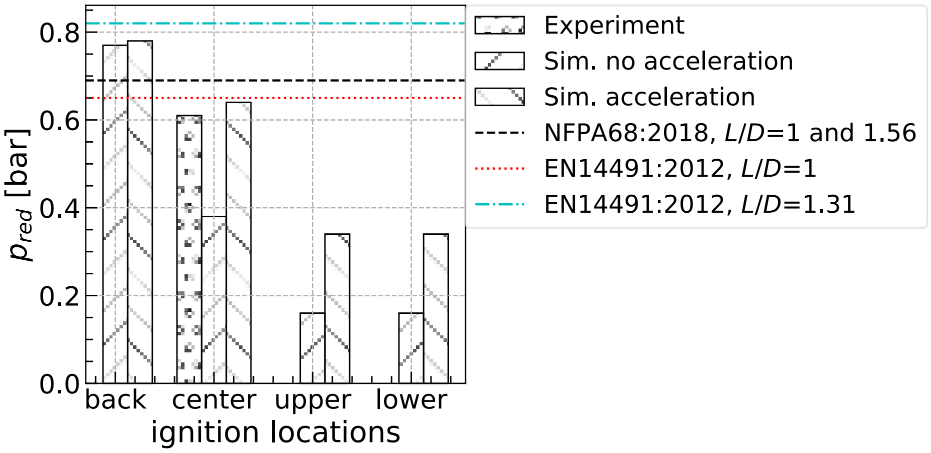


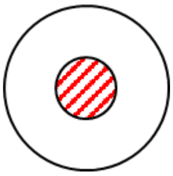
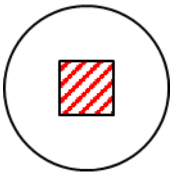
Figure 11 Maximum reduced explosion overpressures versus ignition locations.

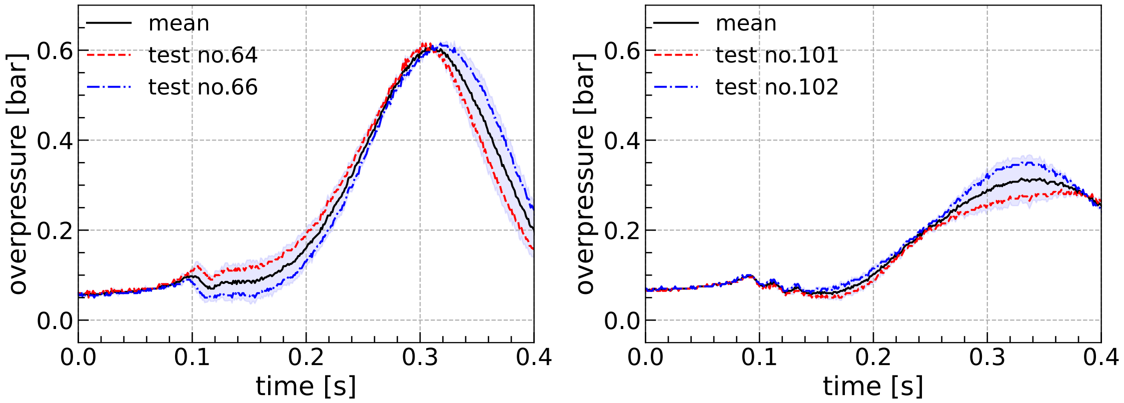
4.4 Challenges and future work

4.4.1 Challenges to the numerical tool

In spite of the successful development of the numerical tool *FSCDustFoam*, challenges still remain. One of the remaining challenges is that a square-shaped vent opening was more effective in reducing the explosion overpressure as compared to a circular-shaped vent opening in both small and large-scale experiments [24, 37]; see vent opening illustrations in Table 2. Figure 12 shows the measured mean explosion overpressure measured using a circular-shaped vent opening is 93 % higher than that obtained using a square-shaped vent opening. The current version of *FSCDustFoam* code does not capture this effect (see Figure 13), which is not addressed by the current venting standards, EN 14491:2012 [33] or NFPA 68:2018 [34] either.

Table 2 Vent opening shapes.

Shapes	Circular	Square
Illustrations		



(a), circular vent opening

(b), square vent opening

Figure 12 Measured explosion overpressures (based on two trials) in an 11.5 m^3 vessel using different vent opening shapes.

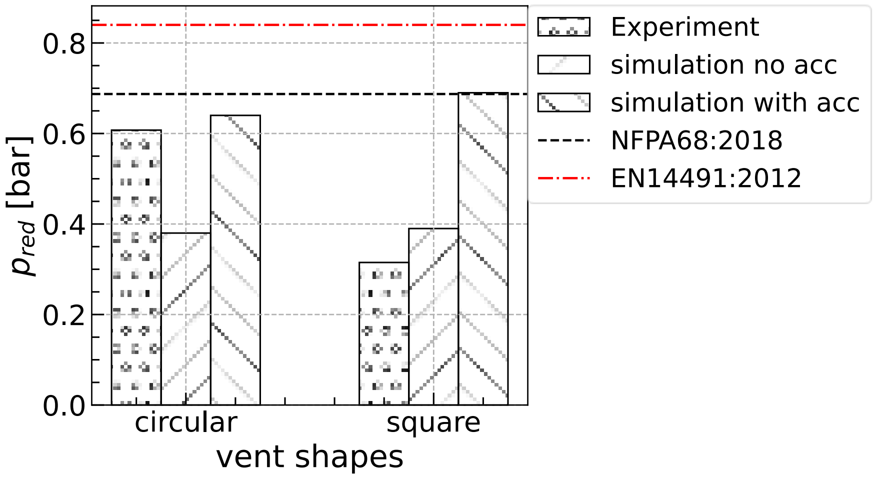


Figure 13 Maximum reduced explosion overpressures versus vent shapes.

4.4.2 Future work

Based on the project results, we propose the following future work directions:

- Apply the *FSCDustFoam* code to typical dust explosion incidents scenarios, i.e., dust explosions in complex geometries such as interconnected vessels.
- Further improve the *FSCDustFoam* code by addressing the effect of vent shapes on the dust explosion venting process and by comparing simulations with large-scale dust explosion experiments in complex geometries.

5. Publications, presentations, and other dissemination within the framework of the project

5.1 Publications and presentations

Apart from two project part reports and this final report, the project results were presented in:

1. **Peer-reviewed journal publications:**

- C. Huang, A.N. Lipatnikov, and K. Nessvi, Unsteady 3-D RANS simulations of dust explosion in a fan stirred explosion vessel using an open source code. Journal of Loss Prevention in the Process Industries, 2020. 67: 104237.
- C. Huang, M. Bloching, and A.N. Lipatnikov, A vented corn starch dust explosion in an 11.5 m3 vessel: Experimental and numerical study. Journal of Loss Prevention in the Process Industries, 2022. 75: 104707.

2. **Peer-reviewed conference publications:**

- C. Huang, and A.N. Lipatnikov, Modelling of vented corn starch dust explosion using OpenFOAM, in The 16th OpenFOAM Workshop (OFW16), 7-11 June 2020, Dublin, Ireland.
- C. Huang, and A. Lipatnikov. Modelling of premixed turbulent combustion of cornflour dust-air cloud using OpenFOAM. in 38th International Symposium on Combustion, Work in Progress Poster. 2021. Adelaide, Australia.
- C., Huang, M. Bloching, and A.N. Lipatnikov, Vented dust explosions: comparison among experiments, simulations and standards, in the 10th International Seminar on Fire and Explosion Hazards, Oslo, 22-27 May, 2022. (Full paper submitted).
- C., Huang, M. Bloching, and A.N. Lipatnikov, An experimental and a numerical study of a vented dust explosion in an 11.5 m3 vessel, in the 14th International Symposium on Hazards, Prevention, and Mitigation of Industrial Explosions, July 11-15, 2022. (Full paper submitted).

3. **Oral presentations:**

- C. Huang, and A.N. Lipatnikov, Modelling of cornflour dust explosion using an open source code, presented at the Global Dust Safety Conference in February 24 - 28, 2020.
- C. Huang, and A.N. Lipatnikov, Modelling of a vented corn starch dust explosion using an open source code, presented at the Global Dust Safety Conference in March 1 - 3, 2021.
- C. Huang, and A.N. Lipatnikov, A new approach based on an open source code for modelling dust explosions and future challenges, presented at the web seminar Brandfarlig vara 2021, November 25th, 2021.
- C. Huang, and A.N. Lipatnikov, Open source dust explosion research, An open digital Workshop on *Dust explosion safety for the process industry*, Jan 21st, 2022.

5.2 Other spreading channels

- A **project homepage** (in Swedish and English) [26] has been created for spreading the project results. The project homepage summarizes the project description, reports, and publications.
- An open **digital workshop** [38] was organized by RISE on the January 21st, 2022, involving speakers from industries such as Scandbio, academies such as Lund University and University of Bergen, experienced safety consultants such as Brandskydsslaget and Vysusgroup, organizations such as RISE and the global combustible dust safety knowledge platform DustEx Research, and AFA Försäkring. There were totally 35 registered participants who joined this online event.

- The open-source code *FSCDustFoam* was published with documentations and tutorial for free downloading at a platform called Zenodo for sharing open research [27].
- A popular science article (in Swedish and English) of this project was published on the project homepage [26].
- The project results were spread to external partners through the following collaboration activities:
  - The Swedish knowledge digital platform safedustexplosion.org for sharing the project results.
  - Collaboration with the global combustible dust safety knowledge platform DustEx Research via oral presentations of project results twice at the digital conferences organized by DustEx Research.
  - Collaboration activities was carried out for performing a comparison study of dust explosion using *FSCDustFoam* and FLACS-DustEx with joint support from University of Bergen and GexCon (a fire and explosion safety consultant company). The collaboration was paused due to the parental leave of a collaborator at GexCon.
- The project results were distributed through social media such as Research Gate, Linkedin and Youtube. Several examples are shown as follows:
  - This project and the relevant publications were published on the Research gate (https://www.researchgate.net/project/Modelling-of-dust-explosions-for-the-process-industry).
  - An animation showing the simulated vented dust explosion was posted on Youtube (https://youtu.be/LER0oACzizw).

References

1. *Flera bränder har drabbat foderfabrik i Lidköping*. Available from: https://www.skaraborgsnyheter.se/lidkoping/flera-brander-har-drabbat-foderfabrik-i-lidkoping/.
2. *Brand i fabrik i Lidköping*. 2020; Available from: https://www.skaraborgsnyheter.se/lidkoping/brand-i-fabrik-i-lidkoping/.
3. Peter Wikström, A., *Personlig kommunikation* H. Persson.
4. *Dammexplosion i stålindustri*. Available from: https://www.sla.se/2020/11/03/dammexplosion-i-stalindustri/.
5. Lindsten, J., *Dammexplosion vid hantering av metalldamm*. in Brandfarlig vara 2019, Nov. 27-28 2019, Malmö.
6. *Brand på Rya HVC*. 2017; Available from: https://www.goteborgenergi.se/Om_oss/Pressrum/Nyheter?DisplayNewsItem=36992077.
7. *Dammexplosion i Huskvarna*. 2017; Available from: https://www.jnytt.se/article/dammexplosion-i-huskvarna/.
8. *Flera skadade efter explosion vid en arbetsplats i Trångsund*; Available from: https://www.svt.se/nyheter/lokalt/stockholm/flera-skadade-efter-explosion-vid-en-arbetsplats-i-trangsund.
9. *Explosion i spannmålstork*.; Available from: https://www.skovdenyheter.se/article/larm-om-ladugardsbrand/.
10. Cloney, C., *2020 COMBUSTIBLE DUST INCIDENT REPORT*. 2020.
11. *Allvarliga arbetsskador och långvarig sjukfrånvaro*. 2018, AFA Försäkring.
12. *Allvarliga arbetsskador och långvarig sjukfrånvaro*. 2019, AFA försäkring.
13. *Allvarliga arbetsskador och långvarig sjukfrånvaro*. 2020, AFA försäkring.
14. *Allvarliga arbetsskador och långvarig sjukfrånvaro*. 2021, AFA försäkring.
15. Lipatnikov, A.N. and Chomiak J., *A simple model of unsteady turbulent flame propagation*. SAE transactions, Section 3, Journal of Engines, 106, pp. 2441-2452, 1997.

16. Lipatnikov, A.N. and Chomiak, J., *Turbulent flame speed and thickness: phenomenology, evaluation, and application in multi-dimensional simulations.* Progress in energy and combustion science, 2002. **28**(1): p. 1-74.
17. Bradley, D., Habik, S. El-Din, Swithenbank, J.R., *Laminar burning velocities of CH4—air-graphite mixtures and coal dusts.* Symposium (International) on Combustion, 1988. **21**(1): p. 249-256.
18. Bradley, D., Chen, Z., Swithenbank, J., *Burning rates in turbulent fine dust-air explosions.* Symposium (International) on Combustion, 1989. **22**(1): p. 1767-1775.
19. Bradley, D., Dixon-Lewis, G., Habik, S. El-Din, *Lean flammability limits and laminar burning velocities of CH4-air-graphite mixtures and fine coal dusts.* Combustion and Flame, 1989. **77**(1): p. 41-50.
20. Lipatnikov, A.N., *Fundamentals of premixed turbulent combustion*. 2012: CRC Press.
21. Lipatnikov, A.N., *RANS simulations of premixed turbulent flames*, in *Modeling and Simulation of Turbulent Combustion*. 2018, Springer. p. 181-240.
22. Huang, C., Bloching, M., Lipatnikov, A.N., *A vented corn starch dust explosion in an 11.5 m3 vessel: Experimental and numerical study.* Journal of Loss Prevention in the Process Industries, 2022. **75**: p. 104707.
23. *IND EX (Intercontinental Association of Experts for Industrial Explosion Protection)*; Available from: https://www.ind-ex.info/en.
24. Bloching, M., *Influence of the Explosion Relief Device Geometry on its Venting Efficiency*. 2018. Rembe test report Nr: 17280018 Rev. 0.
25. Huang, C., Lipatnikov, A.N., Nessvi, K., *Unsteady 3-D RANS simulations of dust explosion in a fan stirred explosion vessel using an open source code.* Journal of Loss Prevention in the Process Industries, 2020. **67**: p. 104237.
26. Huang, C. *Utveckling av ett beräkningsverktyg avseende dammexplosioner*. Available from: https://www.ri.se/sv/vad-vi-gor/projekt/utveckling-av-ett-berakningsverktyg-avseende-dammexplosioner.
27. Huang, C. and Lipatnikov, A.N., *FSCDustFoam - a solver for premixed/partially-premixed combustion and explosion of dust and gas based on OpenFOAM*. 2021; Available from: https://doi.org/10.5281/zenodo.5795161.
28. Hellsten, A. *Some improvements in Menter's k-omega SST turbulence model*. in *29th AIAA, Fluid Dynamics Conference*. 1998.
29. Menter, F. and T. Esch. *Elements of industrial heat transfer predictions*. in *16th Brazilian Congress of Mechanical Engineering (COBEM)*. 2001.
30. Menter, F.R., Kuntz, M., Langtry, R., *Ten years of industrial experience with the SST turbulence model.* Turbulence, heat and mass transfer, 2003. **4**(1): p. 625-632.
31. Shih, T.-H., et al., *A new k-epsilon eddy viscosity model for high Reynolds number turbulent flows: model development and validation.* NASA Sti/recon Technical Report N, 1994. **95**: p. 11442.
32. Shih, T., et al., *A new k-epsilon eddy-viscosity model for high Reynolds number turbulent flows-model development and validation. Computers Fluids, 24 (3): 227-238.* 1995.
33. EN14491, *EN14491 Dust explosion venting protective systems*. 2012.
34. NFPA68, *NFPA 68 Standard on Explosion Protection by Deflagration Venting*. 2018.
35. Huang, C., Bloching, M., Lipatnikov. A.N., *Vented dust explosions: comparison among experiments, simulations and standards*. in *the 10th International Seminar on Fire and Explosion Hazards*. 2022. Oslo. (Full paper submitted).
36. Tascón, A., Aguado, P.J., Ramírez, A., *Dust explosion venting in silos: A comparison of standards NFPA 68 and EN 14491.* Journal of Loss Prevention in the Process Industries, 2009. **22**(2): p. 204-209.
37. Bloching, M., *Prüfungen an Druckentlastungseinrichtungen mit unterschiedlichen Längen / Breiten Verhältnis*. 2020. Rembe test report Nr: 200001638.
38. Huang, C. *Digital workshop: Dust explosion safety for the process industry*. Available from: https://www.ri.se/en/event/dust-explosion-safety-for-the-process-industry.
39. Libby, P.A. and Bray, K., *Variable density effects in premixed turbulent flames.* AIAA Journal, 1977. **15**(8): p. 1186-1193.
40. Bray, K.N.C. and J.B. Moss, *A unified statistical model of the premixed turbulent flame.* Acta Astronautica, 1977. **4**(3): p. 291-319.

41. Launder, B.E. and Spalding, D.B., *Lectures in mathematical models of turbulence.* 1972.
42. Zimont, V., *Theory of turbulent combustion of a homogeneous fuel mixture at high Reynolds numbers.* Combustion, Explosion and Shock Waves, 1979. **15**(3): p. 305-311.
43. Zimont, V.L., *Gas premixed combustion at high turbulence. Turbulent flame closure combustion model.* Experimental Thermal and Fluid Science, 2000. **21**(1): p. 179-186.
44. Prudnikov, A. and Raushenbakh, B., *Burning of homogeneous fuel-air mixtures in a turbulent flow.* Physical principles of the working process in combustion chambers of jet engines, 1967. **5**: p. 244-236.
45. Zimont, V. and Lipatnikov, A., *A numerical model of premixed turbulent combustion of gases.* Chem. Phys. Rep, 1995. **14**(7): p. 993-1025.
46. Zimont, V.L., Lipatnikov, A.N., *To computations of the heat release rate in turbulent flames.* Dokl. Phys. Chem., 1993. **332**: p. 592–594.
47. Hinze, J.O., *Turbulence, second ed.* 1975, New York: McGraw-Hill.
48. Taylor, G.I., *Statistical theory of turbulence.* Proc. Roy. Soc. Lond. A, 1935. **151**: p. 65-78.
49. Huang, C., et al., *Application of flame speed closure model to RANS simulations of stratified turbulent combustion in a gasoline direct-injection spark-ignition engine.* Combustion Science and Technology, 2016. **188**(1): p. 98-131.
50. Lipatnikov, A. and Chomiak, J., *A theoretical study of premixed turbulent flame development.* Proceedings of the Combustion Institute, 2005. **30**(1): p. 843-850.
51. Lipatnikov, A.N. and Chomiak, J., *Self-similarly developing, premixed, turbulent flames: A theoretical study.* Physics of Fluids, 2005. **17**(6): p. 065105.
52. Lipatnikov, A.N., *Testing premixed turbulent combustion models by studying flame dynamics.* International Journal of Spray and Combustion Dynamics, 2009. **1**(1): p. 39-66.
53. Driscoll, J.F., *Turbulent premixed combustion: Flamelet structure and its effect on turbulent burning velocities.* Progress in Energy and Combustion Science, 2008. **34**(1): p. 91-134.
54. Tamadonfar, P. and Gülder, Ö.L., *Flame brush characteristics and burning velocities of premixed turbulent methane/air Bunsen flames.* Combustion and Flame, 2014. **161**(12): p. 3154-3165.
55. Han, D., et al., *Experimental study of CO2 diluted, piloted, turbulent CH4/air premixed flames using high-repetition-rate OH PLIF.* Combustion and Flame, 2018. **193**: p. 145-156.
56. Karlovitz, B., Denniston, Jr, W.D., Wells F.E., *Investigation of Turbulent Flames.* The Journal of Chemical Physics, 1951. **19**(5): p. 541-547.
57. Sponfeldner, T., et al., *The structure of turbulent flames in fractal- and regular-grid-generated turbulence.* Combustion and Flame, 2015. **162**(9): p. 3379-3393.
58. Lipatnikov, A. and Chomiak, J., *Global stretch effects in premixed turbulent combustion.* Proceedings of the Combustion Institute, 2007. **31**(1): p. 1361-1368.
59. Lipatnikov, A. and Chomiak, J., *Turbulent burning velocity and speed of developing, curved, and strained flames.* Proceedings of the Combustion Institute, 2002. **29**(2): p. 2113-2121.
60. Bradley, D., et al., *Turbulent burning velocity, burned gas distribution, and associated flame surface definition.* Combustion and Flame, 2003. **133**(4): p. 415-430.
61. Lipatnikov, A.N. and Chomiak, J., *Effects of premixed flames on turbulence and turbulent scalar transport.* Progress in Energy and Combustion Science, 2010. **36**(1): p. 1-102.
62. Sabelnikov, V.A. and Lipatnikov, A.N., *Recent advances in understanding of thermal expansion effects in premixed turbulent flames.* Annual Review of Fluid Mechanics, 2017. **49**(1): p. 91-117.
63. Bauwens, C.R., Chaffee, J., Dorofeev, S., *Experimental and numerical study of methane-air deflagrations in a vented enclosure.* Fire Safety Science, 2008. **9**: p. 1043-1054.
64. Bauwens, C.R., Chaffee, J., Dorofeev, S.B., *Vented explosion overpressures from combustion of hydrogen and hydrocarbon mixtures.* International Journal of Hydrogen Energy, 2011. **36**(3): p. 2329-2336.
65. Bauwens, C.R., Bergthorson, J.M., Dorofeev, S.B., *Experimental study of spherical-flame acceleration mechanisms in large-scale propane–air flames.* Proceedings of the Combustion Institute, 2015. **35**(2): p. 2059-2066.
66. Gostintsev, Y.A., Istratov, A., Shulenin, Y.V. *Self-similar propagation of a free turbulent flame in mixed gas mixtures.* Combustion, Explosion and Shock Waves, 1988. **24**(5): p. 563-569.

67. Bradley, D., Cresswell, T., Puttock, J., *Flame acceleration due to flame-induced instabilities in large-scale explosions.* Combustion and Flame, 2001. **124**(4): p. 551-559.

68. Bradley, D., *Instabilities and flame speeds in large-scale premixed gaseous explosions.* Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, 1999. **357**(1764): p. 3567-3581.

69. Gostintsev, Y.A., et al., *Self-turbulization of gas flames: An analysis of experimental results.* High Temperature, 1999. **37**(2): p. 282-288.

70. Taylor, G.I., *Diffusion by continuous movements.* Proc. London Math. Soc. Ser. 2, 1921. **20**(196-211).

71. Lipatnikov, A.N. and Chomiak, J., *Application of the Markstein number concept to curved turbulent flames.* Combustion Science and Technology, 2004. **176**(3): p. 331-358.

72. Bray, K., Libby, P.A., Moss, J., *Unified modeling approach for premixed turbulent combustion—Part I: General formulation.* Combustion and flame, 1985. **61**(1): p. 87-102.

73. Zel'dovich, Y.B. and Frank-Kamenetskii, D.A. *On the theory of steady flame propagation.* Dokl. AN SSSR, 1938. **19**(9): p. 693-695.

74. Abdel-Gayed, R., Al-Khishali, K., Bradley, D., *Turbulent burning velocities and flame straining in explosions.* Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 1984. **391**(1801): p. 393-414.

75. Lipatnikov, A.N. and Chomiak, J., *Modeling of Turbulent Flame Propagation*, in *1997 Annual Report*. 1997.

76. Yasari, E., *RANS Simulations of Interaction between Premixed Flame and Turbulence using OpenFOAM Library*. 2015: Chalmers University of Technology.

77. Bloching M., Lottermann J., Bunse R., Barth U., Penno S., *Influence of the Explosion Relief Device Geometry on its Venting Efficiency*, in *The ninth International Seminar on Fire and Explosion Hazards*. 2019: 21-26 April 2019, Saint Petersburg, Russia.

78. FreeCAD. Available from: https://www.freecadweb.org/.

79. Sattar, H., et al., *Turbulent Flames Speeds and Laminar Burning Velocities of Dusts using the ISO 1 m³ Dust Explosion Method.* Chemical Engineering Transactions, 2014. **36**: p. 157-162.

80. *NIST chemistry webbook CO2 gas phase thermochemistry data*. Available from: https://webbook.nist.gov/cgi/cbook.cgi?ID=C124389&Units=SI&Mask=1#Thermo-Gas.

81. *NIST chemistry webbook H2O gas phase thermochemistry data*; Available from: https://webbook.nist.gov/cgi/cbook.cgi?ID=C7732185&Mask=1.

82. Tan, I., et al., *Estimating the specific heat capacity of starch-water-glycerol systems as a function of temperature and compositions.* Starch-Stärke, 2004. **56**(1): p. 6-12.

83. Dahoe, A.E. and de Goey L.P.H., *On the determination of the laminar burning velocity from closed vessel gas explosions.* Journal of Loss Prevention in the Process Industries, 2003. **16**(6): p. 457-478.

84. Phylaktou, H., Gardner, C., Andrews, G., *Flame speed measurements in dust explosions*. in *Proceedings of the sixth international seminar on fire and explosion hazards*. 2010.

85. Proust, C. *Experimental determination of the maximum flame temperatures and of the laminar burning velocities for some combustible dust-air mixtures*. in *5. International Colloquium on Dust Explosions*. 1993.

86. Van Wingerden, K., Stavseng, L., Bergen, N., *Measurements of the laminar burning velocities in dust-air mixtures.* VDI-Berichte, 1996. **1272**: p. 553-564.

87. Skjold, T., *Simulating vented maize starch explosions in a 236 m3 silo.* Fire Safety Science, 2014a. **11**: p. 1469-1480.

88. Dahoe, A.E., Hanjalic, K., Scarlett, B., *Determination of the laminar burning velocity and the Markstein length of powder–air flames.* Powder Technology, 2002. **122**(2): p. 222-238.

89. Skjold, T., et al., *Simulating Dust Explosions with the First Version of DESC.* Process Safety and Environmental Protection, 2005. **83**(2): p. 151-160.

