Chapitre 9. OUTILS DE SIMULATIONS POUR L'INCENDIE

Arnaud Trouvé

Department of Fire Protection Engineering

University of Maryland, College Park 20742 (USA)

ESIA – Ecole des Sciences de l'Incendie et Applications – Obernai, 27 mai au 1er juin 2018

Workshop on CFD-Based Fire Modeling

Objectives

The purpose of this workshop is to:

- Provide an introduction to fire modeling based on the Computational Fluid Dynamics (CFD) approach
- Provide an introduction to the *Fire Dynamics Simulator* (FDS) developed by the US National Institute of Standards and Technology (NIST)
    - The official FDS website can be found at *https://pages.nist.gov/fds-smv/*
- Take a first-time user approach to FDS
    - Walk through the different steps involved in setting up and running FDS
- Simulate with FDS a series of simple test problems in order to illustrate the performance and functionalities of the software and also discuss some of the main technical difficulties encountered at the user level
    - Study the effects of computational grid resolution (for both the gas phase solver and the solid phase solver) and of angular resolution (for the radiation solver)

Additional reading material

- McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Weinschenk, C., Overholt, K., "*Fire Dynamics Simulator – User's Guide*," NIST Special Publication 1019, Sixth Ed., National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.

- McGrattan, K., Hostikka, S., McDermott, R., Floyd, J., Vanella, M., Weinschenk, C., Overholt, K., "*Fire Dynamics Simulator (Version 6) – Technical Reference Guide, Volume 1: Mathematical Model; Volume 2: Verification; Volume 3: Validation; Volume 4: Configuration Management*" NIST Special Publication 1018, Sixth Ed., National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017.
    - The FDS documentation is located in the *FDS/FDS6/Documentation/* folder of the local directory where FDS is installed and is also available at *https://pages.nist.gov/fds-smv/*

Additional FDS examples

- FDS provides an extensive library of example cases that serve to illustrate its capabilities and can be conveniently used to become a proficient user of the software
    - The FDS library is located in the *FDS/FDS6/Examples/* folder of the local directory where FDS is installed

1 Introduction of FDS: *FirsTry.fds*

The Fire Dynamics Simulator (FDS) is an advanced CFD-based fire model developed by the National Institute of Standards and Technology and made available as free, open-source software on the GitHub website (*https://github.com/firemodels/fds*). We assume here that FDS has been successfully downloaded and installed on a local computer. We propose in this section a discussion

of an example case called *FirstTry*. The FDS input file is an ASCII text file called *FirstTry.fds* (this file will be provided at the Workshop). We review below the main features of *FirstTry.fds*.

1.1 Computational Mesh

&MESH IJK=50,40,30, XB=0.0,5.0,0.0,4.0,0.0,3.0 /

This statement produces a uniform computational mesh with cubic-shaped cells of 10-cm size, $\Delta x = \Delta y = \Delta z = 0.1$ m. The values of Δx, Δy and Δz define the computational mesh resolution used in the *FirstTry* simulation, along the *x*-, *y*- and *z*-directions respectively.

The choices made for the computational mesh (also called computational grid) are critical to the quality and accuracy of a simulation. They are also directly controlling the computational cost. Fine-grained simulations are simulations that use a large number of computational cells; fine-grained simulations thereby provide high spatial resolution at the cost of long computational times. In contrast, coarse-grained simulations are simulations that use a small-to-moderate number of computational cells; coarse-grained simulations provide low spatial resolution with the benefit of short computational times.

While guidelines do exist to assist the user in making suitable grid design choices, the choices are problem-dependent and guidelines tend to be difficult to generalize. As a matter of fact, CFD practitioners often work backward starting from an estimate of the time left before a given deadline and then designing a computational grid that will lead to computational times that are consistent with the available computing power and the project schedule. While this is an approach that makes practical sense, it is important for CFD users to understand how close or how far their grid resolution choices might be from a suitably-designed computational grid. In fact, CFD projects are always invited (and often required) to provide both a discussion of grid quality and a justification of the choices that were made for grid resolution.

Grid design may be understood as a process based on a comparison of length scales. It starts first with the identification of relevant physical length scales, noted L, in a given fire scenario and then finishes with a comparison of those length scales to the computational mesh resolution, as measured by the values of Δx, Δy and Δz. The rule of thumb for grid design consists in providing 10 grid cells per relevant physical length scale: $(L/\Delta x) \geq 10$, $(L/\Delta y) \geq 10$, $(L/\Delta z) \geq 10$. The idea is that if this criterion is not met, the physics associated with L will not be correctly captured by the simulation, which will lead in turn to a loss of accuracy. In this argument, L is a generic notation that designates characteristic length scales associated with flame features, flow features, compartment vents, objects, *etc*. For instance, the list of possibly-relevant physical length scales for any fire scenario includes: the flame length L_f, the fire plume thickness δ_{PL}, the smoke layer depth (H-z_D), the thicknesses of wall boundary layers δ_{BL}, the dimensions of vents, H_O and W_O, the dimensions of the fuel source(s), D, *etc*.

The *FirstTry* example corresponds to a simple compartment fire scenario featuring a prescribed fuel source, a flame and a doorway. We now review the ability of the selected computational grid to resolve these respective features. The fuel source is a floor-level square-shaped vent with a size $D = 0.6$ m: we have $(D/\Delta x) = (0.6/0.1) = 6$ and $(D/\Delta y) = (0.6/0.1) = 6$, which means that the spatial resolution of the vent flow is only marginal ($(D/\Delta x)$ and $(D/\Delta y)$ are smaller than 10). Next, we turn to the flame zone. The flame length may be estimated using the classical correlation due to Heskestad: $L_f = (0.235 \times \dot{Q}^{2/5} - 1.02 \times D_f)$, where $\dot{Q}$ is the heat release rate of the fire (in units of kW) and D_f the effective diameter of the fuel source. We use $\dot{Q} = 200$ kW, $D_f = (4 \times (0.6 \times 0.6)/\pi)^{0.5} \approx 0.68$ m and get $L_f \approx 1.26$ m; we then have $\left(L_f/\Delta z\right) = (1.26/0.1) = 12.6$, which means that the spatial resolution of the flame is adequate ($\left(L_f/\Delta z\right)$ is greater than 10), at least in the vertical direction (the flame thickness in the horizontal plane is of order D and the

spatial *x*- and *y*-resolution of the flame is only marginal). Finally, we consider the doorway vent. The door features a height H_O = 2 m and a width W_O = 0.8 m: we have $(H_O/\Delta z) = (2/0.1) = 20$ and $(W_O/\Delta z) = (0.8/0.1) = 8$, which means that the spatial resolution of the vent flow is good in the vertical *z*-direction but only marginal in the *y*-direction.

Overall, the computational grid selected in the *FirstTry* simulation corresponds to an acceptable choice, provided that the simulation is aimed at extracting global features from the fire scenario, for instance the averaged outflow rate of smoke or the bulk