90. Skjold, T., et al., *Simulation of dust explosions in complex geometries with experimental input from standardized tests.* Journal of Loss Prevention in the Process Industries, 2006. **19**(2-3): p. 210-217.

91. Skjold, T., *Flame propagation in dust clouds. Numerical simulation and experimental investigation*. 2014, University of Bergen.

92. Dynamics, W. *Finite Volume Method: A Crash introduction*; Available from: http://www.wolfdynamics.com/wiki/fvm_crash_intro.pdf.

93. Wolf dynamics, *Running in parallel*. Available from: http://www.wolfdynamics.com/wiki/parallel.pdf.
94. Engys. *A Comprehensive Tour of snappyHexMesh*, in the 7th OpenFOAM workshop, 25th June, 2012.
95. Wolf dynamics, *Mesh generation using snappyHexMesh*. Available from: http://www.wolfdynamics.com/wiki/meshing_OF_SHM.pdf.

Acknowledgements

The authors would like to acknowledge AFA Försäkring for financial support of this project (grant number 180028). The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at HPC2N partially funded by the Swedish Research Council through grant agreement no. 2018-05973 and RISE Simulation Lab. The SNIC projects SNIC2021-22-217, SNIC2021-5-185 and SNIC2021-22-821 are acknowledged. Åke Sandgren and Erik Andersson at HPC2N are specially acknowledged. The authors would like to acknowledge IND EX® for providing the real scale test data of the IND EX® research project "Influence of the Explosion Relief Device Geometry on its Venting Efficiency".

Appendix I. Flame Speed Closure (FSC) model and its extended version

In this appendix, the FSC model equations, the basic features of the combustion model, and the derivation of the combustion progress variable equations, the benchmark analytical solutions adopted to verify the model implementation verification are documented.

Appendix I.I The FSC model equations

The FSC model (i) characterizes the thermochemical state of a reacting mixture in a flame using a single combustion progress variable c, which is equal to zero and unity in fresh reactants and equilibrium combustion products, respectively, (ii) invokes the following well-known Bray-Moss-Libby (BML) equations [39, 40]

$$\bar{\rho} = \frac{\rho_u}{1 + (\sigma - 1)\tilde{c}}, \qquad \bar{\rho}\tilde{c} = \rho_b \bar{c}, \tag{Appendix I.1}$$

and (iii) deals with the following transport equation

$$\frac{\partial \bar{\rho}\tilde{c}}{\partial t} + \nabla \cdot (\bar{\rho}\tilde{\mathbf{u}}\tilde{c}) = \nabla \cdot [\bar{\rho}(\kappa + D_t)\nabla\tilde{c}] + \rho_u U_t |\nabla\tilde{c}| + Q + \bar{\rho}W_{ign}, \tag{Appendix I.2}$$

for the Favre-averaged combustion progress variable $\tilde{c}$. Here, ρ is the density; $\sigma = \rho_u/\rho_b$ is the density ratio; t is the time; $\mathbf{u}$ is the flow velocity vector; κ is the molecular heat diffusivity of the mixture; D_t and U_t are the turbulent diffusivity and burning velocity, respectively; Q and W_{ign} are source terms discussed later, see Eqs. (Appendix I.9) and (Appendix I.11), respectively; over-lines designate the Reynolds average, while $\tilde{q} = \overline{\rho q}/\bar{\rho}$ is the Favre-averaged value of q with $q" = q - \tilde{q}$; subscripts u and b designate unburned and burned gas, respectively.

Within the framework of the FSC model, D_t and U_t are evaluated as follows [15, 16],

$$D_t = D_{t,\infty}\left[1 - \exp\left(-\frac{t_{fd}}{\tau_L}\right)\right], \tag{Appendix I.3}$$

$$U_t = U_{t,ISP}\left[1 - \frac{\tau_L}{t_{fd}} + \frac{\tau_L}{t_{fd}}\exp\left(-\frac{t_{fd}}{\tau_L}\right)\right]^{1/2}, \tag{Appendix I.4}$$

where $D_{t,\infty}$ is the fully developed turbulent diffusivity, which can be determined using the following equations

$$D_{t,\infty} = \frac{C_\mu}{Pr_t}\frac{\tilde{k}^2}{\tilde{\varepsilon}}, \tag{Appendix I.5}$$

$$L = C_d \frac{\tilde{k}^{3/2}}{\tilde{\varepsilon}}, \tag{Appendix I.6}$$

$$\tilde{k} = \frac{3}{2}u'^2, \tag{Appendix I.7}$$

within the framework of the standard $k - \varepsilon$ turbulence model [41]; t_{fd} is the flame development time counted starting from ignition; $\tau_L = D_{t,\infty}/u'^2$ is the Lagrangian time scale of turbulence; u' is the rms turbulent velocity;

$$U_{t,ISP} = Au'Da^{1/4} \qquad \text{(Appendix I.8)}$$

is an intermediately steady turbulent burning velocity; A=0.4 [15] is the sole constant of the FSC model; $Da = \tau_t/\tau_f$ is the Damköhler number; $\tau_t = L/u'$ and $\tau_f = \delta_L/S_L$ are turbulent and laminar-flame time scales, respectively; L is an integral turbulent length scale; S_L and $\delta_L = \kappa_u/S_L$ are the laminar flame speed and thickness, respectively. As discussed in detail elsewhere [16, 20, 42], at moderate turbulence, Eq. (Appendix I.8) is qualitatively consistent with various experimental data on the influence of mixture composition, turbulence characteristics, and pressure on turbulent burning velocity or flame speed.

Originally, Eq. (Appendix I.8) was analytically derived by Zimont [42] by assuming that (i) small-scale eddies increase local burning velocity by thickening flamelets and increasing heat and mass transfer within them, with the width of the thickened flamelets being still significantly smaller than L; (ii) large-scale eddies increase local burning velocity by wrinkling the thickened flamelets; and (iii) the mean turbulent flame brush thickness δ_t grows by the turbulent diffusion law. Such a regime characterized by apparently stationary turbulent burning velocity given by Eq. (Appendix I.8), but growing δ_t was later called Intermediate Steady Propagation (ISP) regime [43]. Numerous experimental data reviewed elsewhere [16, 44] indicate that such a combustion regime is a widespread regime of premixed turbulent burning.

The original derivation of Eq. (Appendix I.8) was performed under the following constraints: the turbulent Reynolds number $Re_t = u'L/\nu_u \gg 1$, $Da \gg 1$, the Karlovitz number $Ka = Re_t^{1/2}/Da > 1$, and $\tau_t < t_{fd} \ll \tau_t Da^{1/2}$ [42]. Here, ν_u is the kinematic viscosity of unburned gas. Subsequently, Lipatnikov and Chomiak [16] argued that the aforementioned assumption (i), i.e. thickening of flamelets by small-scale eddies, could be substituted with a more general assumption that the interaction between small-scale turbulent eddies and flamelets is controlled by the mean dissipation rate $\tilde{\varepsilon}$ and chemical time scale τ_c. Under this assumption, which is in fact an extension of the well-recognized Kolmogorov hypothesis to the case of premixed turbulent combustion, the constraint of $Ka > 1$ is substituted with $u'/S_L > 1$ and the model is applicable to moderately turbulent burning also.

If $D_t = D_{t,\infty}$, $U_t = U_{t,ISP}$, and $Q = 0$ in Eq. (Appendix I.2), the FSC model reduces to the TFC model by Zimont and Lipatnikov [45, 46]. The time-dependent terms in square brackets in Eqs. (Appendix I.3) and (Appendix I.4) extend the TFC model and allow us to simulate early stages of premixed turbulent flame development, including the formation of a small flame kernel after ignition, transition to turbulent burning, and development of the turbulent flame. Equation (Appendix I.3) is well known in the turbulence literature [47] and results from the Taylor [48] theory of turbulent diffusion. Equation (Appendix I.4) was derived by Lipatnikov and Chomiak [15] by adapting the Taylor's theory to extend the Zimont model of the intermediately steady turbulent burning velocity.

In order to (i) simulate an early stage of flame kernel growth after spark ignition and (ii) obtain an appropriate balance equation in the limit case of $u' \to 0$, the TFC model was further extended and the following source term [15, 16]

$$Q = \frac{\bar{\rho}(1-\tilde{c})}{t_r(1 + D_t/\kappa_b)} \exp\left(-\frac{\Theta}{\tilde{T}}\right) \qquad \text{(Appendix I.9)}$$

was incorporated into in Eq. (Appendix I.2). Here, Θ is the activation temperature for a single reaction that the combustion chemistry is reduced to ($\Theta = 20000$ K in the present work); the Favre-averaged temperature $\tilde{T}$ is evaluated using the simplest form $\bar{\rho}\tilde{T} = \rho_u T_u$ of the ideal gas state equation; and the reaction time scale t_r is set so that, in the case of $u' = 0$, the burning velocity yielded by stationary, 1-D Eqs. (Appendix I.2), and (Appendix I.9) is equal to the laminar burning velocity S_L, which is an input parameter of the model. This constraint results in

$$t_r = \Psi^2\left(\frac{T_b}{T_u}, \frac{\Theta}{T_u}\right)\frac{\kappa_u}{S_L^2}, \qquad \text{(Appendix I.10)}$$

where the non-dimensional function Ψ approximates values of $S_L\sqrt{t_r/\kappa_u}$, pre-computed for various ratios of T_b/T_u and Θ/T_u by numerically integrating 1-D Eqs. (Appendix I.1), (Appendix I.2), and (Appendix I.9) with $D_t = U_t = 0$. In this case, the FSC Eq. (Appendix I.2) reduces to an equation that models a laminar premixed flame in the case of a single combustion reaction, with the source term Q being introduced into Eq. (Appendix I.2) by Lipatnikov and Chomiak [15, 16] in order to satisfy this constraint. A polynomial approximation of the function Ψ does not feature any tuning parameter and is reported by Huang et al. [49].

Following the work by Zimont and Lipatnikov [45], the ignition source term W_{ign} is written as follows

$$W_{ign} = W_0\exp\left\{-\left[\left(\frac{r}{\sigma_r}\right)^2 + \left(\frac{t-t_0}{\sigma_t}\right)^2\right]\right\}(1-\tilde{c}). \qquad \text{(Appendix I.11)}$$

Here, W_0, σ_r, σ_t are parameters for ignition model. More specifically, t_0 is associated with ignition time, σ_t characterizes ignition duration, and σ_r corresponds to the size of ignition kernel. The factor W_0 is associated with the ignition strength and should be set sufficiently large in order for $\tilde{c}(r = 0, t_0)$ to be close to unity.

Appendix I.II Basic features

If $Q = 0$ in Eq. (Appendix I.2), there is the following exact analytical travelling-wave solution [50-52]

$$\bar{c} = \frac{1}{2}\mathrm{erfc}\left(\xi\sqrt{\pi}\right) = \frac{1}{\sqrt{\pi}}\int_{\xi\sqrt{\pi}}^{\infty} e^{-\zeta^2}d\zeta, \qquad \text{(Appendix I.12)}$$

$$\xi = \frac{x - x_f(t)}{\delta_t(t)}, \qquad \text{(Appendix I.13)}$$

$$x_f(t) = x_f(t=0) + \int_0^t U_t(\theta)d\theta, \qquad \text{(Appendix I.14)}$$

and

$$\delta_t^2(t) = 4\pi\int_0^t D_t(\theta)d\theta \qquad \text{(Appendix I.15)}$$

to Eqs. (Appendix I.1) - (Appendix I.4) for a statistically 1-D planar flame that propagates from left to right in frozen turbulence.

Equations (Appendix I.12) and (Appendix I.13) describe a flame with a self-similar mean structure. As reviewed elsewhere [16, 21, 44, 53] and supported by more recent experimental data [54, 55], various premixed turbulent flames do have such a self-similar mean structure well described by Eqs. (Appendix I.12) and (Appendix I.13). It is worth stressing that a transport equation, which (i) was basically similar to Eq. (Appendix I.2), (ii) had the exact solution given by Eqs. (Appendix I.12)-(Appendix I.15), but (iii) was written in a different form, was introduced into the combustion literature by Prudnikov [44] by considering statistically 1-D planar case.

Equation (Appendix I.15) predicts that the growth of $\delta_t(t)$ follows the turbulent diffusion law. Indeed, as hypothesized by Karlovitz et al. [56], reviewed elsewhere [16, 21, 44, 53] and supported by more recent data [55, 57] , the growth of the mean flame brush thickness does follow the turbulent diffusion law in various experiments.

In the statistically 1-D, but spherical case, the solution given by Eqs. (Appendix I.12) - (Appendix I.15) is not exact. Nevertheless, if the mean flame structure is assumed to be self-similar, i.e. $\bar{c}(r,t) = \bar{c}\left[\left(r - \bar{R}_f\right)/\delta_t\right]$, where r is the radial distance and $\bar{R}_f$ is a mean flame radius, the following analytical relation holds [59]

$$\frac{d\bar{R}_f}{dt} = \sigma U_t \left(\int_0^\infty \tilde{c} r dr\right)\left(\int_0^\infty \bar{c} r dr\right)^{-1} \qquad \text{(Appendix I.16)}$$

for a particular mean flame radius defined elsewhere [59], with this result agreeing quantitatively with the Leeds experimental data [60].

Finally, as already noted in Sect. 2.1 and discussed in detail elsewhere [16, 20, 21], the FSC model was quantitatively validated in RANS simulations of a wide set of experiments performed by various research groups with various (both expanding and statistically stationary) flames under a wide range of substantially different conditions (various fuels, equivalence ratios, initial temperatures, pressures, rms turbulent velocities, and turbulent length scales).

Appendix I.III The extended FSC model

The FSC model addresses the influence of turbulence on combustion but does not allow for the influence of combustion-induced thermal expansion on turbulence. The latter influence should be addressed by a turbulence model. However, in spite of long-term research into such thermal expansion effects and a number of important phenomena found in experimental and direct numerical simulation studies reviewed elsewhere [61, 62] a model with well-documented capabilities for predicting effects of thermal expansion on turbulence in premixed flames has not yet been developed. Nevertheless, such effects should be considered in a CFD study of a gaseous or dust explosion, because they cause significant self-acceleration of a growing flame kernel [63-65]. In such a challenging situation, a simple semi-empirical approach is chosen in the present work as a solution for applied CFD research into large-scale explosions.

The approach is based on a seminal study by Gostintsev et al. [66] who analysed a large amount of experimental data obtained in large-scale experiments with growing flame kernels. Their analysis revealed a self-similar regime of flame kernel growth, characterized by the following empirical relation for the flame kernel radius

$$R_f(t) = R_{f,0}\left(\frac{t}{t_0}\right)^{3/2} \qquad \text{(Appendix I.17)}$$

While the “initial” (for this regime, but not for the flame kernel ignition) values of the flame radius $R_{f,0}$ and time t_0 depend on mixture composition and other experimental conditions, the same power exponent 3/2 fits well to all data analysed by Gostintsev et al. [66]. Subsequently, the existence of such a regime was supported in an experimental study by Bradley et al. [67]. This regime was also addressed in other phenomenological and theoretical studies [68, 69] but discussion of such studies is beyond the scope of the present work.

Here, based on the experimental findings briefly reviewed above, the FSC Eq. (Appendix I.4) is simply modified as follows

$$U_t = U_{t,ISP} \underbrace{\left[1 - \frac{\tau_L}{t_{fd}} + \frac{\tau_L}{t_{fd}} \exp\left(-\frac{t_{fd}}{\tau_L}\right)\right]^{1/2}}_{T_1} \underbrace{\left(\frac{t}{t_{flacc}}\right)^{1/2}}_{T_2}, \qquad \text{(Appendix I.18)}$$

where t_{flacc} is the timing for activating the flame acceleration mechanism in simulations (if $t < t_{flacc}$, the original FSC Eq. (Appendix I.4) is used). Due to the lack of a model or empirical formula, which could be adopted to calculate the values of $R_{f,0}$ and t_0, associated with the onset of the discussed self-similar regime, Eq. (Appendix I.18) requires tuning t_{flacc}. In this work, t_{flacc} was chosen by comparing the simulated explosion overpressure with the experimental one.

While Eq. (Appendix I.18) involves two unsteady terms T_1 and T_2, these terms are associated with different physical mechanisms and control an increase in $U_t(t)$ during different time intervals. More specifically, term T_1 was theoretically obtained by Lipatnikov and Chomiak [15] by combining a model of intermediately steady turbulent flame propagation [42] with the classical theory of turbulent diffusion [47, 48, 70]. As turbulence is mainly rotational motion, term T_1 models an increase in turbulent burning rate due to the influence of the rotational motion on a premixed flame. Moreover, this term rapidly grows from zero with time and reaches unity at $t_{fd} \gg \tau_L$. In particular, this term varies weakly and is close to unity at $t > t_{flacc}$, but plays an important role during an earlier stage of flame kernel growth. Note that during this earlier stage, the kernel self-accelerates also due to a decrease in a ratio of the mean flame brush thickness to the mean kernel radius with time, as discussed in detail by Lipatnikov and Chomiak [58, 71]. Contrary to term T_1, term T_2 plays a role only at $t > t_{flacc}$, grows from unity at $t > t_{flacc}$ to 1.4 at $t = 0.3$ ms, when the overpressure peaks, and models the kernel acceleration under the influence of potential velocity fluctuations caused by rapid propagation of combustion-induced pressure perturbations into unburned mixture. Thus, terms T_1 and T_2 are associated with different types of velocity fluctuations (rotational and potential, respectively) and substantially affect the kernel growth rate during different time intervals.

Appendix I.IV Transport equations for combustion progress and regress variables

Since Open FOAM deal with a transport equation with combustion regress variable $\tilde{b}$, the FSC model equations should be adapted accordingly. This appendix shows how to derivate the transport equation for $\tilde{b}$ from the combustion progress variable $\tilde{c}$ equation. By substituting $\tilde{c} = 1 - \tilde{b}$ into Eq. (Appendix I.2), we get

2], the planar 1-D Equation (Appendix I.2) without the last source term and with $\kappa = 0$, i.e., the truncated FSC model, has the following benchmark analytical solution

$$\bar{c} = \frac{1}{2}\mathrm{erfc}\left(\xi\sqrt{\pi}\right) = \sqrt{\frac{1}{\pi}} \int_{\zeta\sqrt{\pi}}^{\infty} e^{-\zeta^2} d\zeta \qquad \text{(Appendix I.19)}$$

provided that Equation (Appendix I.1) holds and neither D_t nor U_t varies in the space. The complementary error function $\mathrm{erfc}\left(\xi\sqrt{\pi}\right)$ can be calculated using a python script reported in Appendix I.VI. Here,

$$\xi = \frac{x - x_f}{\Delta_t} \qquad \text{(Appendix I.20)}$$

is the normalized distance, x_f is the mean flame position, associated with $\bar{c}$=0.5, mean flame brush thickness Δ_t is defined as follows

$$\Delta_t = \frac{1}{\left|\frac{d\bar{c}}{dx}\right|_{\max}} \qquad \text{(Appendix I.21)}$$

and grows following turbulent diffusion law

$$\Delta_t = 2u'\left\{\pi\tau_L t_{fd}\left[1-\frac{\tau_L}{t_{fd}}+\frac{\tau_L}{t_{fd}}\exp\left(-\frac{t_{fd}}{\tau_L}\right)\right]\right\}^{1/2}. \qquad \text{(Appendix I.22)}$$

This solution describes a developing turbulent wave with self-similar mean structure, i.e. dependence of $\bar{c}$ on two independent variables x and t reduces to $\bar{c}(\xi)$. It is also worth noting that, while the transport Equation (Appendix I.2) describes evolution of the Favre-averaged combustion progress variable, the solution is written for the Reynolds-averaged progress variable using the BML identity $\bar{\rho}\tilde{c} = \rho_b\bar{c}$ [72].

In the considered 1-D planar case, turbulent flame speed is equal to turbulent burning velocity given by Equations (Appendix I.4) and (Appendix I.5). In 1-D spherical case, turbulent flame speed is reduced due to mean curvature of the flame brush. This reduction effect is approximately evaluated as follows

$$\frac{1}{\sigma}\frac{dR_f}{dt} = U_t\int_0^\infty \tilde{c}rdr\left\{\int_0^\infty \bar{c}rdr\right\}^{-1}, \qquad \text{(Appendix I.23)}$$

see Equation (7.147) on page 364 in Ref. [20].

Appendix I.VI Python script of calculating complementary error function

```
import numpy as np   # import numpy library

zeta=np.arange(-2,2.1,0.1)  #zeta is ξ in Equation (Appendix I.30)
xi=zeta/pow(math.pi,0.5)    #xi is ζ in Equation (Appendix I.29)
zn=np.where(zeta>=0,1.,-1.)   # if xi >=0, yield 1, otherwise yeild -1
zz=1./(1.+0.47047*zeta*zn)
f=0.5*(1.+zn*(1.-(0.3480242*zz-0.0958709*zz*zz+0.7478556*zz*zz*zz)*np.exp(-zeta*zeta)))
cBar_ana=1.-f   #cBar_ana is the analytical Reynolds-averaged progress variable
```

Appendix II. Verification of model implementation

In order to verify the implementation of the FSC model into OpenFOAM, we simulated simplified cases where benchmark analytical solution could be obtained, e.g., 1-D planar laminar premixed flame, 1-D planar flame propagating in “frozen” turbulence, and 3-D spherical flame propagating in “frozen” turbulence. Subsequently, results computed using the implemented model were compared with the analytical solutions. It is worth remembering that, at this verification stage, simulations of turbulent flames were performed using the truncated FSC model, i.e., the two terms Q and $\bar{\rho}W_{ign}$ were omitted in Equation (Appendix I.2), because the analytical solutions were obtained for that truncated transport equation.

The implementation of the FSC model was verified using three options, i.e. (i) comparison of the normalized profile of the Reynolds-averaged progress variable $\bar{c}$ with the analytical solution given by Equation (Appendix I.29), (ii) comparison of the computed growth of the mean flame brush thickness with the turbulent diffusion law given by Equation (Appendix I.32), and (iii) comparison of the computed flame speed with the turbulent flame speed yielded by the FSC model expressions given by Equation (Appendix I.4) in the planar case or Equation (Appendix I.33) in the spherical case.

The premixed burning of cornflour dust cloud was simulated. The cornflour chemical equivalent formula is $C_6H_{7.88}O_{4.98}$ with a heat of reaction being -15.8 MJ/kg [18]. Note that a negative value of heat of reaction indicates an exothermic reaction (heat is produced). The chemical reaction of cornflour is as follows

$$C_6H_{7.88}O_{4.98} + 5.48\ (O_2 + 0.79/0.21\ N_2) \Rightarrow 6\ CO_2 + 3.94\ H_2O + 20.615\ N_2 \qquad \text{(Appendix II.1)}$$

Appendix II.I Truncated FSC model: 1-D planar flame in “frozen” turbulence

The 1-D model has a domain size of 0.1 m, and a cross section size of 0.003 × 0.003 m, Figure Appendix II.1. There are 100 cells along the x direction, and three cells in the y and z directions, respectively. Burned products occupy the left-hand side (LHS), whereas unburned reactants occupy the right-hand side (RHS). Zero velocity and free entrainment boundary conditions are set on the right (unburned) and the left (burned) boundaries, respectively. In the latter case, the pressure at the boundary for compressible subsonic flow is calculated using the following equation

$$p_p = p_0 - \frac{1}{2}\rho|\tilde{\mathbf{u}}|^2 \qquad \text{(Appendix II.2)}$$

where p_p is the pressure at the boundary patch, and p_0 is the total pressure.

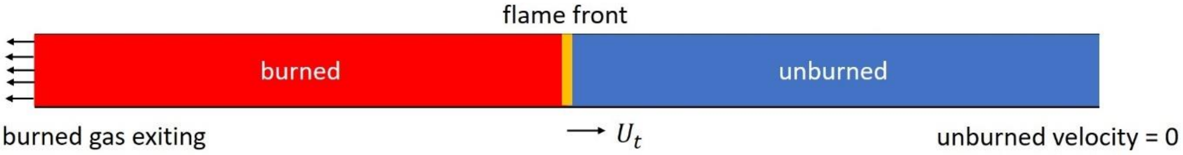


Figure Appendix II.1 Layout of 1-D planar flame.

The premixed turbulent flame propagates from burned to unburned side. The thermo-physical properties of unburned and burned mixture, the initial and boundary conditions are summarized in Tables AppendixII.1 - AppendixII.4, respectively. The case setup is shown in Appendix VI.

Table AppendixII.1 Thermo-physical properties of unburned and burned mixtures.

parameters		value
unburned	T_u [K]	328
	W_u [g/mol]	32.76
	ρ_u [kg/m^3]	1.32138
	μ_u [kg/(m·s)]	1.8e-5
burned	T_b [K]	1599
	W_b [g/mol]	27.15
	ρ_b [kg/m^3]	0.2255
	μ_b [kg/(m·s)]	4.6e-5
others	$\sigma = \rho_u/\rho_b = (T_b W_u)/(T_u W_b)$ [-]	5.86
	Pr_t	0.7
	S_L [m/s]	0.12

Table AppendixII.2 Initial conditions for 1-D planar flame.

parameters	value
T_0 [K]	328
P_0 [Pa]	11 000
$\tilde{k}$ [m^2/s^2]	0.96
u' [m/s]	0.8

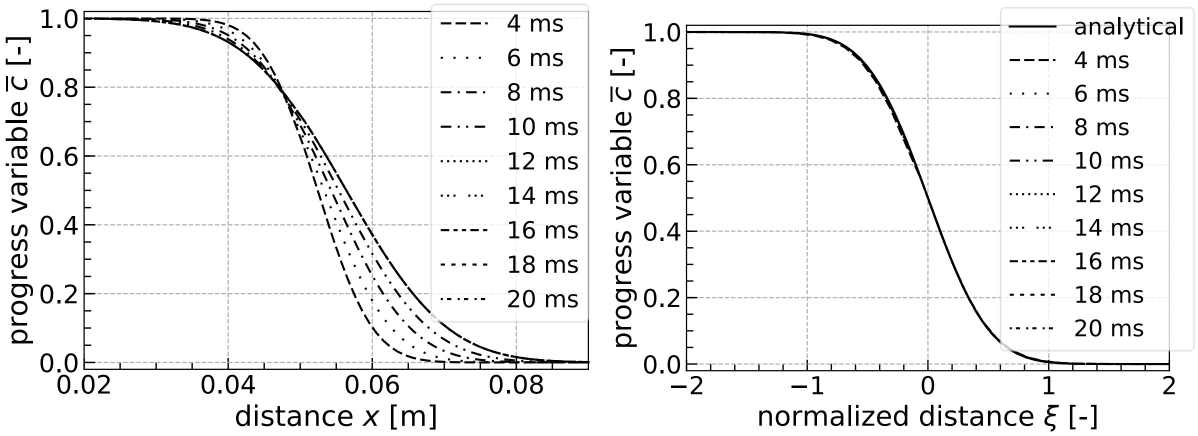
Table AppendixII.3 Different initial turbulence dissipation rates and the corresponding turbulence length scales.

cases	1	2	3
$\tilde{\varepsilon}$ [m^2/s^3]	11.84	69.6	348
L [m]	0.029	0.005	0.001

Table AppendixII.4 Boundary conditions in the x direction.

parameters	burned (left)	unburned (right)
P [Pa]	totalPressure	fixedValue 110 000
$\tilde{\mathbf{u}}$ [m/s]	pressureInletOutletVelocity	fixedValue (0 0 0)
$\tilde{T}$ [K]	zeroGradient	fixedValue 328
$\tilde{b}$ [-]	zeroGradient	fixedvalue 1

The calculated profiles of the Reynolds-averaged combustion progress variable $\bar{c}$ versus distance change with time; see Figure Appendix II.2 (a). However, the profile of $\bar{c}$ versus normalized distance ξ is the same for all the time instants, in line with the well-documented self-similarity of premixed flames [52]. It can be seen in Figure Appendix II.2 (b) that the profiles of $\bar{c}$ versus ξ for different time instants and the analytical solution (see Equation (13) in Ref. [52] or Eq. (Appendix I.29)) agree very well.



(a), Reynolds-averaged combustion progress variable $\bar{c}$ vs distance (b), $\bar{c}$ vs normalized distance

Figure Appendix II.2 Spatial profiles of the Reynolds-averaged combustion progress variable for 1-D planar flame propagating in "frozen" turbulence.

Comparison between calculated and analytical flame speeds is shown in Figure Appendix II.3. In the simulations, the flame speed was evaluated by taking derivative of mean flame position against time. The mean flame position is defined by the *x*-coordinate of a surface $\bar{c} = 0.5$. Since the calculated flame position exhibited fluctuations, UnivariateSpline function in scipy library of Python was used to smooth the data. Figure Appendix II.3 shows that the numerical and analytical results agree well. There is a slight discrepancy in the beginning and in the end of the curve. This may be caused by the smoothing function and the uncertainty caused by the numerical schemes, time-step and grid size.

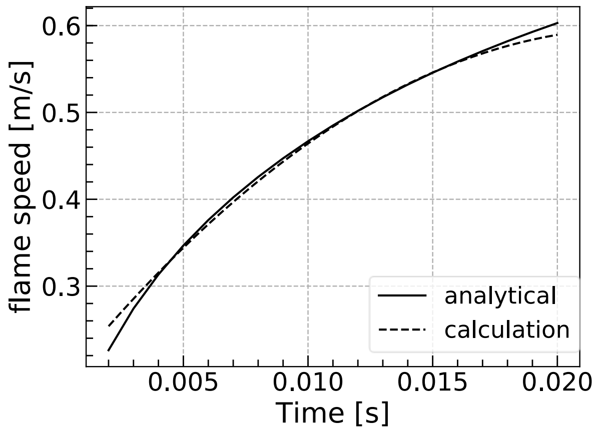


Figure Appendix II.3 Comparison of calculated flame speed with flame speed yielded by FSC model for 1-D planar flame propagating in "frozen" turbulence.

Comparison between mean flame brush thickness evaluated by processing the computed profiles of $\bar{c}(x,t)$ using Equation (Appendix I.31) and the analytical solution given by Equation (Appendix I.32) is shown in Figure Appendix II.4. Overlapping of solid and dashed lines verifies the implementation of the FSC model.

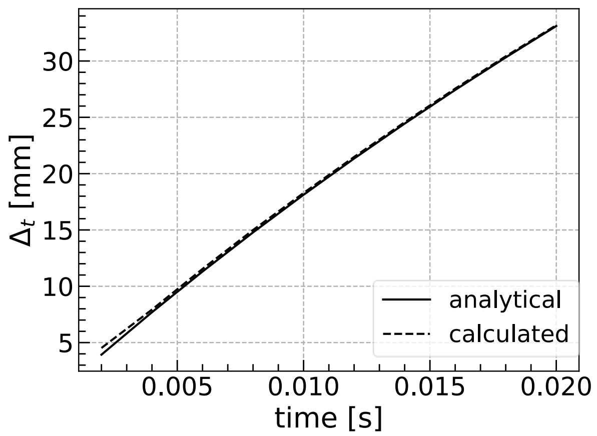
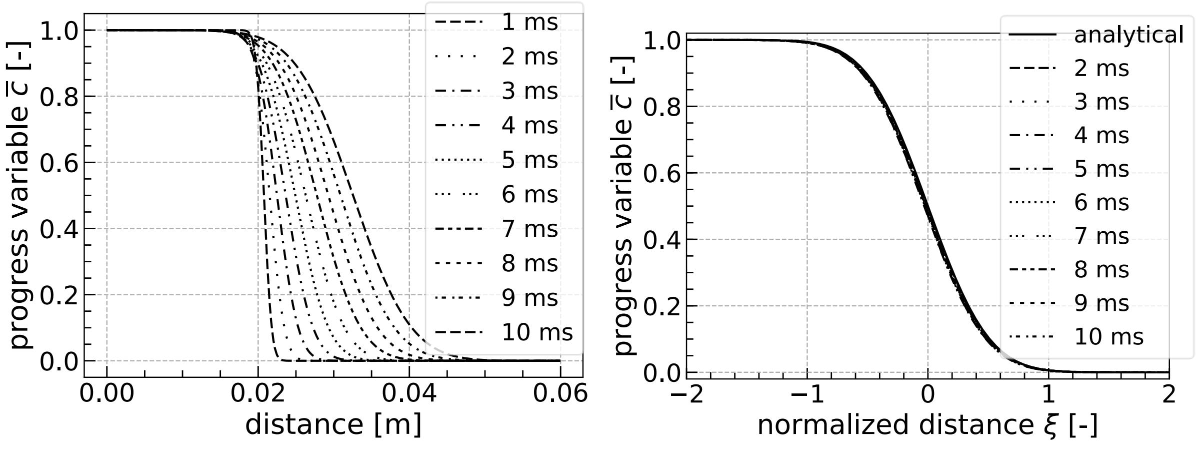


Figure Appendix II.4 Comparison of calculated mean flame brush thickness Δ_t with turbulent diffusion law given by Eq. (Appendix I.32) for 1-D planar flame propagating in "frozen" turbulence.

Appendix II.II Truncated FSC model: 3-D spherical flame in "frozen" turbulence

A cube geometry was created in order to represent one eighth of the total computational domain. The computational domain had an edge of 60 mm and a mesh size of 0.25 mm. Totally 13 824 000 cells were created. The model took around 42 wall clock hours running on Simlab computer using 14 cores. The initial conditions corresponded to a spherical kernel of a radius of 20 mm, filled with combustion products. The rest of the domain was filled with unburned mixture. The detailed case setup is shown in Appendix VII. The initial conditions, boundary conditions, and thermo-physical properties were the same as for 1-D planar case; see Tables AppendixII.1 - AppendixII.4. To perform comparison with approximate analytical solution, results were obtained using the FSC model without extra source term Q (see Equation (Appendix I.2) without Q term).

The Reynolds-averaged progress variable $\bar{c}$ was evaluated by taking the average value of $\bar{c}$ along x, y and z directions. The computed dependencies of $\bar{c}$ on the normalized distance ξ are overlapping and agree well with the complementary error function; see Figure Appendix II.5 (b).



(a), Reynolds-averaged combustion progress variable $\bar{c}$ vs distance (b), $\bar{c}$ vs normalized distance

Figure Appendix II.5 Spatial profiles of the Reynolds-averaged combustion progress variable for 3-D spherical flame propagating in "frozen" turbulence.

For an expanding spherical flame, the burning velocity is lower than that of a planar due to the influence of the mean curvature of the flame brush. The FSC model permits an analytical estimate of the magnitude of such a reduction effect. As shown elsewhere [20], the effect magnitude is equal to $\int_0^\infty \tilde{c} r dr \left\{\int_0^\infty \bar{c} r dr\right\}^{-1}$; see Equation (Appendix I.33). To verify the implementation, a ratio of the computed turbulent flame speed S_t (with respect to unburned mixture) and the theoretical turbulent burning velocity S_t given by Equation (Appendix I.4) was compared with the integral ratio calculated using the analytical solution given by Equation (Appendix I.33). To do so, (i) S_t was calculated by differentiating filtered curve plotted in Figure Appendix II.6 and (ii) since the complementary error function in this solution involved a normalized distance, the integrals of $\int_0^\infty \tilde{c} r dr$ and $\int_0^\infty \bar{c} r dr$ were also evaluated using the normalized distance, e.g.,

$$\int_0^\infty \tilde{c} r dr = \Delta_t \int_{-\frac{r_f}{\Delta_t}}^\infty \tilde{c}\left(\Delta_t \xi + r_f\right) d\xi \qquad \text{(Appendix II.3)}$$

where $\xi = \frac{r-r_f}{\Delta_t}$, and $dr = \Delta_t d\xi$.

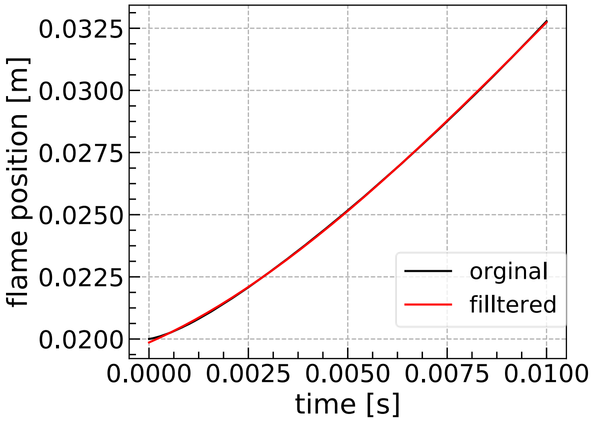


Figure Appendix II.6 Calculated flame position (original and filtered) versus time for 3-D spherical flame propagating in "frozen" turbulence.

As shown in Figure Appendix II.7, the two ratios, i.e. $\int_0^\infty \tilde{c} r dr \left\{\int_0^\infty \bar{c} r dr\right\}^{-1}$ and S_t/U_t, are sufficiently close to one another, thus, further verifying the model implementation. The quantities are not exactly equal, because the analytical estimate of the reduction effect magnitude is an approximate one for the 3-D spherical flame.

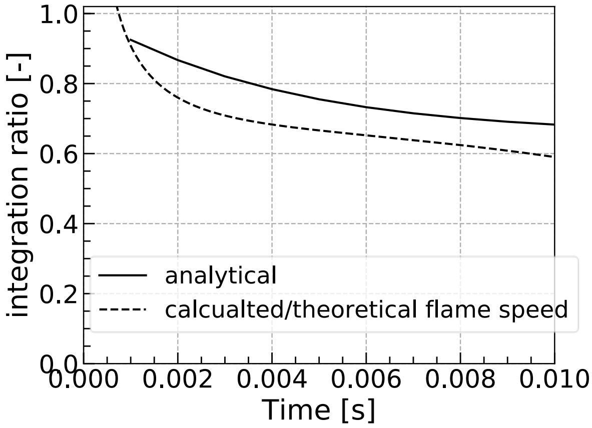


Figure Appendix II.7 Comparison of analytical integral ratio with the ratio of calculated flame speed with respect to unburned mixture divided by theoretical turbulent flame speed by Equation Appendix I.4) for 3-D spherical flame propagating in "frozen" turbulence.

Figure Appendix II.8 shows that analytical and numerical results for the mean flame brush thickness are close to each other. It is worth stressing again that, contrary to the 1-D planar case discussed earlier, the analytical equation is not exact and, consequently, some mild differences between this equation and numerical data are expected.

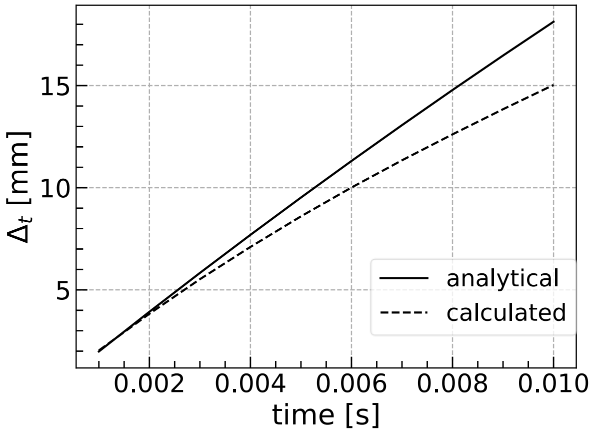


Figure Appendix II.8 Comparison of calculated mean flame brush thickness with Eq. (Appendix I.32) for 3-D spherical flame propagating in "frozen" turbulence.

Appendix II.III Truncated FSC model: Influence of ignition model

Results reported in the previous section were obtained by omitting Q term in Equation (Appendix I.2) (otherwise an analytical benchmark solution is difficult to obtain) and using $W_0 = 0$ in term $\bar{\rho}W_{ign}$, with ignition being simulated by creating a product kernel at the initial instant. Alternatively, ignition could be simulated using uniform initial conditions of $\tilde{b}(\mathbf{x}, t = 0) = 1$ and a large value of W_0. Results of such simulations are discussed in the present subsection.

Black dashed line in Figure Appendix II.9 shows that the use of the latter ignition model yields too small mean flame brush thickness for the turbulent length scale of 5 mm. Examination of the numerical data indicated that the computed thickness was too small, because the gradient $|\nabla\bar{c}|$ was too much in the vicinity of the kernel centre. This problem was solved by using the complete version of the FSC model, i.e., by retaining Q term in Equation (Appendix I.2). Relevant results will be reported in sections Appendices II.IV and II.V.

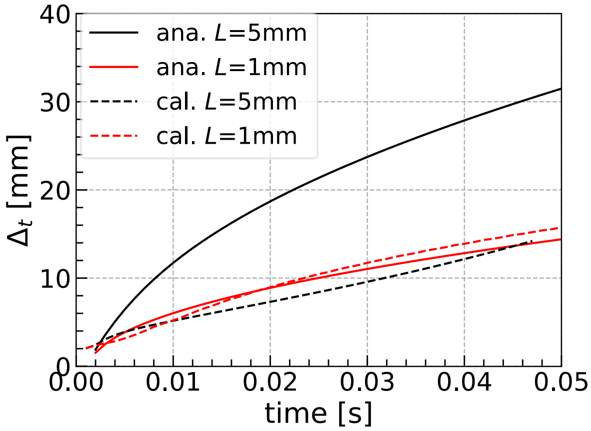


Figure Appendix II.9 Comparison of calculated mean flame brush thickness against turbulent diffusion law in Eq. (Appendix I.32) for turbulent length scale 1, and 5 mm for 3-D "frozen" turbulence spherical flame.

Appendix II.IV Complete FSC model: 1-D laminar planar flame

To verify implementation of the complete FSC model, we still have to verify implementation of Q term in Equation (Appendix I.2). To do so, 1-D planar laminar flame was simulated, because this term is of the most importance under such conditions. Indeed, when turbulent Reynolds number Re_t is increased so that $Re_t \gg 1$, the magnitude of this term is significantly reduced due to a ratio of $D_t/\kappa_b \propto Re_t$ in the denominator. Another goal of such simulations was to find out a value of the reaction time scale t_r that yielded the required value of the laminar flame speed $S_L = 0.12$ m/s.

The cornflour laminar flame thickness δ_L is evaluated as follows

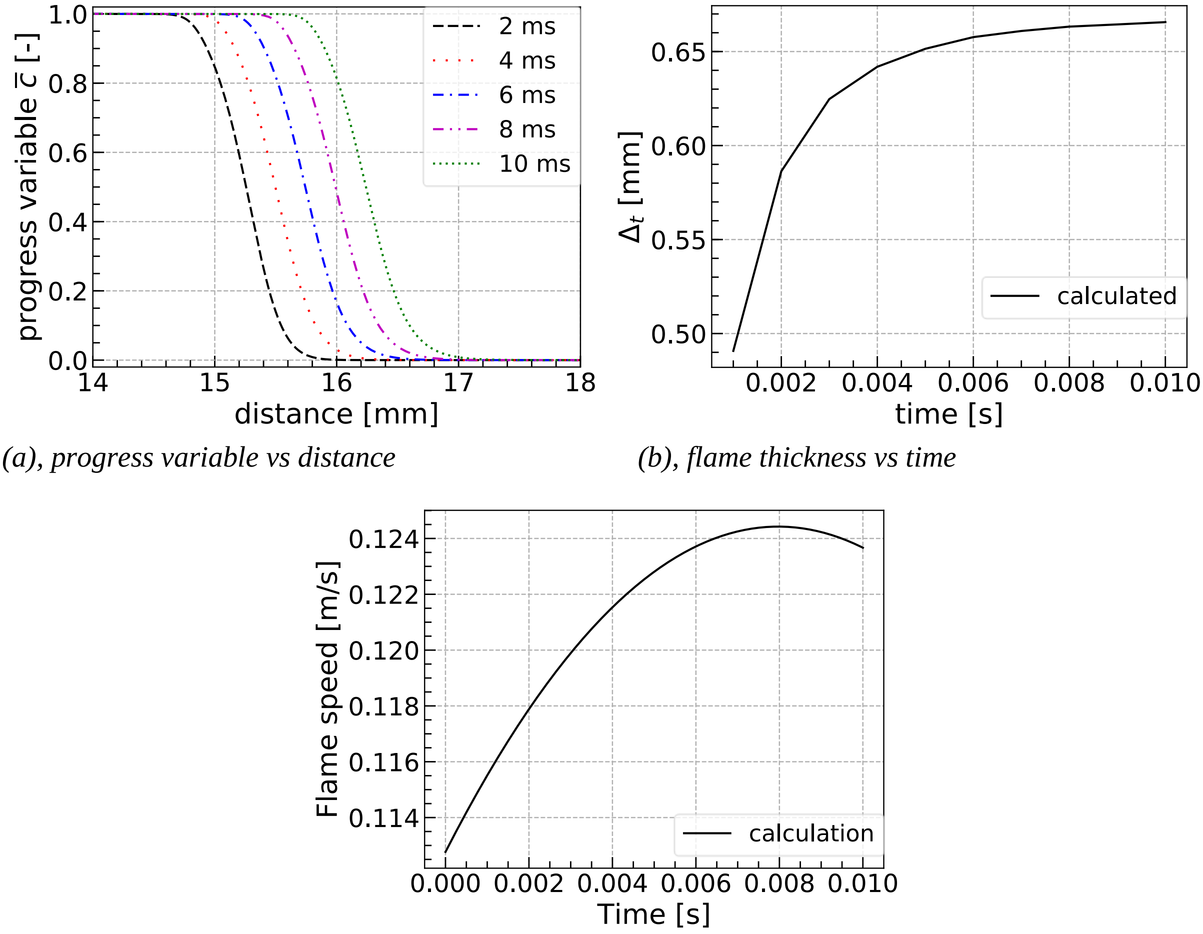
$$\delta_L = \frac{\kappa_u}{S_L} = \frac{\mu_u}{Pr\rho_u S_L}. \qquad \text{(Appendix II.4)}$$

By using the thermo-physical properties of the unburned and burned mixture in Table AppendixII.1, the cornflour laminar flame thickness is estimated to be around 1.6e-4 m. In order to resolve the small thickness of the laminar flame, a tube with a mesh size of 2.5e-5 m and a domain of 3e-2×7.5e-5×7.5e-5 m was constructed. For the first and the last 1e-2 m, a grading mesh with mesh size between 2.5e-5 and 2.5e-4 m was used. For the middle part of the domain where laminar flame propagates, a uniform mesh with size of 2.5e-5m was used. There were 600 grid cells in the *x* direction and 3 grid cells in the y and z directions, respectively, yielding 5400 cells. One numerical run took around 2 h on Simlab computer on one core for simulating flame propagation during 1e-2 s with a time step of 1e-7 s.

The layout of 1-D planar flame is shown in Figure Appendix II.1, in which the left-hand side (0-1.5e-2 m) is filled with burned mixture and the right-hand side (1.5e-2 – 3e-2 m) is filled with unburned mixture. The boundary conditions for the unburned and burned sides are zero velocity and free entrainment, respectively. The initial and boundary conditions for the simulations are shown in Tables AppendixII.1 - AppendixII.4, with the initial temperature and pressure being equal to 328 K and 110 000 Pa, respectively.

The FSC model constant A and turbulent heat diffusivity D_t were set equal to zero, whereas the reaction time scale was varied to obtain the required laminar flame speed. According to the classical theory by Zeldovich and Frank-Kamenetskii [73], $S_L \propto t_r{}^{-0.5}$. The same scaling was obtained in the simulation, thus, verifying the model implementation. Under conditions of the present simulations, the reaction time scale that yields the laminar flame speed of 0.12 m/s is equal 3.4e-11 s.

The computed spatial profiles of the progress variable, flame thickness and flame speed are reported in Figure Appendix II.10. As expected, the flame thickness and speed quickly reach steady value. The computed fully developed flame thickness is about 0.66 mm, i.e., significantly larger than $\delta_L = 0.16$ mm yielded by Equation (Appendix II.4). The point is that the numerical result was obtained by evaluating the maximum gradient of the progress variable and such a method is well known to yield significantly larger value of a laminar flame thickness when compared to Equation (Appendix II.4). Typically, a ratio of the two thicknesses is comparable with the density ratio.



(a), progress variable vs distance

(b), flame thickness vs time

(c), flame speed vs time

Figure Appendix II.10 Computed spatial profiles of the progress variable and evolution of thickness and speed of 1-D planar laminar premixed flame. t_r=3.4e-11 s.

Appendix II.V Complete FSC model: 3-D spherical flame in "frozen" turbulence

Reported in this section are results of application of the complete FSC model supplemented with the ignition source term to unsteady 3-D simulations of 3-D spherically symmetrical premixed flames propagating in "frozen" turbulence. The computational domain was a box with a size of 140 mm. Grading mesh was used in order to reduce the number of grid sizes. The grid size in the center of the domain was 0.25 mm and the mesh size grew with the distance from the center, with the largest mesh size being 14.6 mm near the boundary. The use of the grading mesh allowed us to simulate the problem on a mesh of 512 000 grid points. The *blockMeshDict* file for generating mesh is shown in Appendix VII. In the experiments, discussed in Appendix III, the measured turbulent length scale was 20 mm [61], the measured rms turbulent velocity u' =0.8 m/s. Accordingly, the dissipation rate ε =17.4 m^2/s^3 for C_d =0.37. The rest of the initial and boundary conditions were the same as in the 1-D planar case discussed earlier. The ignition model parameters are: W_0=1e6, σ_r=1.5e-3 m, t_0=1e-3 s, $\sigma_t = t_0/5$.

Figure Appendix II.11 shows that the complete FSC model yields a smaller mean flame brush thickness when compared to the analytical solution to the truncated FSC model in 1-D statistically planar case. This effect is attributed to Q term in Eq. (Appendix I.19), which limits the growth of mean flame brush thickness, as discussed in detail elsewhere [16]. The spatial profiles of the Reynolds-averaged combustion progress variable, computed at different instants, are shown in Figure Appendix II.12. It indicates a complete combustion in the centre of the flame, where the Reynolds-averaged combustion progress variable is almost equal to unity.

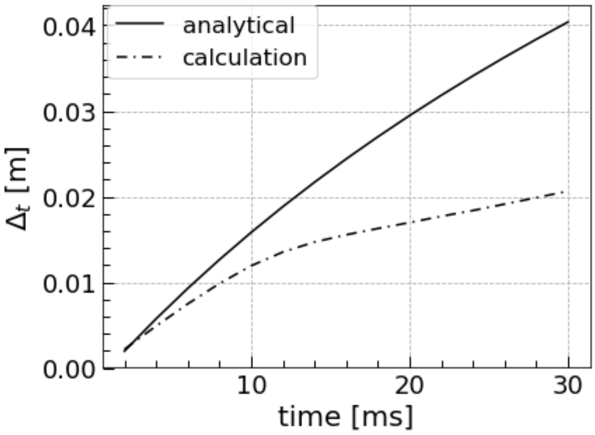


Figure Appendix II.11 Comparison of calculated mean flame brush thickness with the turbulent diffusion law given by Eq. (Appendix I.32).

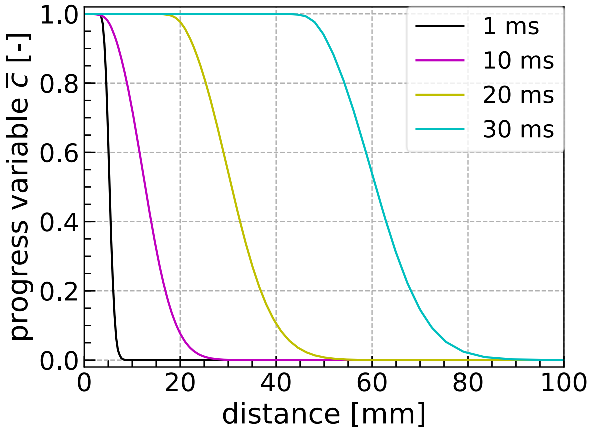


Figure Appendix II.12 Spatial profiles of the Reynolds-averaged combustion progress variable at different instants obtained from 3-D statistically spherical premixed flame propagating in "frozen" turbulence.

The calculated integral ratio and flame speed are similar to those obtained in the analytical solutions; see Figures Appendix II.13 and Appendix II.14, respectively.

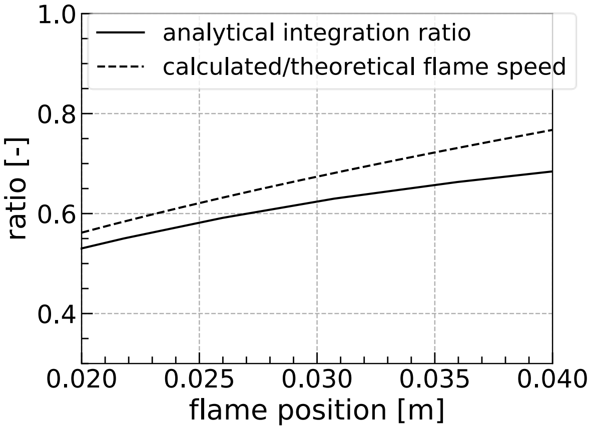


Figure Appendix II.13 Comparison of analytical integral ratio and the ratio of calculated flame speed with respect to unburned mixture divided by theoretical turbulent flame speed by Equation (Appendix I.4) for 3-D statistically spherical premixed flame propagating in "frozen" turbulence.

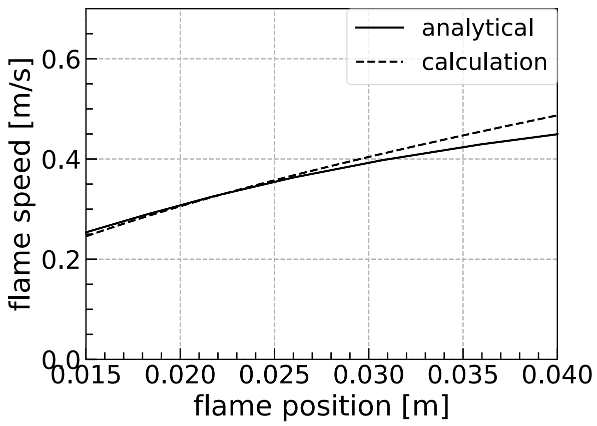


Figure Appendix II.14 Comparison of analytical and calcualted flame Speed of 3-D statistically spherical premixed flame propagating in "frozen" turbulence.

Figure Appendix II.15 shows that the size of ignition kernel weakly affects computed flame speed at a later stage of turbulent flame development. Accordingly, the use of experimental data obtained from sufficiently large flame kernels offers an opportunity to test the FSC model independently of the ignition model.

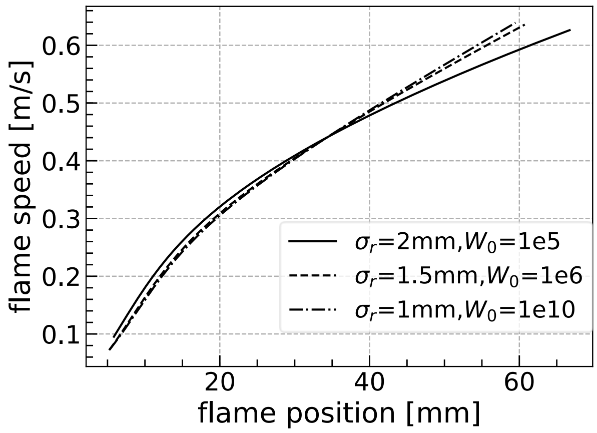


Figure Appendix II.15 Influence of the size of ignition kernel on the computed speed of 3-D statistically spherical premixed flame propagating in "frozen" turbulence.

Appendix II.VI Summary

Numerical tests discussed in Appendix II, as well as many other numerical tests skipped for brevity, show that the developed numerical platform well predicts the mean flame structure, the mean flame brush thickness, and the mean flame speed in three benchmark cases: 1-D planar laminar premixed flame, 1-D statistically planar premixed flame that propagates in "frozen" turbulence, and 3-D statistically spherical premixed flame that propagates in "frozen" turbulence. Thus, these tests verify numerical implementation of the FSC model into OpenFOAM performed within the framework of the project.

Appendix III. Model and code validation

Experimental data on cornflour dust explosion in the well-known Leeds fan-stirred combustion vessel [18] were chosen to begin assessing the model. The experimental setup is illustrated in Figure Appendix III.1. The vessel diameter is equal to 305 mm and the vessel volume is equal to 0.023 m^3. The vessel has three pairs of orthogonal quartz windows of 150 mm diameter.

Turbulence is generated by four fans, whose rotation speed was changed to vary the rms turbulent velocity u'. In the discussed experiments, the fan speed was varied from 8 to 50 Hz, which corresponded to variations in u' from 0.80 to 5.00 m/s. While the integral length scale was not reported in Ref. [18], it was reported in other papers by the Leeds group. In particular, Bradley et al. [60] have stated that the longitudinal integral length scale measured using laser Doppler velocimetry was found "*to be 20 mm and independent of fan speed between 1 000 and 10 000 rpm*", with 1 000 rpm corresponding to 16.5 Hz. It is worth noting, however, that, in the experiments with the lean dust-air mixture, the lowest fan speed was less than 16.5 Hz and a decrease in L at low fan speeds was reported in an earlier paper by the Leeds group [74]. However, those data cannot be used here, because they were obtained using thermo-anemometry, but such a method performs poorly in flows with zero mean velocity. For instance, the earlier Leeds measurements with thermo-anemometry overestimated L at large fan speeds by a factor of about two. Thus, in the Leeds experiments with the dust-air mixture, the turbulence length scale of 20 mm could be overestimated at low fan speeds. Nevertheless, when compared to other experimental data on dust explosions, the Leeds measurements were performed under well-defined laboratory conditions, i.e., the initial and boundary conditions were well controlled.

To study dust explosion, a premixed dust-air cloud was ignited by a spark in turbulent medium in the centre of the vessel. Subsequently, turbulent flame kernel growth was recorded using high-speed Schlieren system. By processing Schlieren images, an equivalent mean flame radius $\bar{R}_f$, i.e. the radius of a circle whose area was equal to the area enveloped by the flame surface on the image, was calculated and turbulent flame speed with respect to combustion products was evaluated by differentiating the measured $\bar{R}_f(t)$-curves, i.e.

$$S_{t,b} = \frac{d\bar{R}_f}{dt}. \qquad \text{(Appendix III.1)}$$

To avoid an influence of the spark on the speed, the measurements were performed in a range of 20 mm$\leq \bar{R}_f(t) \leq$35 mm. For such flame kernels, whose radius was less than the vessel radius by a factor of about 5, an increase in the pressure in the vessel was negligible.

In addition to values of $S_{t,b}$ obtained at four different $\bar{R}_f$ and five different fan speeds, reported in Ref. [18] are the values of the laminar flame speed S_L and density ratio σ for the studied dust-air mixture. However, methods and precision of evaluation of these values are not discussed. Furthermore, the value of the laminar flame thickness δ_L, which is required to calculate an important input parameter of the FSC mode such as the chemical time scale $\tau_c = \delta_L/S_L$, is not reported. Thus, even in the considered case of the small-scale well-controlled Leeds experiments, some information important for the model validation is missing. This is a typical problem for testing any model of dust explosion.

To save computational time, one eighth of a cube was simulated, with the cube volume being equal to the volume of the vessel. A computational mesh was created with an edge size of 0.14 m. The same was also used in the 3-D simulations of flame expansion in "frozen" turbulence, discussed earlier. The mesh is shown in Figure Appendix III.2. One simulation took around 6 h on simlab computer for a simulation duration of 10 ms using 2e-6 s timestep.

The initial conditions are reported in Tables Appendix III.1 and Appendix III.2. It is worth noting that the measured burned temperature of 1500 K [18] is used here instead of the calculated burned temperature of 1592 K, because neither the method, nor precision of the calculation is discussed in

Ref. [18]. The use of the former temperature yields the density ratio σ of 5.06, whereas $\sigma = 5.49$ reported in Ref. [18] corresponds to the latter (higher) temperature.

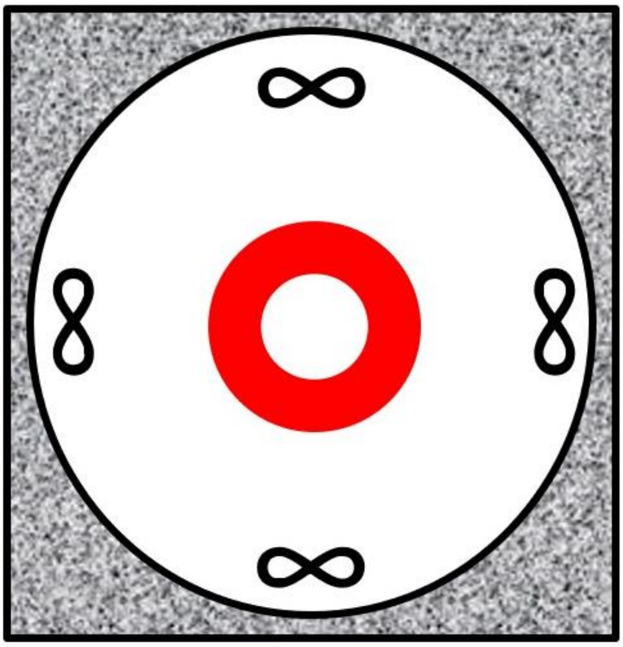


Figure Appendix III.1 Illustration of Leeds fan stirred vessel.

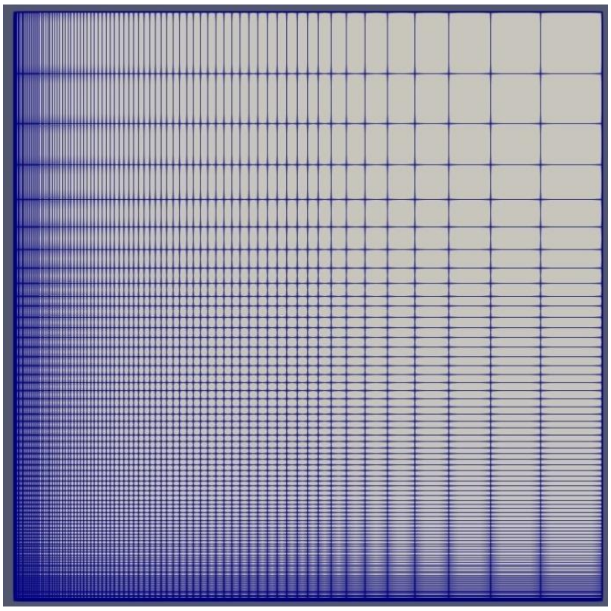


Figure Appendix III.2 Illustration of computational mesh.

Table Appendix III.1 Thermo-physical properties of unburned and burned mixture of equivalence ratio 0.77.

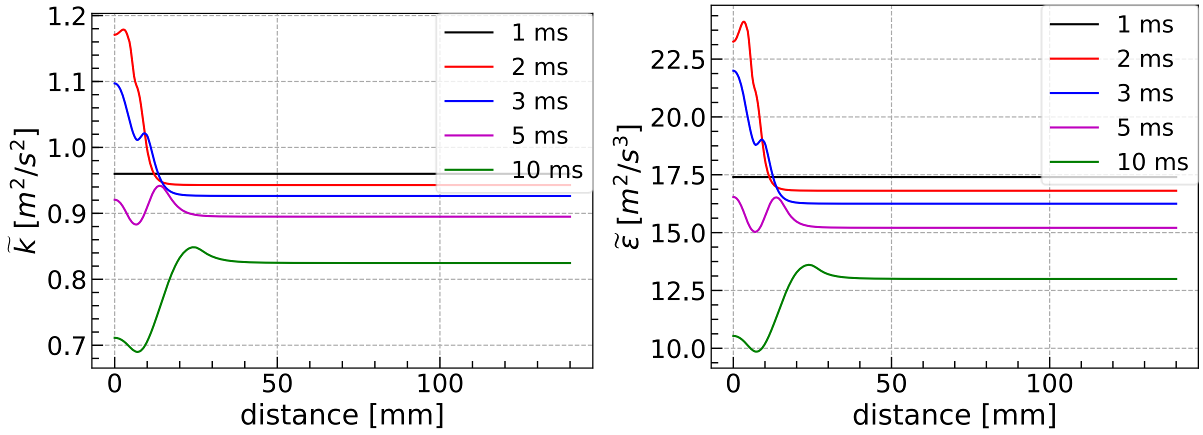
Parameters		value
unburned	T_u [K]	328
	W_u [g/mol]	32.80
	ρ_u [kg/m^3]	1.3214
	μ_u [kg/(m·s)]	1.8e-5
burned	T_b [K]	1500
	W_b [g/mol]	29.66
	ρ_b [kg/m^3]	0.2480
	μ_b [kg/(m·s)]	4.6e-5
others	$\sigma = \rho_u/\rho_b = (T_b W_u)/(T_u W_b)$ [-]	5.06
	S_L	0.12

Table Appendix III.2 Initial condition for cornflour explosion in Leeds fan stirred vessel.

parameters	value
T_0 [K]	328
P_0 [Pa]	110 000

Appendix III.I Extra source terms in standard $k - \varepsilon$ turbulence model

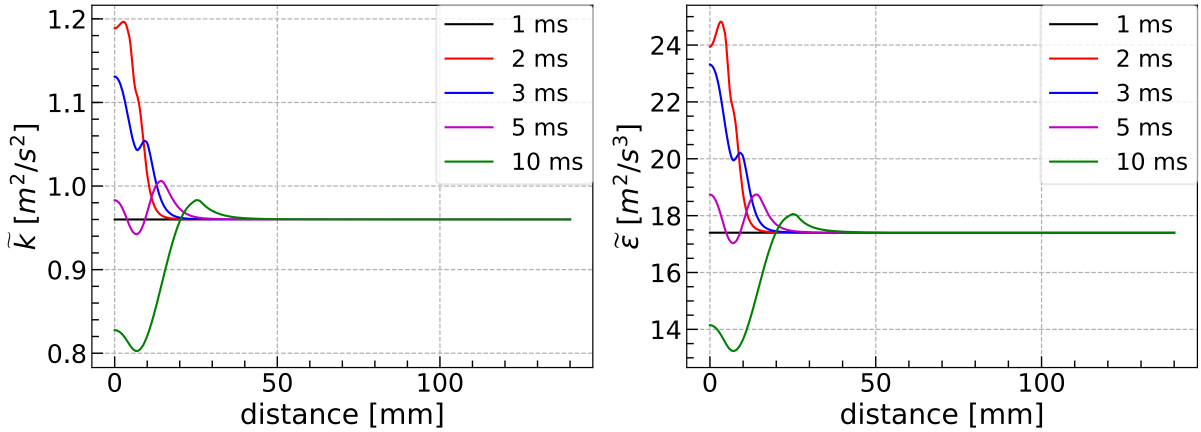
In the Leeds fan-stirred bomb experiment, a statistically stationary, spatially uniform, and isotropic turbulence is generated in the central zone of the vessel by four rotating fans. The rms velocity can be changed by varying the fan speed. If the fans are not included in simulations, which address solely the central region of the vessel, i.e. the region where the measurements were performed, the standard $k - \varepsilon$ turbulence model yields decaying turbulence; see Figure Appendix III.3. To mimic turbulence generation by the fans and to simulate statistically stationary turbulence, an extra source term $\bar{\rho}\tilde{\varepsilon}_0$ was added to the transport equations for $\tilde{k}$ and $\tilde{\varepsilon}$ following Lipatnikov and Chomiak [75]. Detailed implementation of this source term is described in Appendix VIII. Figure Appendix III.4 shows that, upstream of the flame, turbulence characteristics computed using the modified $k - \varepsilon$ model are statistically steady, in line with the measurements.



(a), turbulent kinetic energy

(b), turbulent dissipation rate

Figure Appendix III.3 Turbulent kinetic energy and dissipation rate computed using the standard $k - \varepsilon$ turbulence model at different instants. The initial $\tilde{k}$=0.96 m^2/s^2, $\tilde{\varepsilon}$=17.4 m^2/s^3, C_d=0.37, L_t=0.02 m, Turbulence model is activated at 1 ms.



(a), turbulent kinetic energy *(b), turbulent dissipation rate*

Figure Appendix III.4 Turbulent kinetic energy and dissipation rate computed using the modified $k-\varepsilon$ turbulence model at different instants. Other details are provided in caption to Figure Appendix III.3.

Appendix III.II Sensitivity study

Before performing computations of the Leeds experiments, sensitivity of numerical results to input parameters that are not well known should be investigated. There are three groups of such parameters.

First, the FSC model involves a single constant, i.e., A required to evaluate turbulent burning velocity, see Equation (Appendix I.5). For various gaseous flames, the use of the same value of $A = 0.4$ yielded good results [76]. However, the value of A for dust explosions could be different. Moreover, as already noted above, the value of the laminar flame thickness δ_L is not known for the dust-air mixture investigated in Leeds. However, because both A and δ_L are included in the same model equation, i.e., $U_T = Au'(\tau_t S_L/\delta_L)^{1/4}$, the lack of knowledge on δ_L could be compensated by tuning A. In particular, since dust particles should volatilize before burning, the laminar flame thickness of a dust-air mixture could be larger than the thickness of a gaseous laminar premixed flame characterized by the same S_L. In such a case, the use of the latter thickness could be compensated by adopting a lower $A < 0.4$.

Second, the $k-\varepsilon$ turbulence model involves a set of constants. For some of them, almost the same values are commonly adopted, but values of other constants depend substantially on conditions and, hence, are tunable. The latter group of constants includes turbulent Prandtl number Pr_t required to calculate fully-developed turbulent diffusivity, see Equations (Appendixes I.5-I.7), and a constant C_d required to link the mean dissipation rate $\tilde{\varepsilon}$ and turbulent length scale L (i.e. $\tilde{\varepsilon} = C_d\, u'^3/L$), e.g. when setting the initial conditions or evaluating $\tau_t = L/u'$ based on the computed $\tilde{k}$ and $\tilde{\varepsilon}$.

Third, to study dust explosion, the dust-air mixture should be ignited, but ignition of a flammable mixture in a turbulent flow is a very complicated phenomenon, which is beyond the scope of the present project. Accordingly, to mimic ignition an extra source (sink) term W_{ign} is inserted into the transport equation for progress (regress) variable. The term serves solely to rapidly create a small spherical kernel filled with combustion products. Accordingly, the model involves four more input parameters whose values could be varied. These are the source magnitude W_0, the kernel size σ_r, the ignition time t_0, and duration σ_t. Such a simplified method may be used, because (i) the Leeds experimental data were obtained from sufficiently large flame kernels [18] and (ii) the computed speeds of expansion of such large kernels are weakly affected by parameters of the ignition model, as already discussed earlier in the case of "frozen" turbulence. However, if turbulence evolution is simulated adopting the $k-\varepsilon$ model, then, short, highly localized, and very strong heat release associated with the term W_{ign} can strongly affect the computed fields of $\tilde{k}$ and $\tilde{\varepsilon}$. Such a numerical effect, in fact, significantly changes the initial conditions. However, this effect is a numerical artifact, because neither $k-\varepsilon$, nor another model can describe influence of strongly localized heat release on turbulence, as reviewed elsewhere [61, 62]. To circumvent the problem and avoid the discussed unphysical effects, the $k-\varepsilon$ model should be activated after ignition [75], i.e., when term W_{ign} becomes small (it decays rapidly with time at $t > t_0$). Accordingly, there is one more important input parameter, i.e., time of activation of the $k-\varepsilon$ model.

To summarize the above discussion, model constants and input parameters that were not varied in the present study are reported in Table Appendix III.3, whereas model constants and input parameters that were varied in the sensitivity study are shown in Table Appendix III.4.

Table Appendix III.3 Model constants and input parameters that were not varied in the present study.

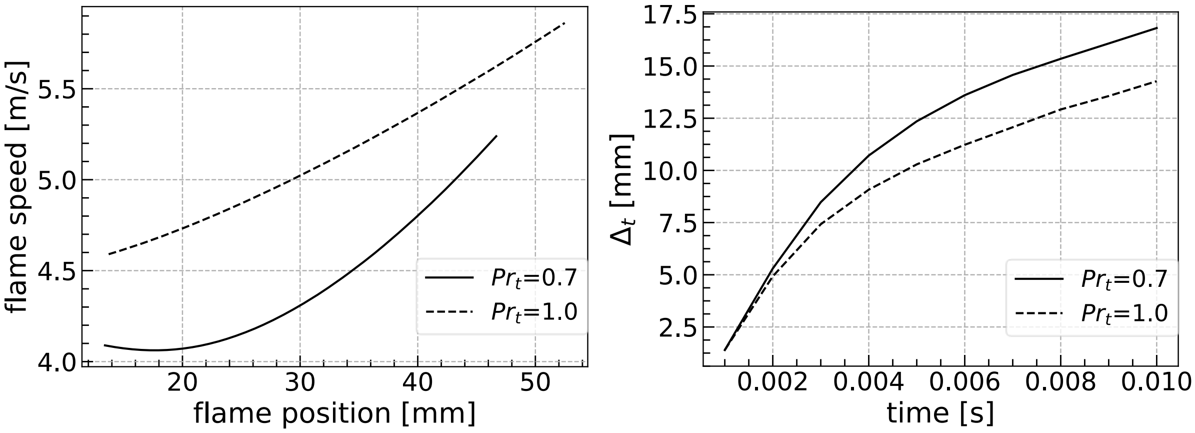
	Parameter	Value
Turbulence model	C_μ [-]	0.09
	C_1 [-]	1.44
	C_2 [-]	1.92
	σ_k [-]	1.0
	σ_ε [-]	1.3
Combustion model	t_r [s]	3.4e-11
	Θ [K]	2e4

Table Appendix III.4 Model constants and input parameters that were varied in the present study.

	Parameter	Value range	note
Ignition model	W_0 [-]	-	case dependent
	t_0 [s]	around 1 ms	
	σ_t [s]	$\sigma_t = t_0/5$	depends on t_0
	σ_r [m]	around 1 mm	related to ignition kernel
Turbulence model	Pr_t [-]	0.3-1.0	
	activation timing of turbulence model	1-10 ms	
	C_d [-]	0.37-2.0	
Combustion model	A [-]	0.2-0.5	0.4 for gas burning

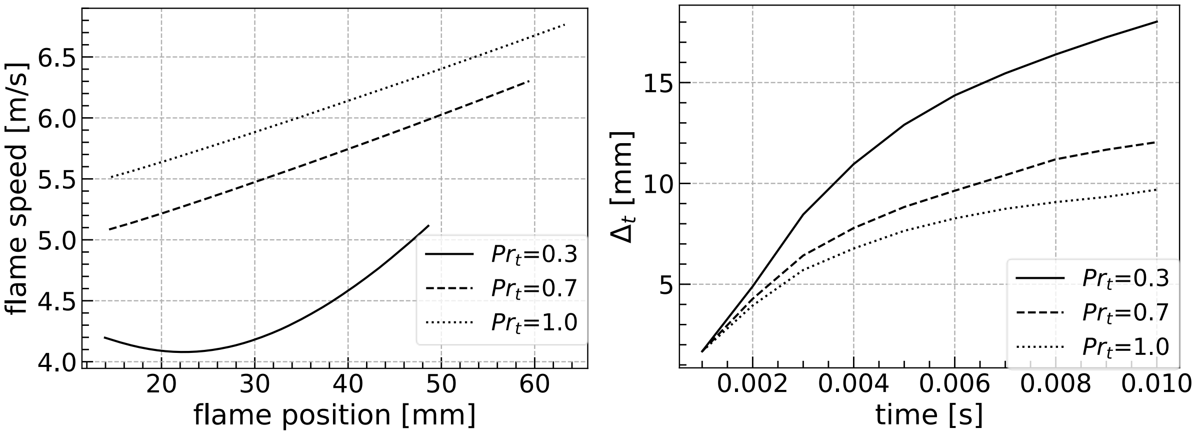
Appendix III.II.I Turbulent Prandtl number

Effect of turbulent Prandtl number on the computed flame speed and flame thickness is shown in Figures Appendix III.5 and Appendix III.6. The two figures report results computed with slightly different model constants, because higher values of C_d, $\tilde{\varepsilon}$ and σ_r, see Figure Appendix III.6, were required to obtain complete combustion in the centre at a low Pr_t. Accordingly, results obtained using $Pr_t = 0.3$ are not shown in Figure Appendix III.5, because the initial kernel was eroded due to turbulent diffusion for those values of C_d, $\tilde{\varepsilon}$ and σ_r (note that turbulence model was activated during ignition that case and, due to thermal expansion effects, the computed turbulent diffusivity was strongly increased) . Nevertheless, trends observed in Figures Appendix III.5 and Appendix III.6 are similar. A decrease in Pr_t results in increasing mean turbulent flame speed and decreasing mean flame brush thickness.



(a), flame speed *(b), flame thickness*

Figure Appendix III.5 Mean flame speed vs. mean flame position and mean flame brush thickness vs. time, obtained for different turbulent Prandtl numbers from 3-D statistically spherical flames with the initial $\tilde{k}$=9.077 m^2/s^2, the initial $\tilde{\varepsilon}$=1367.36 m^2/s^3, C_d=1.0, L_t=0.02 m, σ_r= 1.5e-3 m, W_0=1e14, t_0=1e-3 s, A=0.4. Turbulence model was activated from the beginning.

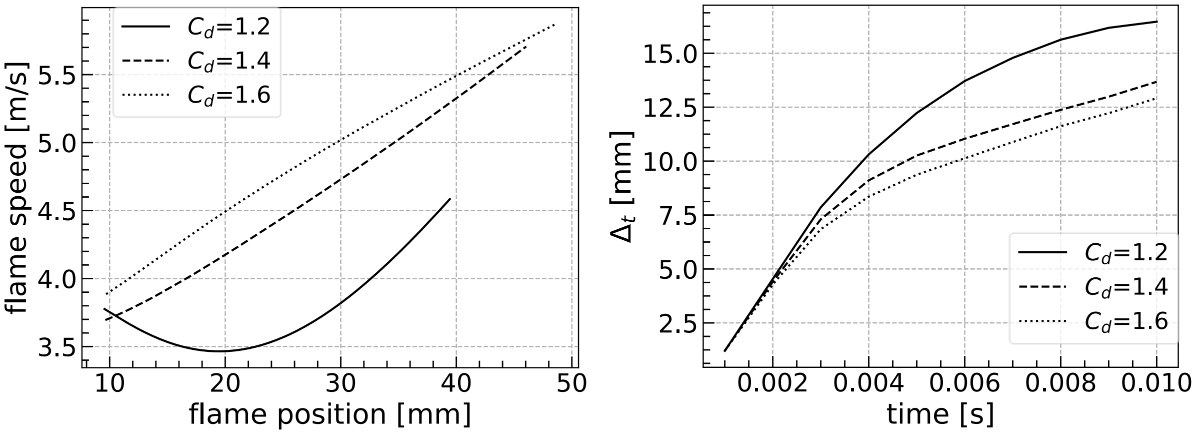


(a), flame speed *(b), flame thickness*

Figure Appendix III.6 Mean flame speed vs. mean flame position and mean flame brush thickness vs. time, obtained for different turbulent Prandtl numbers from 3-D statistically spherical flames with the initial $\tilde{k}$=9.077 m^2/s^2, the initial $\tilde{\varepsilon}$=2734.72 m^2/s^3, C_d=2.0, L_t=0.02 m, σ_r= 2e-3 m, W_0=1e15, t_0=1e-3 s, A=0.4. Turbulence model was activated at 2 ms.

Appendix III.II.II C_d coefficient

Effects of C_d coefficient on the computed results are shown in Figure Appendix III.7. An increase in C_d results in increasing the flame speed and decreasing the flame thickness. Thus, a decrease in Pr_t and an increase in C_d act in the same directions.

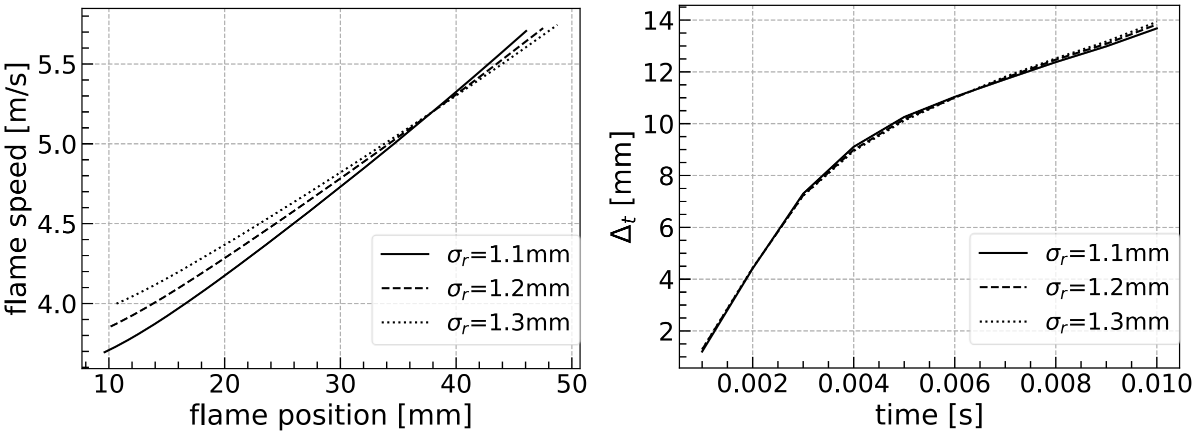


(a), flame speed *(b), flame thickness*

Figure Appendix III.7 Mean flame speed vs. mean flame position and mean flame brush thickness vs. time, obtained for different constants C_d from 3-D statistically spherical flames with the initial $\tilde{k}$=9.077 m^2/s^2, L_t=0.02 m, σ_r= 1.1e-3 m, W_0=1e15, t_0=1e-3 s, A =0.4, Pr_t =0.7. Turbulence model was activated at 2 ms.

Appendix III.II.III Ignition kernel size σ_r

Effects of the ignition kernel size σ_r on the computed flame speed and flame thickness are shown in Figure Appendix III.8. This parameter weakly affects the computed results provided that combustion in the center is complete, i.e., mean value of the progress variable is close to unity in the center after a transition time interval. However, if σ_r is small, the kernel is eroded due to turbulent diffusivity and $\tilde{c}(r = 0, t)$ decreases with time. In such a case, the computed flame speed is low, and the kernel shrinks. Thus, Figure Appendix III.8 indicates that the value of σ_r weakly affects the mean speed of a sufficiently large flame kernel provided the value of σ_r is sufficient to get the complete combustion in the center. This finding is in line with the results plotted in Figure Appendix II.15.



(a), flame speed *(b), flame thickness*

Figure Appendix III.8 Mean flame speed vs. mean flame position and mean flame brush thickness vs. time, obtained for different values of σ_r from 3-D statistically spherical flames with the initial $\tilde{k}$=9.077 m^2/s^2, L_t=0.02 m, σ_r= 1.1-1.3 mm, C_d=1.4, W_0=1e15, t_0=1e-3 s, A =0.4, Pr_t =0.7. Turbulence model was activated at 2 ms.

Appendix III.II.IV Timing for activating turbulence model

Effects of timing for activating turbulence model are shown in Figure Appendix III.9. The result indicates that earlier activation of the turbulence model yields a higher flame speed. The reason is that a higher turbulent kinetic energy is calculated when activating turbulence model early; see Figure Appendix III.10. However, this is so if combustion in the center is complete. Otherwise, earlier activation of the turbulence model can result in shrinking the kernel due to too intense turbulent diffusivity yielded by the $k - \varepsilon$ model.

It is also worth noting that the Leeds experiments did not reveal generation of turbulence after ignition. For instance, Bradley et al. [60] have stated that "*laser anemometry ahead of the flame showed ... no evidence of any significant change in* u'". Solid curve in Figure Appendix III.10 agrees qualitatively with the cited observation, thus, justifying activation of the $k - \varepsilon$ model after the end of ignition.

The timing for activating turbulence model has a minor effect on the computed flame thickness provided that combustion is complete in the centre. Accordingly, the figure is not reported here.

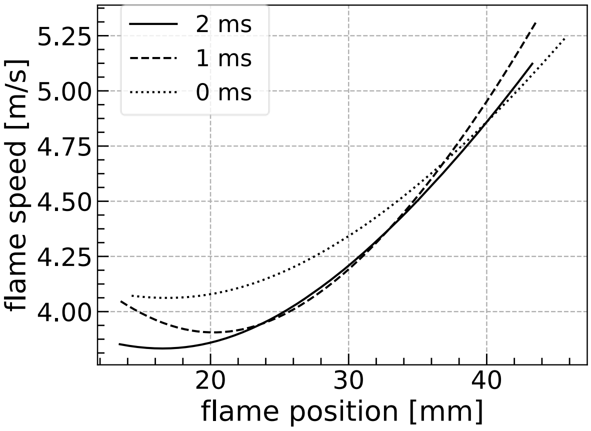


Figure Appendix III.9 Computed flame speed vs flame position for different timing for activating turbulence model for 3-D spherical flame with initial $\tilde{k}$=9.077 m^2/s^2, L_t=0.02 m, σ_r= 1.5e-3 m, W_0=1e14, t_0=1e-3 s, A =0.4, Pr_t =0.7. Turbulence model is activated at 0, 1 and 2 ms.

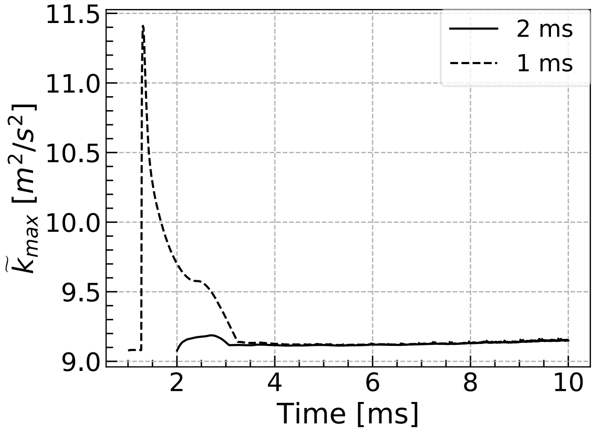


Figure Appendix III.10 Computed maximum turbulent kinetic energy $\tilde{k}$ vs. time for different timing for activating turbulence model. The initial $\tilde{k}$=9.077 m^2/s^2, L_t=0.02 m, σ_r= 1.5e-3 m, W_0=1e14, t_0=1e-3 s, A =0.4, Pr_t =0.7. Turbulence model is activated at 1 and 2 ms.

Appendix III.III Simulations of dust explosion in Leeds fan stirred explosion vessel

A summary of the initial turbulence characteristics for the model input is shown in Table Appendix III.5. The rest of the initial and boundary conditions for simulation are reported in Tables Appendix III.1-4.

Table Appendix III.5 Initial turbulence characteristics using C_d*=1.2 and* L_t*=0.02 m.*

Cases	Case 1	Case 2	Case 3	Case 4
u' [m/s]	0.80	1.62	2.45	3.31
$\tilde{k}$ [m^2/s^2]	0.96	3.94	9.004	16.43
$\tilde{\varepsilon}$ [m^2/s^3]	56.44	468.63	1621.01	3997.35

Figures 6 and 7 summarize the most important findings of the model validation. Numerical results show that the model well predicts (i) an increase in the apparent turbulent flame speed by the rms turbulent velocity at moderate turbulence and (ii) a slow increase in the flame speed with growth of the mean flame radius. Accordingly, this indicates that the adapted dust explosion model is an appropriate building block for developing an advanced numerical tool for CFD research into large-scale explosions of fine dust particles with high volatile contents.

Appendix IV. Simulation of an industrial large-scale vented dust explosion

In this section, results of simulations of measurements of the vented corn starch dust explosion in a 11.5 m^3 vessel, performed in Rembe Research and Technology Center, are presented. This work is performed in close collaboration with colleagues from the IND EX® who provide the project members with experimental data. The study consisted of three stages. First, the input parameters for the adopted computational model, such as the thermophysical properties of corn starch dust and the laminar burning velocity were investigated. Second, the sensitivity of the model input parameters to the computed results was studied. Finally, the computed and measured explosion overpressures were compared.

Appendix IV.I Experimental setup

Corn starch vented explosion experiments were carried out at Rembe Research and Technology Center during 2017 and 2018 with an aim of studying the effect of vent geometry on the vent efficiency [24, 77]. The motivation for performing these test series was the lack of requirements on the vent geometry in the current explosion protection regulations, e.g., NFPA 68:2018 or EN 14491:2012, except for vent size [24, 77].

An 11.5 m^3 vessel at the REMBE® Research + Technology Center in Brilon, Germany was used to perform the vented corn starch dust explosions. The corn starch used was a St1 dust having a K_{St}-value of 220 bar·m/s ± 15 %. The corn starch dust has a median diameter D_{50} of 97 µm and a moisture content of 9.87 %. The calculated and applied vent area was set to A_v= 0.5 m^2 in a circular shape. The vent opening was closed with a layer of 70 µm low mass aluminium foil with a static activation pressure (p_{stat}) of 0.1 ± 15 %.

To create an explosive atmosphere inside the vessel two pressurized dust containers were used for blowing the dust into the test vessel. A dust concentration of 750 g/m^3 was chosen. An ignition delay of 800 ms was selected via multiple tests in order to achieve the required K_{St}-value with the above dust concentration.

The resulting explosive atmosphere was ignited using a pair of pyro-technique igniters with a total ignition energy of 10 kJ in the center of the test vessel. Two pressure detectors P1 and P2 were installed to measure the reduced maximum explosion overpressures (p_{red}) inside the vessel (see Fig. Appendix IV.1). The data was recorded using an oscilloscope (Tektronix TDS 2014C). A snapshot of the vented corn starch dust explosion is shown in Fig. Appendix IV.2.

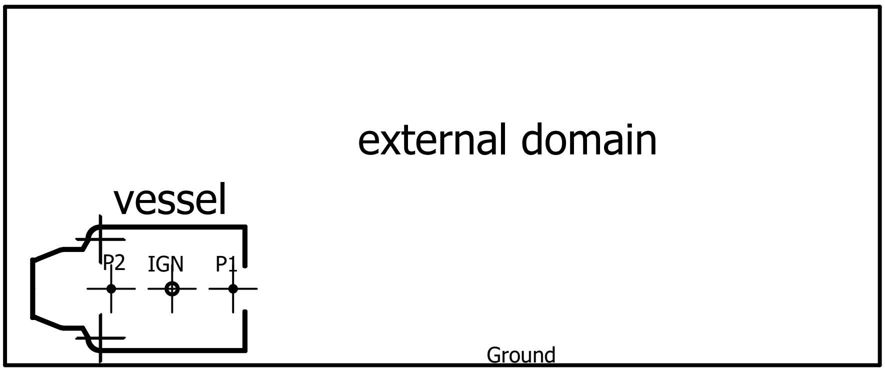


Figure Appendix IV.1 2D drawing of Rembe explosion vessel and the computational domain.

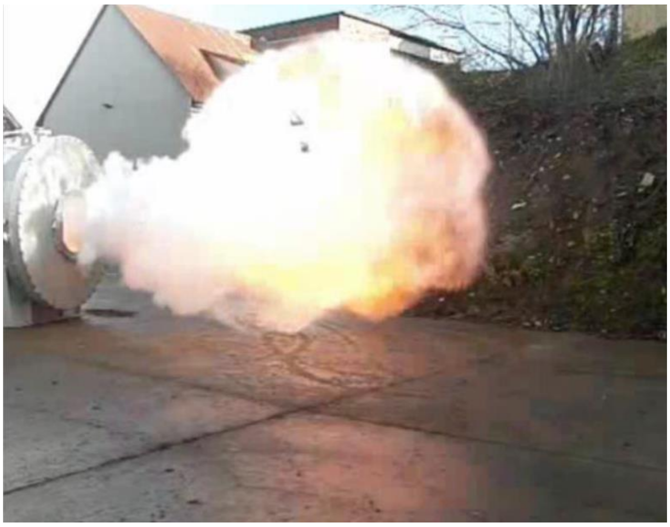


Figure Appendix IV.2 Vented corn starch dust explosion in the 11.5 m³ vessel.

Appendix IV.II Numerical setup

The vented explosion simulations were performed for a computational domain of 15.5 m × 5 m × 6.355 m which include the volume of the vessel and a volume outside of the vessel to capture the venting process. Only half of the vessel and the outside volume is simulated to save computational time with the assumption of symmetric condition with respect to the vertical plane.

The CAD geometry of the vessel was provided by Rembe as stp files, and the files were read in an open source 3D CAD modelling tool FreeCAD [78]. The detailed geometry of the vessel was obtained in FreeCAD, and the geometry of the vessel shell was exported in a stl file in the ASCII format. The geometry is then imported into the OpenFOAM, and the computational mesh was generated using a so-called snappyHexMesh tool in OpenFOAM. A detailed description of importing geometry in OpenFOAM and generating the computational mesh is reported in Appendix IX.

There are two sets of computational meshes used for the simulations. The first mesh is the one used for simulating dust explosion before the vent panel ruptures; see Figure Appendix IV.3. This computational mesh consists of three blocks, i.e., (i) a sphere with a centre of ignition location and a radius of 0.15 m and with a mesh size of 6.5 mm, (ii) a sphere of a radius 0.7 m with a mesh size of 12.5 mm, and (iii) the rest of the domain with a mesh size of 25 mm. It is worth noting that the mesh near the walls were refined with a size of 12.5 mm to better resolve the parameters on the walls. The total number of the first cell size is approximately 1.35 million. It took about 3.5 h (wall-clock time) for running a simulation of duration of 0.12 s on 10 processors with a maximum Courant number being 0.05.

The second mesh is the one used for simulating dust explosion after the vent panel ruptures; see Figure Appendix IV.4. The simulation is stopped when the pressure inside the vessel reaches a critical pressure, i.e., 0.1 bar, and the results are saved. Then, the results are mapped to the second mesh with the pressure and temperature outside of the vessel equal to 1 atm and 273 K, respectively. We are not aware on such “two-mesh” simulations performed by other research groups. The second mesh includes mesh sizes of four levels, i.e. (i) a sphere with a radius of 0.7 m with a mesh size of 12.5 mm, (ii) the inside of the vessel and a cylindrical domain with a length of 1 m from the vent opening with a mesh size of 25 mm, (iii) a cylindrical domain in the far-field with a mesh size of 50 mm, and (iv) the rest of the domain with a mesh size of 100 mm. It is worth noting that the vessel surfaces were resolved with a finer cell size, especially near the wall where vent panel is located, which has 5 layers of cell on the surface yielding an average y_+ value of around 50. Note that y_+ is a non-dimensional distance. It is often used to describe how coarse or fine a mesh is for a particular flow pattern. The total number of the second mesh cells is approximately 2.43 million cells. It took about 5 days (wall-clock time) for running a simulation of duration of 0.18 s on 28 processors with a maximum Courant number being 0.05.

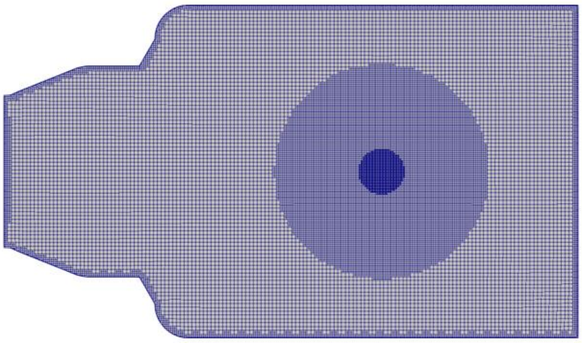


Figure Appendix IV.3 Computational mesh of Rembe vented explosion vessel. This mesh is used before the rupture of the vent panel.

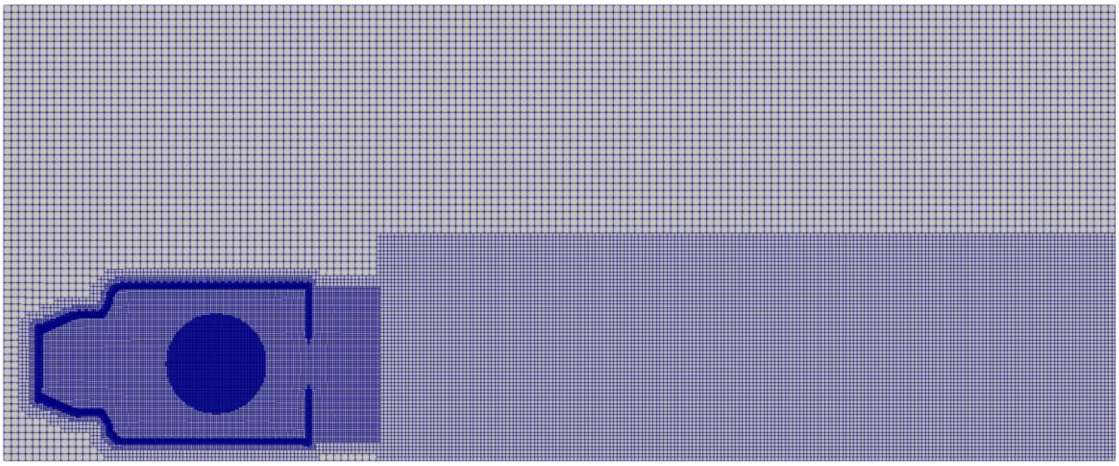


Figure Appendix IV.4 Computational mesh of Rembe vented explosion vessel and the area outside of the vessel. This mesh is used after the rupture of the vent panel.

Appendix IV.III Thermophysical properties of corn starch dust

Thermophysical properties of dust are required for calculating the mass and heat transfer processes in a CFD simulation. These properties involve chemical formula for knowing the molecular weight, heat of reaction, standard heat of formation, specific heat capacity and adiabatic flame temperature. Following Bradley et al. [18] and Sattar et al. [79], the chemical formula of $C_6H_{7.88}O_{4.98}$ is adopted here.

The standard heat of formation

The standard heat of formation, $H_{f,}^{\ominus}$ is defined by the change of enthalpy for the formation of 1 mol of a compound from its component elements when the component elements are each in the standard states. The standard state means the 1 atm pressure and 298.15 K temperature. The standard enthalpy of formation of any element in its most stable form is zero by definition, e.g., H_2, O_2, C(s), N_2.

A negative enthalpy of formation means that the compound has lower enthalpy, more stable as compared to its elements, heat should be released to form this compound in an exothermic reaction. In contrast, a positive enthalpy of formation means that the compound has a higher enthalpy, less stable as compared to its elements, heat should be added to form this compound, or endothermic reaction.

Heat is released during the burning of the corn starch dust particle and air mixture by assuming a complete combustion (see Eq. (Appendix II.1)).

Due to the energy balance, the following equation holds

$$n_{cs}H^0_{f,cs} + n_{cs}\Delta H_{reaction} = n_{CO2}H^0_{f,CO2} + n_{H2O}H^0_{f,H2O} \quad \text{(Appendix IV.1)}$$

where $H_{f,cs}^{\Theta}$, $H_{f,CO2}^{\Theta}$=-393.51 kJ/mol [80], $H_{f,H2O}^{\Theta}$=-241.826 kJ/mol [81] represents the standard heat of formation for corn starch, CO_2 and H_2O in gas phase, respectively; $\Delta H_{reaction}$=-2521.048 kJ/mol, is the heat of reaction measured using calorimetry for corn starch by Bradley et al. [18]; n_{cs}, n_{CO2}, and n_{H2O} are mole numbers of corn starch, CO_2 and H_2O in a complete stochiometric chemical reaction in Eq. (Appendix II.1), i.e., 1, 6, and 3.94, respectively. According to Equation (Appendix IV.1), the standard heat of formation of corn starch is $H_{f,cs}^{\Theta}$=-792.644 kJ/mol or -4.97E+03 kJ/kg using W_{cs}= 0.15956 kg/mol.

Specific heat capacity

The specific heat capacity of a normal maize starch with a amylose and moisture content of 28 % and 10.5 %, respectively, mixed with water and glycerol, was measured by Tan et al. [82]. Here, we assume that the maize starch is equivalent to corn starch. Therefore, the term corn starch is used in the following text with an aim of consistence. The specific heat capacity of corn starch (disregarding water and glycerol contents) within a temperature range of 40-120 °C is as follows [82]

$$c_{p,cs} = 5.24T - 170.52 \qquad \text{(Appendix IV.2)}$$

where $c_{p,cs}$ is the specific heat capacity of corn starch in J/(kg·K) and T is the temperature in K.

In order to use the specific heat capacity and the standard heat of formation of corn starch in the specific OpenFOAM library adopted in this project, a brief explanation of how these parameters are used is presented. An OpenFOAM thermo type janafThermo, is used in this work, in which the heat capacity c_p [J/(kg K)] and the so-called absolute enthalpy H_a [J/kg], can be calculated using the NIST-JANAF polynomial equations as follows

$$c_{p,cs} = R_{spec,cs}\left(a_{4,cs}T^4 + a_{3,cs}T^3 + a_{2,cs}T^2 + a_{1,cs}T + a_{0,cs}\right) \qquad \text{(Appendix IV.3)}$$

$$H_{a,cs} = R_{spec,cs}\left(\frac{a_{4,cs}}{5}T^5 + \frac{a_{3,cs}}{4}T^4 + \frac{a_{2,cs}}{3}T^3 + \frac{a_{1,cs}}{2}T^2 + a_{0,cs}T + a_{5,cs}\right) \qquad \text{(Appendix IV.4)}$$

where $R_{spec,cs} = R_0/W_{cs}$=52.11 is the specific gas constant for corn starch in J/(kg·K); R_0=8.314 J/(mol·K) is the ideal gas constant; W_{cs}= 0.15956 kg/mol, is the molecular weight of corn starch based on the chemical formula $C_6H_{7.88}O_{4.98}$ by Bradley et al. [18]; $a_{0,cs}$, $a_{1,cs}$, $a_{2,cs}$, $a_{3,cs}$, $a_{4,cs}$ and $a_{5,cs}$ are the JANAF coefficients.

By comparing Equations (Appendix IV.2) and (Appendix IV.3), we get $a_{2,cs} = a_{3,cs} = a_{4,cs}$=0; $a_{0,cs} = -170.52/52.11 = -3.2726E + 00$ $a_{1,cs} = 5.24/52.11 = 1.0056E - 01$.

By substituting T=298.15 K, $H_{f,cs}^{\Theta}$-4.97E+03 kJ/kg and the values of $a_{0,cs}$, $a_{1,cs}$, $a_{2,cs}$, $a_{3,cs}$, $a_{4,cs}$ into Equation (Appendix IV.4), we get $a_{5,cs} = H_{f,cs}^{\Theta}/R_{spec,cs} - \frac{a_{1,cs}}{2}T^2 - a_{0,cs}T$=-9.9808E+04. To summarize, the JANAF coefficients for corn starch including $a_{0,cs}$, $a_{1,cs}$, $a_{2,cs}$, $a_{3,cs}$, $a_{4,cs}$, $a_{5,cs}$ and $a_{6,cs}$ are -3.2726, 0.10056, 0, 0, 0, -99808, 0. Note that the entropy offset $a_{6,cs}$ is set equal to zero since it is not used here.

Dependence of the calculated adiabatic flame temperature on the equivalence ratio of corn starch dust and air mixture is shown in Figure Appendix IV.5. A utility based on the standard utility adiabaticFlameT was made to perform the calculation with the thermophysical properties from the above-mentioned sources and initial pressure and temperature equal to 1 atm and 273 K. In contrast, there are no documented thermophysical properties or initial conditions for flame temperature calculation in the work by Sattar et al. [79]. Nevertheless, the calculated flame temperatures agree well with each other especially in the range of lean mixture, which ensures the reasonableness of the estimated thermophysical property data.

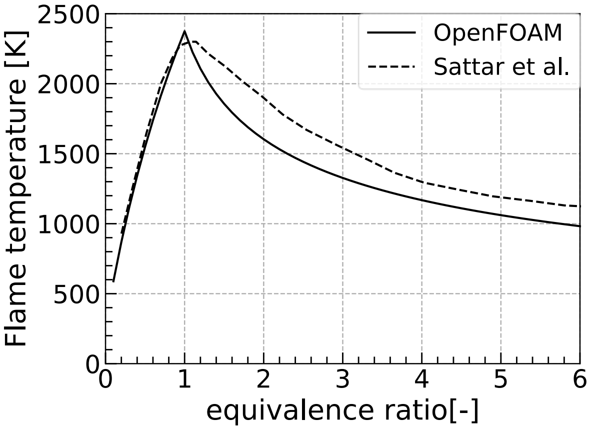


Figure Appendix IV.5 Comparison of adiabatic flame temperature calculated using OpenFOAM with initial pressure and temperature being equal to 1 atm and 273 K, respectively, and the calculated data by Sattar et al. [79].

Appendix IV.IV Laminar burning velocity of corn starch dust

In comparison with the measured data on the laminar burning velocity of gases, the measured data for the dust contains a larger uncertainty margin even for the most studied dust corn starch. For many dusts the measured laminar burning velocity is not available. Such a knowledge gap is associated with dust properties. First, there is an increased difficulty in experiments using high-speed schlieren photography [83] with dust-air mixture which has unfavourable optical properties. Second, there is a difficulty in balancing between dust settlement and a laminar flow in the measurements [84-86]. Third, the dust properties such as particle size and moisture content are factors that influence the laminar burning velocity of a dust-air cloud. Nevertheless, a summary of a few available measured corn starch laminar burning velocity data is reported here.

Leeds data from Bradley et al. [18]

Bradley et al. [18] reported a laminar burning velocity of 0.12 m/s for a dust concentration of 0.26 kg/m^3. Those authors measured turbulent burning velocities at different turbulence levels in a 305 mm diameter spherical vessel using high speed Schlieren technique. The laminar burning velocity was obtained by extrapolating the turbulent burning velocities to zero turbulence level. There are several uncertainties in the measured laminar flame speed. First, the amount of dust which participates in the combustion is less than the injected dust, as stated in the paper, i.e., "*no less than 10%, and possibly about 20% of the dust had adhered to such surfaces.*" It is unclear how much dust was mixed with the air during the combustion. This problem was discussed by Skjold [87] who proposed a method of calculating the burned fraction of dust by processing data obtained from a 20 l vessel. This calculation uses the dust chemical composition, specific heat capacity, heat of formation, and product composition from simplified chemical equilibrium calculations. Second, Dahoe et al. [88] pointed out that the burning velocities measured by Bradley et al. [18] in the fan-stirred vessel are relatively low as compared to that of a planar flame. The reason is due to the flame stretch and curvature effects [88].

Leeds data from Sattar et al. [79]

Sattar et al. [79] estimated the laminar burning velocity of corn starch dust using measurements in the 1 m^3 dust explosion vessel; see Figure Appendix IV.2. The measured data was fitted using a 4th order polynomial function; see the dashed line in Figure Appendix IV.6. A turbulence factor was determined from methane gas explosions. Then, this turbulence factor was applied directly to estimate the dust laminar burning velocity in the same vessel. The laminar burning velocity of corn starch was presented versus corrected dry ash free equivalence ratio. However, there are still uncertainty in the reported data from Sattar et al. [79]. First, directly applying such turbulence factor obtained in gas explosion to dust explosion is questionable. The reason is that there are at least three differences between gas and dust explosions, such as turbulence level before ignition, flame thickness and flame development process. Second, as described in the cited paper [79], the corrected equivalence ratio includes only dust which participated in the explosion. Without knowing the fraction of dust which was burned and the chemical formula used in this experiment for corn starch, it is not possible to convert the data from Sattar et al. [79] to a unit of dust concentration in kg/m^3 in order to make a proper comparison with other data. Third, in Ref. [79], there is no information about the dust properties, such as dust particle size or moisture contents, which are essential for the explosion characteristics.

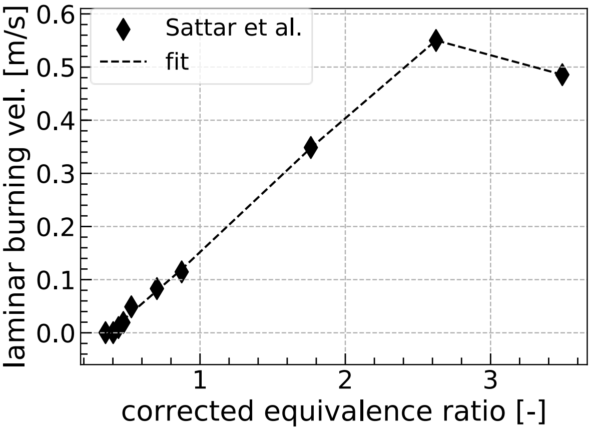


Figure Appendix IV.6 Measured and fitted laminar burning velocity of corn starch dust versus corrected equivalence ratio by Sattar et al. [79].

Data from Dahoe et al. [88]

The corn starch laminar burning velocity was measured by Dahoe et al. [88] in a powder burner with laser Doppler anemometry. The powder air mixture flows upwards and forming a stable dust flame. The unstretched laminar burning velocity of corn starch dust is between 0.15 and 0.3 m/s for concentrations between 0.26 and 0.38 kg/m^3. However, there is no information on the corn starch dust properties, such as chemical formula, particle size or moisture content. The measured laminar burning velocity of corn starch dust versus corrected equivalence ratio and dust concentration by Dahoe et al. [88] is shown in Figure Appendix IV.7. The equivalence ratio was recalculated based on the mass fraction burnt data in Ref. [89, 90].

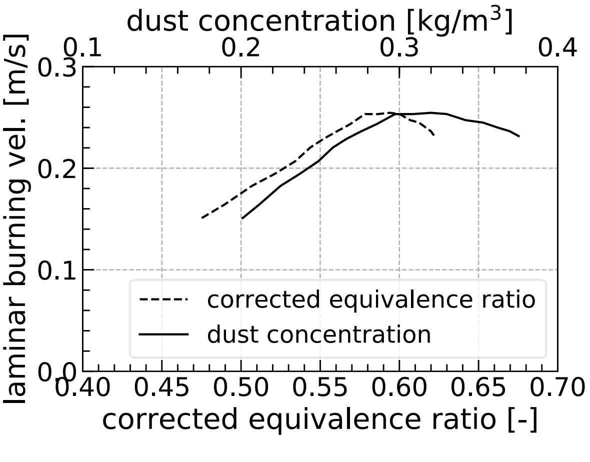


Figure Appendix IV.7 Laminar burning velocities of corn starch, measured by Dahoe et al. [88], versus corrected equivalence ratio and dust concentration.

Data from Skjold et al. [91]

Skjold et al. [94] estimated turbulent burning velocity of dust using a pressure time trace measured in the 20-l explosion vessel. Skjold et al. [91] calculated the laminar burning velocity of dust by using an empirical correlation between turbulent and laminar burning velocity. However, the turbulent burning velocity measured in 20 L vessel and the empirical correlation involve uncertainties, which put into question the accuracy of estimated laminar burning velocity. Figure Appendix IV.8 shows the laminar burning velocity of corn starch dust versus corrected equivalence ratio and dust concentration.

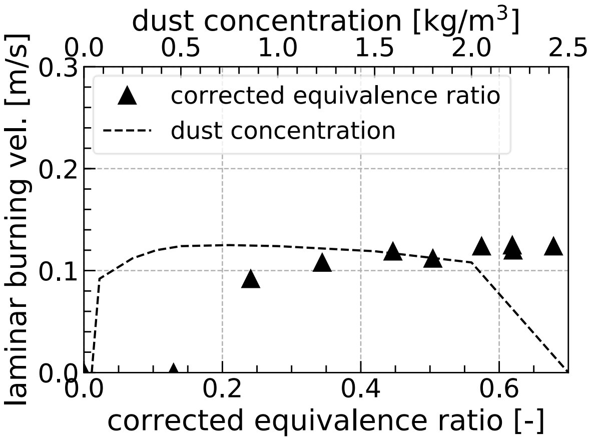


Figure Appendix IV.8 Laminar burning velocity of corn starch dust versus corrected equivalence ratio and dust concentration in Ref. [89, 90].

Summary

To summarize, a comparison of the data from different sources is presented in Figure Appendix IV.9. Note that the data from Bradley et al. [18], Dahoe et al. [88] were originally reported based on the dust concentration. In Figure Appendix IV.9, the dust concentration is converted to a corrected equivalence ratio using the mass fraction burnt data in Ref. [89, 90].

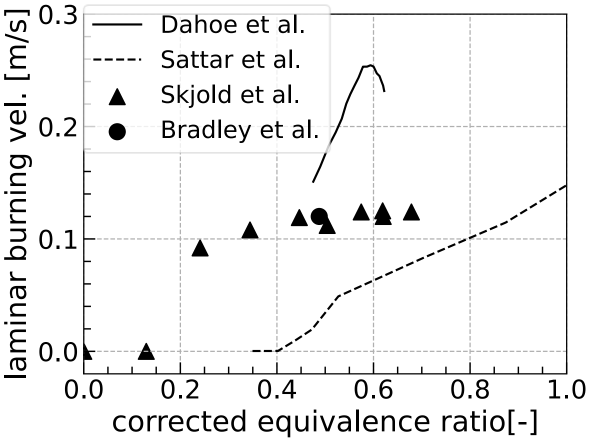


Figure Appendix IV.9 Comparison of laminar burning velocities reported by Dahoe et al. [88], Sattar et al. [79], Skjold et al. [89, 90] and Bradley et al. [18].

Appendix IV.V Sensitivity to input parameters of numerical model

Turbulence model

Turbulence has important effect on the burning process. In order to quantitatively compare the effect of different turbulence models on the computed results, two parameters were compared, i.e., a normalized flame arrival time and a normalized time of the rupture of the vent panel. The flame arrival time is defined as the time instant when the flame front characterized by a Reynolds-averaged combustion progress variable $\bar{c}$ being equal to 0.5 arrives at the vent opening. The vent panel rupture time is defined as the time instant when the pressure increase is equal to 0.1 bar at P2 position shown in Figure Appendix IV.1. Both the flame arrival time and the vent panel rupture time are normalized by the values obtained using the k-epsilon model.

Figures Appendix IV.10 and Appendix IV.11 show the effect of different turbulence models (implemented into OpenFOAM) on the normalized flame arrival time and the normalized vent panel rupture time, respectively, for a relatively high initial turbulence velocity fluctuation (the initial rms velocity u'=5.8 m/s). Figures Appendix IV.12 and Appendix IV.13 show the effect of different turbulence models on the normalized flame arrival time and the normalized vent panel rupture time, respectively for a relatively low initial turbulence velocity fluctuation (u'=0.5 m/s). Most of the RANS turbulence models available in OpenFOAM-v1812 were tested here. Note that the sensitivity study was performed for the stoichiometric methane-air mixture, because those simulations were run before or in parallel with investigation of available data on the properties of dust-air mixtures. Differences between results obtained adopting different turbulent models are more pronounced in the low turbulence level case. It is worth noting that a relatively low Courant number of 0.05 - 0.1 was used here to get converged results.

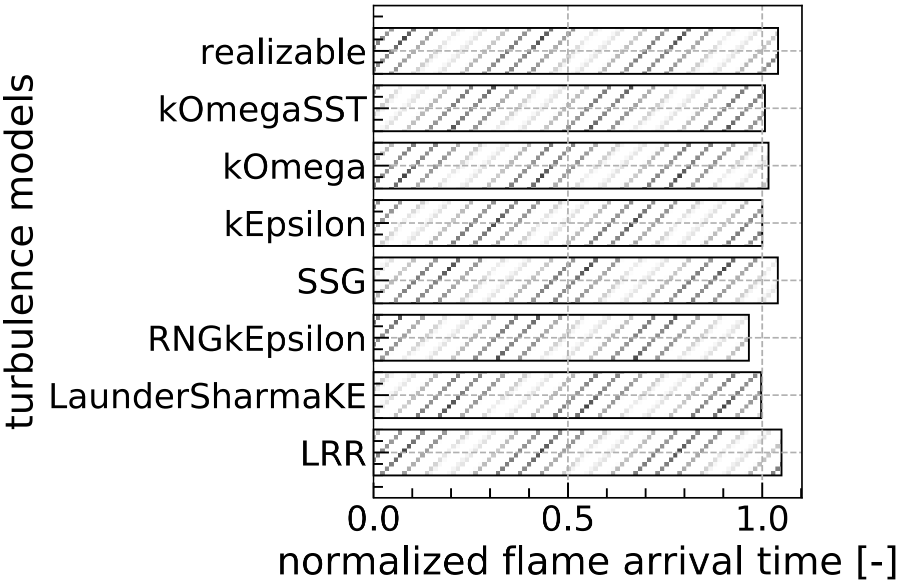


Figure Appendix IV.10 Effect of different turbulence models on the normalized flame arrival time. u'=5.8 m/s, L=0.1 m, Co=0.1, a whole domain with 2.4 million mesh with a stoichiometric methane-air mixture in the vessel.

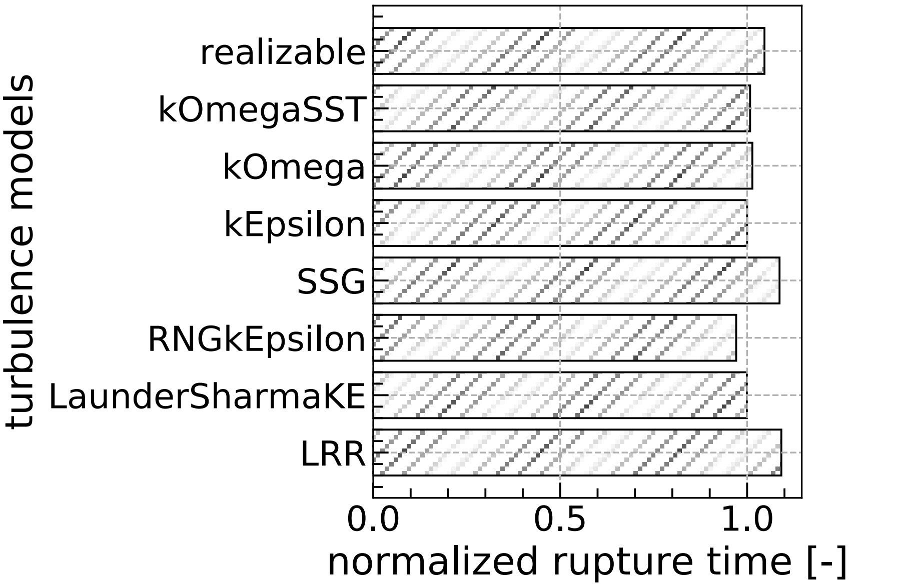


Figure Appendix IV.11 Effect of different turbulence models on the normalized vent panel rupture time. The same parameters as in Figure Appendix IV.10 were used.

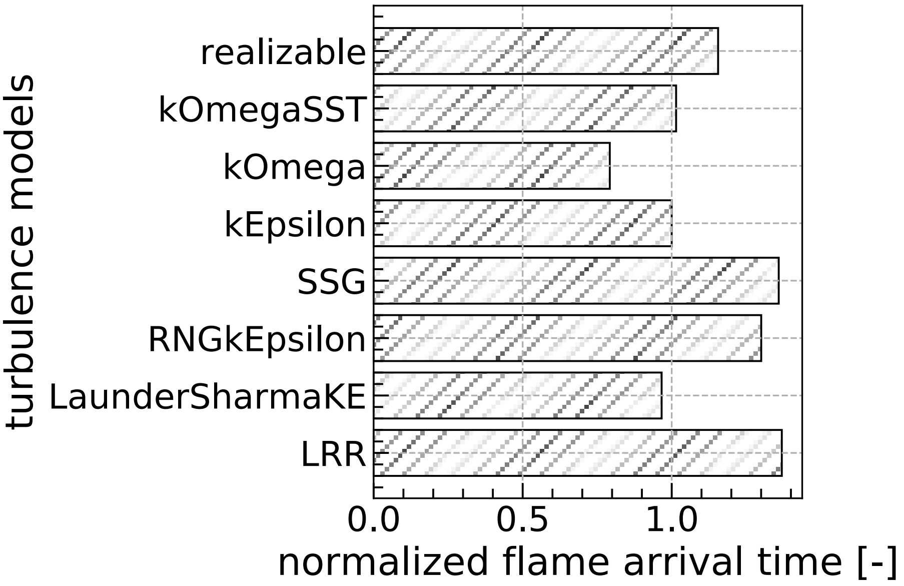


Figure Appendix IV.12 Effect of different turbulence models on the normalized flame arrival time. u'*=0.5 m/s,* L*=0.1 m,* Co*=0.1, a whole domain with 2.5 million mesh with a stoichiometric methane-air mixture in the vessel.*

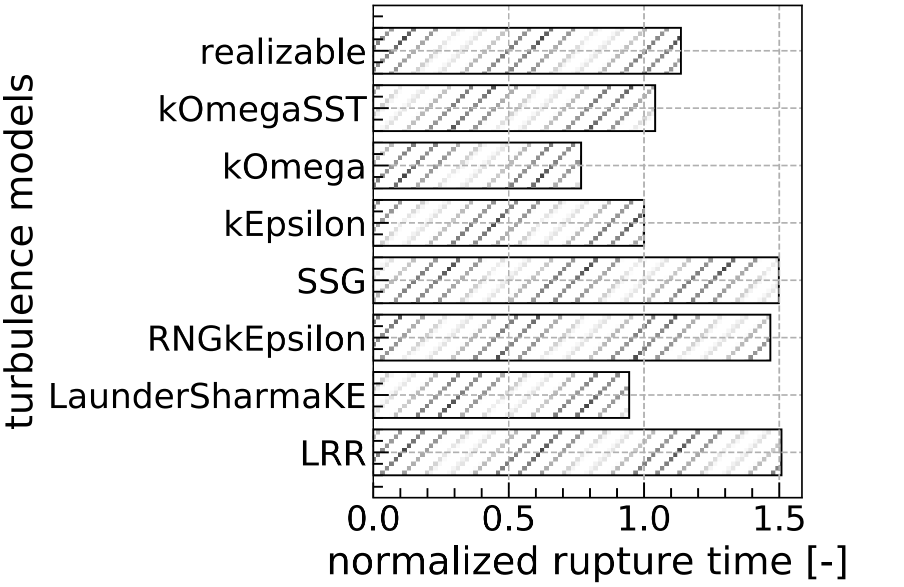


Figure Appendix IV.13 Effect of different turbulence models on the normalized vent panel rupture time. The same parameters as in Figure Appendix IV.12 were used.

Sensitivity study for other parameters

A sensitivity study was also performed for other parameters including initial turbulent kinetic energy, turbulent length scale, turbulent Prandtl number, ignition model parameters, C_d, and burned temperature. In order to compare the effect of different input parameters, a sensitivity coefficient $c_{sen,i}$ is defined as follows

$$c_{sen,i} = \frac{\frac{y_i - y_0}{y_0}}{\frac{x_i - x_0}{x_0}} \qquad \text{(Appendix IV.5)}$$

where, the subscript i is the i-th parameter for the investigation; the subscript 0 corresponds to a reference value; x is an input parameter; and y is the numerical result, i.e. the time instant when the flame arrives at the vent opening. It can be inferred from Equation (Appendix IV.5) that a positive sensitivity coefficient means that an increase in an input parameter results in increasing the flame arrival time.

Figure Appendix IV.14 shows sensitivity coefficients for various input parameters. In the present case, the turbulent Prandtl number and burned temperature have the most pronounced effect on the flame arrival time.

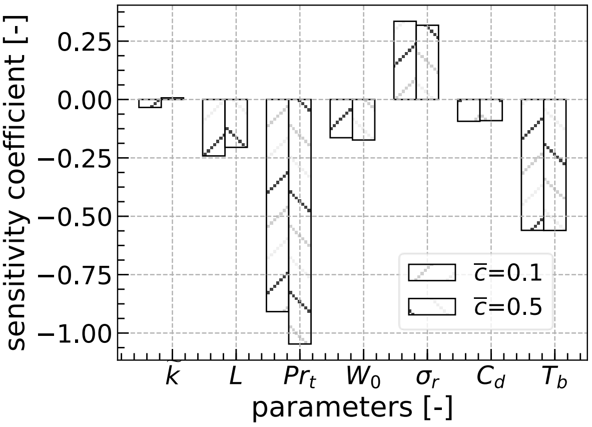


Figure Appendix IV.14 Sensitivity coefficients on different input parameters on the computed flame arrival time at the vent opening based on $\bar{c}$=0.1 and $\bar{c}$=0.5. u'=0.5 m/s, L=0.1 m, Co=0.1, a whole domain with 2.5 million mesh with a stoichiometric methane-air mixture in the vessel.

Appendix IV.VI Comparison of turbulence models

Turbulence is well known to substantially affect burning process. Evolutions of corn starch dust explosion overpressures computed using the conventional FSC model, i.e., Eqs. (Appendix I.3) and (Appendix I.4), and different turbulence models are reported in lines in Fig. Appendix IV.15, with black circles showing the experimental data. By investigating these experimental data, four different stages could be found: (i) an increase in the overpressure during dust explosion in the closed vessel before the rupture of the vent panel, i.e., at $t < 0.1$ s, (ii) a decrease in the overpressure after the rupture of the vent panel, i.e., at $0.1 < t < 0.12$ s, followed by a slow increase in the overpressure at $0.12 < t < 0.16$ s, (iii) a rapid increase in the overpressure at $0.16 < t < 0.30$ s, and (iv) a decrease in the overpressure at $t > 0.31$ s. Results measured or computed during stages (i) and (ii) are zoomed in Fig. Appendix IV.16 (a) and (b), respectively. During stage (i), all explosion products are confined to the closed vessel. During stage (ii), the products appear outside the vessel. During stage (iii), the explosion kernel grows outside the vessel.

Figure Appendix IV.15 shows that realizable (cyan long-dashed lines), k-epsilon (black dashed lines), and Launder-Sharma (yellow short-dashed lines) turbulence models belong to a group which yields a fast pressure rise rate during stages (i)-(iii) of the dust explosion and the highest explosion overpressures. On the contrary, SSG turbulence model yields the slowest rate of pressure rise (green dotted lines) and the lowest explosion overpressure. During stage (ii), pressure drop computed using SSG model is significantly overestimated, with subsequent computed pressure increase being too slow. During stage (i) disagreement between the measured data and the results computed using k-epsilon, realizable, Launder-Sharma, or SSG turbulence model can be handled by adjusting the initial turbulence characteristics, which significantly affect simulated pressure curves (see Figs. Appendix IV.17 and Appendix IV.18). However, during stage (ii), such a disagreement is much more difficult to handle.

From this perspective, results obtained using RNG (red dotted lines), k-omega (blue dotted-dashed lines), and k-omega-SST (violet double-dotted-dashed lines) turbulence models appear to most promising during stage (ii), as shown in Fig. Appendix IV.16 (b). However, RNG turbulence model yields too fast pressure rise rate before the rupture of the vent panel (see Fig. Appendix IV.16 (a)). If the first stage ($t < 0.1$ s) and the second stage ($0.1 < t < 0.16$ s) of the dust explosion are considered jointly, k-omega and k-omega-SST turbulence models belong to a group which yields a reasonable agreement with the experimental data. Accordingly, these two turbulence models are selected for further study, with this choice being mainly based on quite moderate rate of pressure rise, yielded by these two models at $0.1 < t < 0.2$ s (see blue and volet dotted-dashed lines in Fig. Appendix IV.16 (b)).

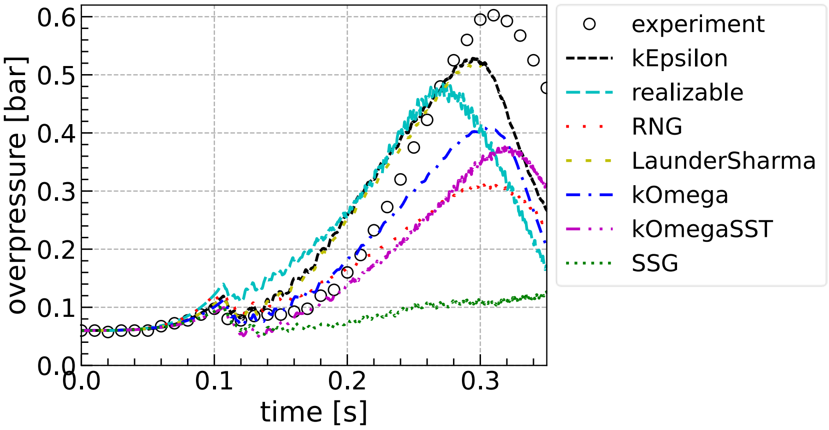
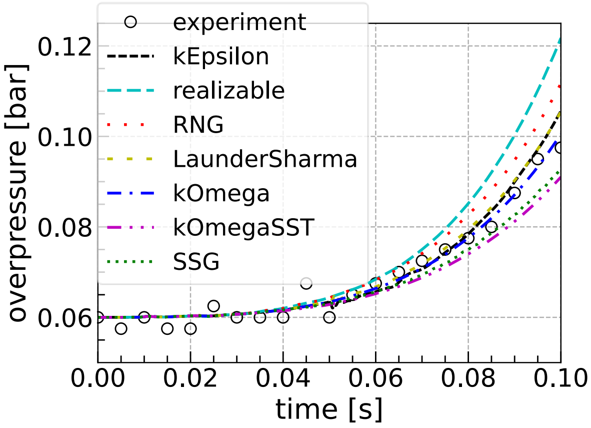
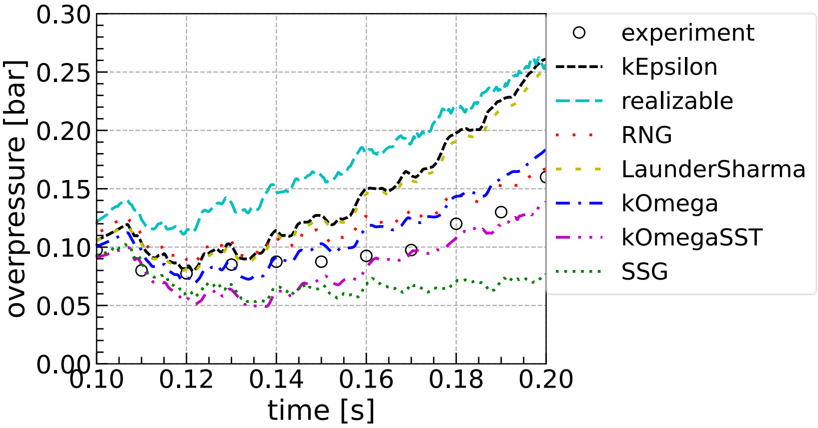


Figure Appendix IV.15 Effect of different turbulence models on the calculated explosion overpressure in a closed vessel. u′=0.75 m/s, L=0.1 m.



(a)



(b)

Figure Appendix IV.16 Effect of different turbulence models on the explosion overpressure computed during (a) the first and (b) second stages of the dust explosion.

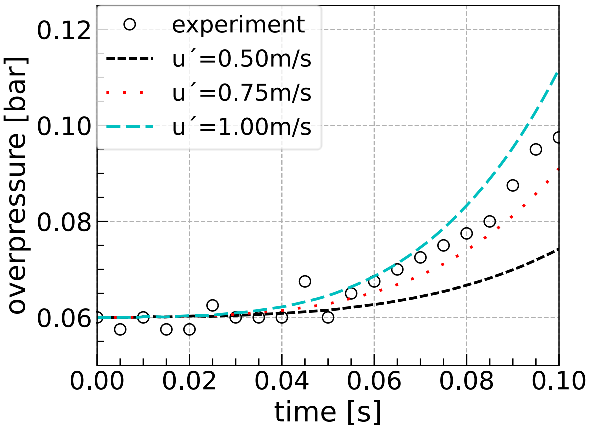


Figure Appendix IV.17 Effect of turbulence velocity fluctuations u' on the calculated explosion overpressure before the rupture of the vent panel. L=0.1 m, k-omega-SST turbulence model.

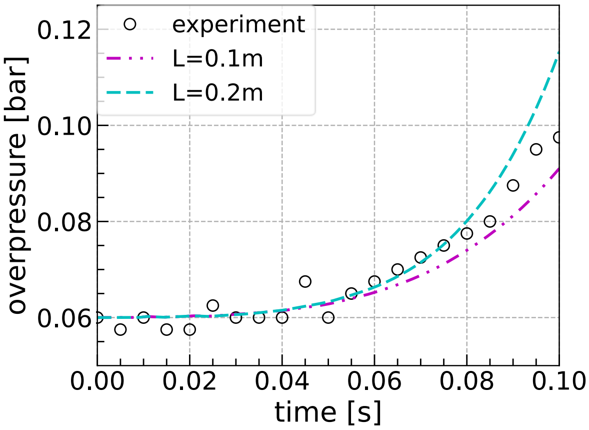


Figure Appendix IV.18 Effect of turbulence length scale on the calculated explosion overpressure before the rupture of the vent panel. u'=0.75 m/s, k-omega-SST turbulence model.

Appendix IV.VII Validation results

As discussed above, a number of input parameters is poorly known under conditions of the simulated experiments, and this is a typical problem for dust explosion modeling. Nevertheless, the experimental data appears to be useful for validating the extended FSC model and developed code. The point is that, within the framework of the FSC model, major dust properties and turbulence characteristics are taken into account in the expression for the intermediately steady turbulent burning velocity, see Eq. (Appendix I.8). Accordingly, the lack of reliable data on one input parameter, e.g., the laminar flame speed S_L for a dust-air mixture, can be offset by tuning another poorly known input parameter, e.g., the initial value of the rms turbulent velocity u'. Such a method was adopted in the present study, with the tuning of u' being done by comparing results measured and computed during the stage (i) only (dust explosion in a closed vessel before rupture of the vent panel). During simulations of the subsequent stages (ii)-(iv), the same input parameters were used. Accordingly, comparison of results measured and computed during stages (ii) and (iii) appears to be a sufficiently solid test, which is difficult to pass. For instance, only two of the seven probed turbulence models were capable for passing this test.

Figure Appendix IV.19 shows that results computed using the conventional FSC model and k-omega turbulence model substantially underestimate the measured overpressure. Figure Appendix IV.21 indicates that this problem can be handled by adopting the extended FSC model, i.e., by substituting Eq. (Appendix I.4) with Eq. (Appendix I.18), which allows for the self-similar flame acceleration discovered by Gostintsev et al. [66]. Such a method requires tuning of a single input parameter, i.e., time t_{flacc} when the acceleration term is acivated in Eq. (Appendix I.18). Recall that

Eq. (Appendix I.4) is used at $t < t_{flacc}$. Since the self-similar regime of flame acceleration was documented for large unconfined flames but the entire explosion kernel is sufficiently small and confined before the rupture of the vent panel at $t = 0.1$ s, the activation time t_{flacc} should be well larger than 0.1 s. The use of a too small t_{flacc} results in overestimated overpressure, see cyan dashed line in Fig. Appendix IV.19.

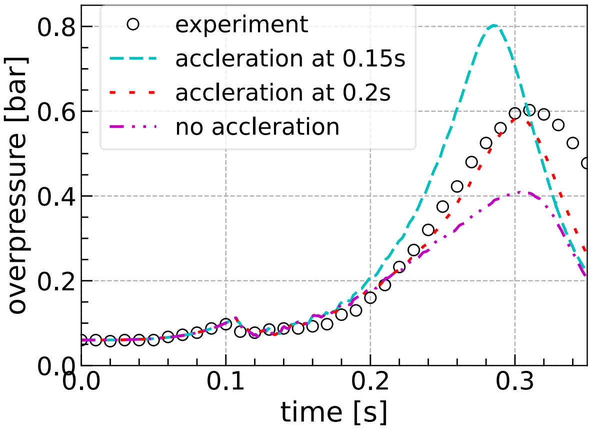


Figure Appendix IV.19 Effect of flame acceleration sub-model on the calculated explosion overpressure. u′=0.75 m/s, L=0.1 m, k-omega turbulence model.

Finally, comparison of red dashed lines with circles in Figs. Appendix IV.19 - Appendix IV.21 shows that the use of the extended FSC model with $t_{flacc} = 0.15$ s and standard k-omega (Fig. Appendix IV.19) or k-omega-SST (Figs. Appendix IV.20 and Appendix IV.21) turbulence model has allowed us to well predict the measured overpressure history during all four studied stages of the dust explosion. In particular, the peak overpressure and the corresponding instants are well predicted. The use of the conventional FSC model without the acceleration factor in Eq. (Appendix I.18) yields significantly underpredicted overpressure when compared to the experimental data, cf. cyan line with symbols in Fig. Appendix IV.21. These results stress importance of the extension of the FSC model, developed within the framework of the present project.

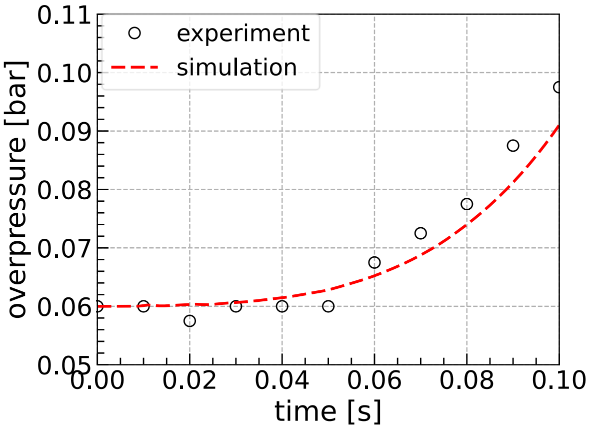


Figure Appendix IV.20 Comparison between measured and simulated explosion overpressures until the rupture of the vent panel.u′=0.75 m/s, L=0.1 m, k-omega-SST turbulence model.

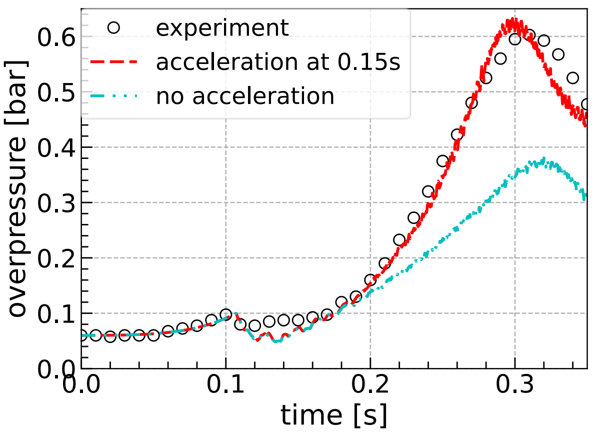


Figure Appendix IV.21 Comparison between measured and simulated explosion overpressures. u′=0.75 m/s, L=0.1 m, k-omega-SST turbulence model.

Appendix V. Calculation of laminar and turbulent viscosities and heat diffusivity

This appendix documents the calculation of laminar and turbulent viscosities and heat diffusivity in OpenFOAM-v 1812.

Kinematic viscosity ν [m2/s]
nu(), laminar kinematic viscosity, src/TurbulenceModels/ turbulenceModels/turbulenceModel.H
nut(), turbulent kinematic viscosity
nuEff(), effective kinematic viscosity, turbulent + laminar

Dynamic viscosity μ [kg/m/s]
mu(), laminar dynamic viscosity, $src/TurbulenceModels/turbulenceModels/turbulenceModel.H
mut(), turbulence dynamic viscosity
muEff(), effective dynamic viscosity, turbulent + laminar

Thermal diffusivity α [kg/m/s]

Laminar thermal diffusivity, thermo.alpha()=thermo.mu()/Pr; see code src/thermophysicalModels/specie/transport/const/constTransportI.H; by changing constant/thermopysicalProperties, reactants or products mu and Pr, you can change the laminar dynamic and thermal viscosity.

alphat(), turbulent thermal diffusivity for enthalpy, src/TurbulenceModels/compressible/EddyDiffusivity/EddyDiffusivity.H

alphaEff() [kg/m/s], effective turbulent thermal diffusivity for enthalpy

alphat_ = this->rho_*this->nut()/Prt_; src/TurbulenceModels/ compressible/EddyDiffusivity/EddyDiffusivity.C

Appendix VI. Case setup for 1-D "frozen" turbulence planar flame

The case structure is as follows

```
0.org  Allclean  Allrun  constant  system
```

In 0.org/ folder, it contains files for setting up initial and boundary conditions

```
alphat  b  epsilon  k  nut  p  Su  T  Tb  Tu  U  Xi
```

Part of the files are shown here in order to save space. In 0.org/alphat file,

```
    object      alphat;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -1 0 0 0 0];


internalField   uniform 0.013224; //for unburned, C_mu*rhou*k2/(Pr_t*epsilon)

boundaryField
{
    left
    {
        type            zeroGradient;
    }
    right
    {
        type            zeroGradient;
    }
    top
    {
        type            cyclic;
    }
    bottom
    {
        type            cyclic;
    }
    front
    {
        type            cyclic;
    }
    back
    {
        type            cyclic;
    }
}
```

In 0.org/b file,

```
    object      b;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];
```

```


internalField   uniform 1;

boundaryField
{
    left
    {
        type            zeroGradient;
    }
    right
    {
        type            fixedValue;
        value           uniform 1;
    }
```

In 0.org/epsilon file,

```
    object      epsilon;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -3 0 0 0 0];


internalField   uniform 11.84;

boundaryField
{
    left
    {
        type            fixedValue;
        value           $internalField;
    }
    right
    {
        type            fixedValue;
        value           $internalField;
    }
```

In 0.org/k file,

```
    object      k;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -2 0 0 0 0];


internalField   uniform 0.96; //0.96;

boundaryField
{
    left
    {
        type            fixedValue;
        value           $internalField;
    }
    right
    {
        type            fixedValue;
        value           $internalField;
    }
```

In 0.org/nut file,

```
    object      nut;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 2 -1 0 0 0 0];


internalField   uniform 0.007;  //C_mu*k2/epsilon

boundaryField
{
    left
    {
        type            zeroGradient;
    }
    right
    {
        type            zeroGradient;
    }
```

In 0.org/p file,

```
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -2 0 0 0 0];


internalField   uniform 110000;

boundaryField
{
    left
    {
        type            totalPressure;
        p0              $internalField;
    }
    right
    {
        type            fixedValue;
        value           $internalField;
    }
```

In 0.org/Su file,

```
    object      Su;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
```

```
dimensions      [0 1 -1 0 0 0 0];


internalField   uniform 0.12;

boundaryField
{
   left
   {
      type          fixedValue;
      value          $internalField;
   }
   right
   {
      type          fixedValue;
      value          $internalField;
   }
```

In 0.org/T file,

```
   object      T;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];


internalField   uniform 328;

boundaryField
{
   left
   {
      type          zeroGradient;
   }
   right
   {
      type          fixedValue;
      value          uniform 328;
   }
```

In 0.org/Tb file,

```
   object      Tb;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];


internalField   uniform 1592;

boundaryField
{
   left
   {
       type            fixedValue;
       value           uniform 1592;
   }
   right
   {
       type            fixedValue;
       value           uniform 1592;
   }
```

In 0.org/Tu file,

```
    object      Tu;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 1 0 0 0];


internalField   uniform 328;

boundaryField
{
    left
    {
        type            fixedValue;
        value           uniform 328;
    }
    right
    {
        type            fixedValue;
        value           uniform 328;
    }
```

In 0.org/U file,

```
    object      U;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 1 -1 0 0 0 0];


internalField   uniform (0 0 0);

boundaryField
{
    left
    {
        type            pressureInletOutletVelocity;
        value           $internalField;
    }
    right
    {
        type            fixedValue;
        value           $internalField;
    }
```

In 0.org/Xi file,

```
   object      Xi;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [0 0 0 0 0 0 0];


internalField   uniform 1;

boundaryField
{
   left
   {
      type           zeroGradient;
   }
   right
   {
      type           zeroGradient;
   }
```

In constant/ folder, it contains files for setting up the model

```
combustionProperties thermophysicalProperties  turbulenceProperties
```

In combustionProperties file

```
   object      combustionProperties;
}
……
fuel           C6H7_88O4_98;  //cornflour

Su             Su [0 1 -1 0 0 0 0] 0.12; //laminar burning velocity of cornflour of phi 0.77

SuModel        unstrained;

equivalenceRatio equivalenceRatio [0 0 0 0 0 0 0] 1.0;
…

XiModel        FSC;
…
// coeffeicient for FSC model, evaluating turbulent length scale
CdCoef CdCoef [0 0 0 0 0 0 0] 0.37;

// coeffeicient for FSC model, evaluating turbulent flame speed
A A [0 0 0 0 0 0 0] 0.5;
…
//smooth ignition parameters
W_0 W_0 [0 0 -1 0 0 0 0] 0;         //[1/s]
t_0 t_0 [0 0 1 0 0 0 0] 0;         //[s] no effect at least a factor of 10 of time step
sigma_r sigma_r [0 1 0 0 0 0 0] 1e-3;  //[m] at least a factor of 2 of mesh size
sigma_t sigma_t [0 0 1 0 0 0 0] 5e-5;  //[s] no effect at least a factor of 2 of time step
```

In thermophysicalProperties file

```
   object      thermophysicalProperties;
}
……
thermoType
{
   type             heheuPsiThermoBML_cornflour;
   mixture         homogeneousMixture;
   transport      const;
   thermo          hConst;
   equationOfState perfectGas;
   specie          specie;
   energy          absoluteEnthalpy;
}

stoichiometricAirFuelMassRatio
   stoichiometricAirFuelMassRatio [0 0 0 0 0 0 0] 4.67;

reactants
{
   specie
   {
       molWeight       32.76; //phi = 0.77 cornflour
   }
   thermodynamics
   {
                   Cp           1007;
                  Hf            0;
   }
   transport
   {
       mu            1.8e-5;
       Pr            0.7;

       As            1.67212e-06;
       Ts            170.672;
   }
}

products
{
   specie
   {
       molWeight       27.15;  //phi = 0.77 cornstarch products
   }
   thermodynamics
   {
                   Cp           1007;
                  Hf            0;
   }
   transport
   {
       mu            4.6e-5;
       Pr            0.7;

       As            1.67212e-06;
       Ts            170.672;
   }
}
```

In turbulenceProperties file

```
   object      turbulenceProperties;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

simulationType  RAS;

RAS
{
   RASModel        kEpsilon;

   kEpsilonCoeffs
   {
      Prt 0.7;
   }
   turbulence      off; //on;

   printCoeffs     on;
}
```

In system/ folder, it contains files for setting up the numerics

```
blockMeshDict  controlDict      fvSchemes      residuals
cBar0D1Dict    Residuals.txt
cBar0D5Dict    flameFrontDict    fvSolution        setFieldsDict
```

In blockMeshDict file,

```
   object      blockMeshDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

scale   0.001;

vertices
(
   (0   0   0)   //0
   (100 0   0)   //1
   (100 3   0)   //2
   (0   3   0)   //3
   (0   0   3)   //4
   (100 0   3)   //5
   (100 3   3)   //6
   (0   3   3)   //7
);

blocks
(
   hex (0 1 2 3 4 5 6 7) (100 3 3) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
   left
   {
       type patch;
       faces
       (
           (0 4 7 3)
       );
   }
   right
   {
       type patch;
       faces
       (
           (2 6 5 1)
       );
   }
   top
   {
       type cyclic;
       neighbourPatch bottom;
       faces
       (
           (4 5 6 7)
       );
   }
   bottom
   {
       type cyclic;
       neighbourPatch top;
       faces
       (
           (0 3 2 1)
       );
   }
   front
   {
       type cyclic;
       neighbourPatch back;
       faces
       (
           (0 1 5 4)
       );
   }
   back
   {
       type cyclic;
       neighbourPatch front;
       faces
       (
           (7 6 2 3)
       );
   }
);
```

```
mergePatchPairs
(
);
```

In fvSchemes file, the second order accuracy numerical scheme are used as recommended in Ref. [92]

```
    object      fvSchemes;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

ddtSchemes
{
    default         backward;
}

gradSchemes
{
    default         cellLimited Gauss linear 1;
    grad(U)         cellLimited Gauss linear 1;
}

divSchemes
{
    default             none;
    div(phi,U)          Gauss linear;  //linearUpwindV grad(U);
    div(phid,p)         Gauss linear;  //limitedLinear 1;
    div(phi,k)          Gauss linearUpwind default;  //limitedLinear 1;
    div(phi,K)          Gauss linear;  //limitedLinear 1;
    div(phi,epsilon)    Gauss linearUpwind default;  //limitedLinear 1;
    div(phi,R)          Gauss linear;  //limitedLinear 1;
    div(R)              Gauss linear;  //linear;
    div(phiXi,Xi)       Gauss linear;  //limitedLinear 1;
    div(phiXi,Su)       Gauss linear;  //limitedLinear 1;
    div(phiSt,b)        Gauss limitedLinear01 1;
    div(phi,ft_b_ha_hau) Gauss multivariateSelection
    {
        fu  limitedLinear01 1;
        ft  limitedLinear01 1;
        b   limitedLinear01 1;
        ha  limitedLinear 1;
        hau limitedLinear 1;
    };
    div(U)              Gauss linear;
    div((Su*n))         Gauss linear;
    div((U+((Su*Xi)*n))) Gauss linear;
    div(((rho*nuEff)*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear limited 1; //corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         limited 1; //orthogonal; //corrected;
}


// ************************************************************************* //
```

In fvSolution file,

```
    object      fvSolution;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

solvers
{
    "(p|rho)"
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-08;
        relTol          0;
        minIter         3;
        maxIter         100;
    }

    "(p|rho)Final"
    {
        $p;
        tolerance       1e-08;
        relTol          0;
    }

    "(U|b|Su|Xi|ha|hau|k|epsilon)"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
        minIter         3;
        maxIter         100;
    }

    "(U|b|Su|Xi|ha|hau|k|epsilon)Final"
    {
        solver          PBiCG;
        preconditioner  DILU;
        tolerance       1e-08;
        relTol          0;
    }
}

PIMPLE  // setup for PISO algorithm
```

```
{
   momentumPredictor yes;
   nOuterCorrectors 1;
   nCorrectors     2;
   nNonOrthogonalCorrectors 1;
   turbOnFinalIterOnly true;

}
```

In residual file, tell OpenFOAM which residuals to be saved

```
#includeEtc "caseDicts/postProcessing/numerical/residuals.cfg"

fields (p U k epsilon b);
```

In setFieldDict file, set left half of the tube as burned

```
   object      setFieldsDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

defaultFieldValues
(
   volScalarFieldValue b 1
   volScalarFieldValue T 328
   volScalarFieldValue alphat 0.013224
);

regions
(
   boxToCell
   {
      box (0 0 0) (0.05 0.003 0.003);
      fieldValues
      (
         volScalarFieldValue b 0
         volScalarFieldValue T 1592
         volScalarFieldValue alphat 0.00225798
      );
   }
);
```

In controlDict file,

```
   object      controlDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

application     FSCFoam_cornflour;

startFrom       startTime;

startTime       0; //2e-3;

stopAt          endTime; //noWriteNow;

endTime         20e-3;//5e-2;

deltaT          1e-06;
```

```
writeControl    timeStep;

writeInterval   2e3;

purgeWrite      0;

writeFormat     ascii;

writePrecision  10;

writeCompression on;

timeFormat      general;

timePrecision   6;

runTimeModifiable true;

adjustTimeStep  no;

maxCo           0.1;

maxDeltaT       1;

functions
{
   #includeFunc  residuals
   sample1
   {
       type       sets;
       libs       ("libsampling.so");
       writeControl writeTime;
       setFormat   raw;
       sets
       (
          line1
          {
              type       uniform;
              axis       distance;

              start      (0   0.0015 0.0015);
              end        (0.1 0.0015 0.0015);
              nPoints     1000;
          }
       );
       interpolationScheme cellPoint;
       fields         (rho c cBar U);
   }
}
```

The script Allrun for running the case is

```
#!/bin/sh
# reset the case
#echo "reset the case"
```

```
#sed -i "s/turbulence      on/turbulence      off/g" constant/turbulenceProperties
#sed -i "s/startTime       2e-4/startTime       0/g" system/controlDict
#sed -i "s/endTime         5e-3/endTime         2e-4/g" system/controlDict
#copy 0 dir
cp -r 0.org/ 0/
wait
echo "blockMesh"
blockMesh > log.blockMesh &
wait
echo "renumberMesh"
#increase the speed of linear solver by renumberMesh utility
renumberMesh -overwrite > log.renumberMesh &
wait
echo "checkMesh"
checkMesh > log.checkMesh &
wait
echo "set field"
setFields > log.setFields &
wait
echo "run "
FSCFoam_cornflour > log.FSCFoam_cornflour &
```

The script Allclean for cleaning the case is

```
rm -rf constant/polyMesh
rm -rf postProcessing
rm -rf VTK
rm log*
foamListTimes -rm
rm -rf 0/
rm -rf dynamicCode
```

Appendix VII. Case setup for 3-D “frozen” turbulent spherical flame

The case structure is as follows

```
0.org  Allclean_parallel  Allrun_parrallel  constant  system
```

In 0.org/ folder, it contains files for setting up initial and boundary conditions

```
alphat  b  epsilon  k  nut  p  Su  T  Tb  Tu  U  Xi
```

Part of the files are shown here in order to save space. Only 0.org/alphat file is shown here, and the rest of files resembles the setup for 1-D planar flame case in Appendix VI.

```
    object      alphat;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

dimensions      [1 -1 -1 0 0 0 0];


internalField   uniform 0.013224; //for unburned, C_mu*rhou*k2/(Pr_t*epsilon)

boundaryField
{
    left
    {
        type            symmetryPlane;
    }
    right
    {
        type            zeroGradient;
    }
    top
    {
        type            zeroGradient;
    }
    bottom
    {
        type            symmetryPlane;
    }
    front
    {
        type            symmetryPlane;
    }
    back
    {
        type            zeroGradient;
    }
}
```

The rest of the setup of files resembles that of 1-D planar turbulent flame case, and will not be reported here. Only system/decomposeParDict is shown here. Note according to Ref. [93], scotch method for decomposing the domain is used.

```
    object      decomposeParDict;
}
```

```
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

numberOfSubdomains 16;

method         scotch;
```

The script Allrun_parallel for running the case in parallel is as follows

```
#!/bin/sh

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions
cp -r 0.org/ 0/
wait
echo "create mesh"
runApplication blockMesh
echo "renumber mesh"
#increase the speed of linear solver by renumberMesh utility
runApplication renumberMesh -overwrite
echo "check mesh"
runApplication checkMesh
echo "set field"
runApplication setFields
echo "decompose"
# Decompose
runApplication decomposePar
echo "run"
# Run
runParallel `getApplication`
echo "reconstruct"
# Reconstruct
runApplication reconstructPar
```

The script Allclean_parallel for cleaning the case in parallel is as follows

```
rm -rf constant/polyMesh
rm -rf processor*
rm -rf postProcessing
rm log*
foamListTimes -rm
rm -rf VTK
rm -rf dynamicCode
rm -rf 0
```

```
scale   0.001;

box 140;
meshSize 0.25;
boxLayer 80;

grading_factor_1 10;//grading factor for near filed
grading_factor_2 6;//grading factor for far field
vertices
(
   (0   0   0)   //0
   ($box 0   0)   //1
   ($box $box   0)   //2
```

```
    (0   $box   0)   //3
    (0   0  $box)   //4
    ($box 0   $box)   //5
    ($box $box  $box)   //6
    (0   $box  $box)   //7
);

blocks
(
hex (0 1 2 3 4 5 6 7) ($boxLayer $boxLayer $boxLayer) simpleGrading (
(
  (0.5 0.875 $grading_factor_1)   //50% of distance and 87.5% of cells
  (0.5 0.125 $grading_factor_2)
)
(
  (0.5 0.875 $grading_factor_1)
  (0.5 0.125 $grading_factor_2)
)
(
  (0.5 0.875 $grading_factor_1)
  (0.5 0.125 $grading_factor_2)
)
)
);
```

Appendix VIII. Implementation of an extra source term in the standard k-epsilon turbulence model

The OpenFOAM implementation of standard k-ε turbulence model involves the following balance equations

$$\frac{\partial \bar{\rho}\tilde{k}}{\partial t} + \nabla.\left(\bar{\rho}\tilde{\mathbf{u}}\tilde{k}\right) = \nabla.\left[\bar{\rho}D_k\nabla\tilde{k}\right] + G_k - \frac{2}{3}\bar{\rho}(\nabla.\tilde{\mathbf{u}})\tilde{k} + S_k - \bar{\rho}\tilde{\varepsilon} \qquad \text{(Appendix VIII.1)}$$

$$\frac{\partial \bar{\rho}\tilde{\varepsilon}}{\partial t} + \nabla.(\bar{\rho}\tilde{\mathbf{u}}\tilde{\varepsilon}) = \nabla.[\bar{\rho}D_\varepsilon\nabla\tilde{\varepsilon}] + \frac{C_1 G_k \tilde{\varepsilon}}{\tilde{k}} - \left(\frac{2}{3}C_1 - C_{3,RDT}\right)\bar{\rho}(\nabla.\tilde{\mathbf{u}})\tilde{\varepsilon} - C_2\bar{\rho}\frac{\tilde{\varepsilon}^2}{\tilde{k}} + S_\varepsilon \qquad \text{(Appendix VIII.2)}$$

To simulate flames expanding from the centre of a fan-stirred bomb, an extra source term $\bar{\rho}\varepsilon_0$ is added to mimic the flux of turbulent energy from the fans to the centre of the vessel [75]

$$\frac{\partial \bar{\rho}\tilde{k}}{\partial t} + \nabla.\left(\bar{\rho}\tilde{\mathbf{u}}\tilde{k}\right) = \nabla.\left[\bar{\rho}D_k\nabla\tilde{k}\right] + G_k - \frac{2}{3}\bar{\rho}(\nabla.\tilde{\mathbf{u}})\tilde{k} + S_k - \bar{\rho}\tilde{\varepsilon} + \bar{\rho}\varepsilon_0 \qquad \text{(Appendix VIII.3)}$$

$$\frac{\partial \bar{\rho}\tilde{\varepsilon}}{\partial t} + \nabla.(\bar{\rho}\tilde{\mathbf{u}}\tilde{\varepsilon}) = \nabla.[\bar{\rho}D_\varepsilon\nabla\tilde{\varepsilon}] + \frac{C_1 G_k \tilde{\varepsilon}}{\tilde{k}} - \left(\frac{2}{3}C_1 - C_{3,RDT}\right)\bar{\rho}(\nabla.\tilde{\mathbf{u}})\tilde{\varepsilon} - C_2\frac{\tilde{\varepsilon}}{\tilde{k}}(\bar{\rho}\tilde{\varepsilon} - \bar{\rho}\varepsilon_0) + S_\varepsilon \qquad \text{(Appendix VIII.4)}$$

Since the TurbulenceModels is a templated class (due to the first capital letter), the implementation of new turbulence model is different from the traditional way. A file, makeTurbModel.C, is created in $WM_PROJECT_USER_DIR/src/TurbulenceModels/turbulenceModels/ with the contents as follows

```
#include "CompressibleTurbulenceModel.H"
#include "compressibleTransportModel.H"
#include "fluidThermo.H"
#include "addToRunTimeSelectionTable.H"
#include "makeTurbulenceModel.H"

#include "ThermalDiffusivity.H"
#include "EddyDiffusivity.H"

#include "laminarModel.H"
#include "RASModel.H"
#include "LESModel.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
#define createBaseTurbulenceModel(                                        \
Alpha, Rho, baseModel, BaseModel, TDModel, Transport)                \
\
namespace Foam                                                           \
{                                                                        \
typedef TDModel<BaseModel<Transport>>                              \
Transport##BaseModel;                                              \
typedef RASModel<EddyDiffusivity<Transport##BaseModel>>    \
RAS##Transport##BaseModel;                                          \
typedef LESModel<EddyDiffusivity<Transport##BaseModel>>     \
LES##Transport##BaseModel;                                          \
}

createBaseTurbulenceModel
(
geometricOneField,
volScalarField,
compressibleTurbulenceModel,
CompressibleTurbulenceModel,
ThermalDiffusivity,
fluidThermo
);

#define makeRASModel(Type)                                          \
makeTemplatedTurbulenceModel                                    \
(fluidThermoCompressibleTurbulenceModel, RAS, Type)

#define makeLESModel(Type)                                          \
makeTemplatedTurbulenceModel                                    \
(fluidThermoCompressibleTurbulenceModel, LES, Type)

#include "mykEpsilon.H"
makeRASModel(mykEpsilon);
```

Take a copy of standard k-epsilon turbulence model into the directory $WM_PROJECT_USER_DIR/src/TurbulenceModels/turbulenceModels/RAS/mykEpsilon. Rename the files from kEpsilon to mykEpsilon. Make the changes in the mykEpsilon.C file for including the source terms as follows

```
……
//read in epsilon0_ from turbulenceProperties dictionary
IOdictionary turbulenceProperties
(
IOobject
(
"turbulenceProperties",
this->runTime_.constant(),
this->mesh_,
IOobject::MUST_READ_IF_MODIFIED,
IOobject::NO_WRITE
)
);
dimensionedScalar epsilon0_("epsilon0_", dimensionSet(0,2,-3,0,0,0,0), turbulenceProperties);
// calculate epsilon0 for including extra source term
volScalarField epsilon0=epsilon_-epsilon0_;
……
// Dissipation equation
tmp<fvScalarMatrix> epsEqn
(
fvm::ddt(alpha, rho, epsilon_)
+ fvm::div(alphaRhoPhi, epsilon_)
fvm::laplacian(alpha*rho*DepsilonEff(), epsilon_)
==
C1_*alpha()*rho()*G*epsilon_()/k_()
fvm::SuSp(((2.0/3.0)*C1_ - C3_)*alpha()*rho()*divU, epsilon_)
//      - fvm::Sp(C2_*alpha()*rho()*epsilon_()/k_(), epsilon_)
fvm::SuSp(C2_*alpha()*rho()*epsilon0/k_(), epsilon_)//extra source term
+ epsilonSource()
+ fvOptions(alpha, rho, epsilon_)
);
……
// Turbulent kinetic energy equation
tmp<fvScalarMatrix> kEqn
(
fvm::ddt(alpha, rho, k_)
+ fvm::div(alphaRhoPhi, k_)
fvm::laplacian(alpha*rho*DkEff(), k_)
==
alpha()*rho()*G
fvm::SuSp((2.0/3.0)*alpha()*rho()*divU, k_)
//      - fvm::Sp(alpha()*rho()*epsilon_()/k_(), k_)
fvm::SuSp(alpha()*rho()*epsilon0()/k_(), k_)  //extra source term
+ kSource()
+ fvOptions(alpha, rho, k_)
);
```

The Make/files and Make/options files located in $WM_RPOJECT_USER_DIR/TurbulenceModels/turbulenceModels are shown as follows

```
makeTurbModel.C

LIB = $(FOAM_USER_LIBBIN)/libmyTurbulenceModels
```

```
EXE_INC = \
-I$(LIB_SRC)/TurbulenceModels/compressible/lnInclude \
-I$(LIB_SRC)/TurbulenceModels/turbulenceModels/lnInclude \
-I$(LIB_SRC)/transportModels/compressible/lnInclude \
-I$(LIB_SRC)/thermophysicalModels/basic/lnInclude \
-I$(LIB_SRC)/thermophysicalModels/specie/lnInclude \
-I$(LIB_SRC)/thermophysicalModels/solidThermo/lnInclude \
-I$(LIB_SRC)/thermophysicalModels/solidSpecie/lnInclude \
-I$(LIB_SRC)/finiteVolume/lnInclude \
-I$(LIB_SRC)/meshTools/lnInclude

LIB_LIBS = \
-lcompressibleTurbulenceModels \
-lcompressibleTransportModels \
-lfluidThermophysicalModels \
-lsolidThermo \
-lsolidSpecie \
-lturbulenceModels \
-lspecie \
-lfiniteVolume \
-lmeshTools
```

Appendix IX. Mesh generation using snappyHexMesh in OpenFOAM

SnappyHexMesh is a utility in OpenFOAM and it creates mesh using stl file in ASCII format in three steps: castellated mesh, snapping mesh and layer addition. It includes more than 70 parameters in a text-based file for adjusting the mesh quality. Detailed explanations of the usage of different parameters and the examples are reported online [94, 95]. In this section, a step-by-step description of mesh generation using snappyHexMesh for this specific case is presented.

First, a background mesh is created before using snappyHexMesh utility. The background mesh is usually produced using blockMesh utility in OpenFOAM. The important part of the blockMeshDict script of the current Rembe vessel background mesh is shown below with a domain size of 15.5×5×6.355 m and a grid size of 0.1 m.

```
scale   1; //unit in metre
//define parameters for fast change of mesh
x_min -0.5;  //min x-coord
x_max 15.0; //max x-coord
y_min 0.0;    //min y-coord
y_max 5.0;    //max y-coord
z_min -1.355;//min z-coord
z_max 5;        //max z-coord

x_layer 155;   //num of layers in x-coord
y_layer 50;     //num of layers in y-coord
z_layer 64;      //num of layers in z-coord

vertices
(
   ($x_min   $y_min   $z_min)   //0
   ($x_max   $y_min   $z_min)   //1
   ($x_max   $y_max   $z_min)   //2
   ($x_min   $y_max   $z_min)   //3
   ($x_min   $y_min   $z_max)   //4
   ($x_max   $y_min   $z_max)   //5
   ($x_max   $y_max   $z_max)   //6
   ($x_min   $y_max   $z_max)   //7
);

blocks
(
hex (0 1 2 3 4 5 6 7) ($x_layer $y_layer $z_layer) simpleGrading (1 1 1)
);

edges
(
);

boundary
(
   left
   {
      type patch;
```

```
        faces
        (
            (0 4 7 3)
        );
    }
    right
    {
        type patch;
        faces
        (
            (2 6 5 1)
        );
    }
    top
    {
        type patch;
        faces
        (
            (4 5 6 7)
        );
    }
    bottom
    {
        type wall;
        faces
        (
            (0 3 2 1)
        );
    }
    front
    {
        type symmetryPlane;  //symmetric plane
        faces
        (
            (0 1 5 4)
        );
    }
    back
    {
        type patch;
        faces
        (
            (7 6 2 3)
        );
    }
);

mergePatchPairs
(
);
```

Second, prepare the geometry file. Make sure the stl geometry file is saved in ASCII format, and use the unit in meter, otherwise problems will occur when using snappyHexMesh utility. The stl file can be converted from mm to m using the command surfaceTransformPoints -scale “(0.001 0.001 0.001)” sourceStlFile.stl targetStlFile.stl.

Third, place the stl file in the folder of $case/constant/triSurface/. More advanced method of extracting special edge features is available online, e.g. https://www.youtube.com/watch?v=Fvq-UfSVz0M, and https://www.youtube.com/watch?v=ObsFQUiVi1U&t=2569s.

Fourth, extract surface feature using command surfaceFeatureExtract.

Fifth, produce castellated mesh using snappyHexMesh and activating only castellated mesh in the system/snappyHexDict.

Sixth, produce snapped mesh using snappyHexMesh by activating only snappy.

Seventh, produce layered mesh using snappyHexMesh by activating only addlayer. It is worth noting that most of the case, addlayer feature causes unsatisfactory mesh.

The important part of the snappyHexMeshDict file of this case is shown below with comments for understanding the script.

```
// Which of the steps to run
castellatedMesh false;    //switch to true in step 5
snap            false;    //switch to true in step 6
addLayers        true;    //switch to true in step 7

// Geometry. Definition of all surfaces. All surfaces are of class
// searchableSurface.
// Surfaces are used
// - to specify refinement for any mesh cell intersecting it
// - to specify refinement for any mesh cell inside/outside/near
// - to 'snap' the mesh boundary to the surface
geometry
{
  rembe_vessel_circle_part1_meter.stl   //the shell of vessel main part
  {
    type triSurfaceMesh;
    name rembe_vessel_circle_part1_meter;
  }
  rembe_vessel_circle_part2_meter.stl   //the vessel vent wall
  {
    type triSurfaceMesh;
    name rembe_vessel_circle_part2_meter;
  }
  rembe_vessel_circle_inside_meter.stl   //the inside of the vessel
  {
    type triSurfaceMesh;
    name rembe_vessel_circle_inside_meter;
  }
  outer_kernel   //geometrical entity for ignition kernel
  {
    type searchableSphere;
    centre  (2.461 0 0);
    radius  0.7;
  }
  vent_near_field   //geometrical entity for near field volume of venting
  {
    type searchableCylinder;
    point1  (3.74 0 0);
    point2  (4.74 0 0);
    radius  1.1;  //was 1.85
  }
```

```
    vent_far_field   //geometrical entity for far field volume of venting
    {
        type searchableCylinder;
        point1  (4.74 0 0);
        point2  (15 0 0);
        radius  1.85;
    }
}

// Settings for the castellatedMesh generation.
castellatedMeshControls
{

    // Refinement parameters
    // ~~~~~~~~~~~~~~~~~~~~~~

    // If local number of cells is >= maxLocalCells on any processor
    // switches from from refinement followed by balancing
    // (current method) to (weighted) balancing before refinement.
    maxLocalCells 1000000; //was 100000

    // Overall cell limit (approximately). Refinement will stop immediately
    // upon reaching this number so a refinement level might not complete.
    // Note that this is the number of cells before removing the part which
    // is not 'visible' from the keepPoint. The final number of cells might
    // actually be a lot less.
    maxGlobalCells  8000000;//was 2000000

    // The surface refinement loop might spend lots of iterations refining just a
    // few cells. This setting will cause refinement to stop if <= minimumRefine
    // are selected for refinement. Note: it will at least do one iteration
    // (unless the number of cells to refine is 0)
    minRefinementCells 0;  //was 10

    // Allow a certain level of imbalance during refining
    // (since balancing is quite expensive)
    // Expressed as fraction of perfect balance (= overall number of cells /
    // nProcs). 0=balance always.
    maxLoadUnbalance 0.10;

    // Number of buffer layers between different levels.
    // 1 means normal 2:1 refinement restriction, larger means slower
    // refinement.
    nCellsBetweenLevels 3; //was 3

    // Explicit feature edge refinement
    // ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    // Specifies a level for any cell intersected by its edges.
    // This is a featureEdgeMesh, read from constant/triSurface for now.

    features
    (
        {
            file "rembe_vessel_circle_part1_meter.eMesh";  //generated by running
```

```
surfaceFeatureExtract
            level 2; //was 3
        }
        {
            file "rembe_vessel_circle_part2_meter.eMesh";
            level 3; //was 3
        }
    );

    // Surface based refinement
    // ~~~~~~~~~~~~~~~~~~~~~~~~

    // Specifies two levels for every surface. The first is the minimum level,
    // every cell intersecting a surface gets refined up to the minimum level.
    // The second level is the maximum level. Cells that 'see' multiple
    // intersections where the intersections make an
    // angle > resolveFeatureAngle get refined up to the maximum level.

    refinementSurfaces
    {
        rembe_vessel_circle_part1_meter
        {
            // Surface-wise min and max refinement level
            level (3 3); // 3 3
        }

        rembe_vessel_circle_part2_meter
        {
            // Surface-wise min and max refinement level
            level (3 3); // 4 4
        }
    }

    // Resolve sharp angles
    resolveFeatureAngle 10;  // this should be 180 -  includedAngle in surfaceFeatureExtracDict file

    // Region-wise refinement
    // ~~~~~~~~~~~~~~~~~~~~~~

    // Specifies refinement level for cells in relation to a surface. One of
    // three modes
    // - distance. 'levels' specifies per distance to the surface the
    //   wanted refinement level. The distances need to be specified in
    //   descending order.
    // - inside. 'levels' is only one entry and only the level is used. All
    //   cells inside the surface get refined up to the level. The surface
    //   needs to be closed for this to be possible.
    // - outside. Same but cells outside.

    refinementRegions
    {
        rembe_vessel_circle_inside_meter
        {
            mode inside;
            levels ((2 2)); // 3 3
```

```
        }
        outer_kernel
        {
            mode inside;
            levels ((3 3)); // 4 4
        }
        vent_near_field
        {
            mode inside;
            levels ((2 2)); // 4 4
        }
        vent_far_field
        {
            mode inside;
            levels ((1 1)); // 4 4
        }
    }

    // Mesh selection
    // ~~~~~~~~~~~~~~
    // After refinement patches get added for all refinementSurfaces and
    // all cells intersecting the surfaces get put into these patches. The
    // section reachable from the locationInMesh is kept.
    // NOTE: This point should never be on a face, always inside a cell, even
    // after refinement.
    locationInMesh (2.461 1e-2 1e-2);

    // Whether any faceZones (as specified in the refinementSurfaces)
    // are only on the boundary of corresponding cellZones or also allow
    // free-standing zone faces. Not used if there are no faceZones.
    allowFreeStandingZoneFaces true;
}

// Settings for the snapping.
snapControls
{
    //- Number of patch smoothing iterations before finding correspondence
    //  to surface
    nSmoothPatch 3;  //was 3

    //- Relative distance for points to be attracted by surface feature point
    //  or edge. True distance is this factor times local
    //  maximum edge length.
    tolerance 2; // was 2.0

    //- Number of mesh displacement relaxation iterations.
    nSolveIter 100; //was 30 -100

    //- Maximum number of snapping relaxation iterations. Should stop
    //  before upon reaching a correct mesh.
    nRelaxIter 5; //was 5

    // Feature snapping
```

```
        //- Number of feature edge snapping iterations.
        //  Leave out altogether to disable.
        nFeatureSnapIter 10;  //was 10, affect compuational time greatly

        //- Detect (geometric only) features by sampling the surface
        //  (default=false).
        implicitFeatureSnap false;

        //- Use castellatedMeshControls::features (default = true)
        explicitFeatureSnap true;

        //- Detect points on multiple surfaces (only for explicitFeatureSnap)
        multiRegionFeatureSnap false; // was false
}

// Settings for the layer addition.
addLayersControls
{
    // Are the thickness parameters below relative to the undistorted
    // size of the refined cell outside layer (true) or absolute sizes (false).
    relativeSizes true;

    // Per final patch (so not geometry!) the layer information
    layers
    {
        rembe_vessel_circle_part2_meter
        {
            nSurfaceLayers 5; //was 3
        }
    }

    // Expansion factor for layer mesh
    expansionRatio 1.1; //was 1.1

    // Wanted thickness of final added cell layer. If multiple layers
    // is the thickness of the layer furthest away from the wall.
    // Relative to undistorted size of cell outside layer.
    // See relativeS2izes parameter.
    finalLayerThickness 0.5; //was 0.5

    // Minimum thickness of cell layer. If for any reason layer
    // cannot be above minThickness do not add layer.
    // Relative to undistorted size of cell outside layer.
    minThickness 0.1; //was 0.1

    // If points get not extruded do nGrow layers of connected faces that are
    // also not grown. This helps convergence of the layer addition process
    // close to features.
    // Note: changed(corrected) w.r.t 1.7.x! (didn't do anything in 1.7.x)
    nGrow 0;

    // Advanced settings

    // When not to extrude surface. 0 is flat surface, 90 is when two faces
    // are perpendicular
    featureAngle 360; //was 60;

    // At non-patched sides allow mesh to slip if extrusion direction makes
    // angle larger than slipFeatureAngle.
    slipFeatureAngle 180; // was 30; one half of featureAngle

    // Maximum number of snapping relaxation iterations. Should stop
    // before upon reaching a correct mesh.
    nRelaxIter 5;  //was 3, typically 5

    // Number of smoothing iterations of surface normals
    nSmoothSurfaceNormals 1;  //was 1

    // Number of smoothing iterations of interior mesh movement direction
    nSmoothNormals 3; //was 3

    // Smooth layer thickness over surface patches
    nSmoothThickness 100; //was 10

    // Stop layer growth on highly warped cells
    maxFaceThicknessRatio 0.5;  //0.5

    // Reduce layer growth where ratio thickness to medial
    // distance is large
    maxThicknessToMedialRatio 0.3; //0.3

    // Angle used to pick up medial axis points
    // Note: changed(corrected) w.r.t 1.7.x! 90 degrees corresponds to 130
    // in 1.7.x.
    minMedialAxisAngle 90;

    // Create buffer region for new layer terminations
    nBufferCellsNoExtrude 0;

    // Overall max number of layer addition iterations. The mesher will exit
    // if it reaches this number of iterations; possibly with an illegal
    // mesh.
    nLayerIter 50; //was 50
}
```

Besides snappyHexMeshDict file, a dictionary file called surfaceFeatureExtractDict should be created as well. See the important part of the file below

```
rembe_vessel_circle_part1_meter.stl
{
    // How to obtain raw features (extractFromFile || extractFromSurface)
    extractionMethod    extractFromSurface;

    // Mark edges whose adjacent surface normals are at an angle less
    // than includedAngle as features
    // - 0  : selects no edges
    // - 180: selects all edges
    includedAngle       170;  //this should be 180 - resolveFeatureAngle

    subsetFeatures
    {
        // Keep nonManifold edges (edges with >2 connected faces)
        nonManifoldEdges       no;

        // Keep open edges (edges with 1 connected face)
        openEdges       yes;
    }



    // Write options

    // Write features to obj format for postprocessing
    writeObj           yes;
}
```

Appendix X.Map results between different geometries in OpenFOAM

OpenFOAM allows mapping results between consistent and inconsistent geometries. First, place a mapFieldsDict in the system directory, which reads as follows

```
/*--------------------------------*- C++ -*----------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  v1812                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.com                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      mapFieldsDict;
}

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

// List of pairs of source/target patches for mapping
patchMap
(
);

// List of target patches cutting the source domain (these need to be
// handled specially e.g. interpolated from internal values)
cuttingPatches
(
);

// ************************************************************************* //
```

Create a folder with name of the instance when the new simulation will start, e.g. 0.08

Run the following command

```
mapFields ../source_case -sourceTime 0.08
```

In contrast, if the fields are consistent, use the flag -consistent.

Appendix XI. Set initial field using equations by funkySetFields

funkySetFields is a utility in swak4Foam for creating complicated initial field using mathematical expressions. You need to download and compile the swak4Foam package following the link

https://openfoamwiki.net/index.php/Contrib/swak4Foam

An example of using funkySetFields for setting the initial Favre-averaged regress variable following an approximation of complementary error function.

Since we are setting two different expressions depending on the domain located inside or outside of the mean flame radius, two funkySetFieldsDict files are needed and the funkySetFields application should be run twice, i.e. funkySetFields -time 0. Example of a funkySetFieldsDict file is as follows

```
FoamFile
{
   version     2.0;
   format      ascii;
   class       dictionary;
   location    "system";
   object      funkySetFieldsDict;
}
//#include "./constant/combustionProperties"
defaultVariables
(
   "kernel_radius=0.52;" //ignition kernel radius
   "kernel_thickness=0.065;" //ignition kernel flame thickness, approximately 25% of kernel radius
   "ign_loc_x=2.461;"    //ignition location x
   "ign_loc_y=0.0;"      //ignition location y
   "ign_loc_z=0.0;"      //ignition location z
   "radius=pow(pow(pos().x-ign_loc_x,2)+pow(pos().y-ign_loc_y,2)+pow(pos().z-ign_loc_z,2),0.5);"//radius
   "zeta=pow(pi,0.5)*(radius-kernel_radius)/kernel_thickness;" //zeta in estimating complementary error function
   "xi=zeta/pow(pi,0.5);"  //xi
   "zn=1.0;"    //when zeta >=0 which yields zn=1.0
   "zz=1./(1.+0.47047*zeta*zn);"
   "f=0.5*(1.0+zn*(1.0-(0.3480242*zz-0.0958709*zz*zz+0.7478556*zz*zz*zz)*exp(-zeta*zeta)));" //bBar
   "sigma=6.37;"    //density ratio
   "b=sigma*f/((1.0-f)*(1.0-sigma)+sigma);"    //Favre averaged b
);
expressions
(
   setB
   {
      field b;
      variables $defaultVariables;
                    expression "b";
                    condition "radius-kernel_radius >=0";
    }
);
```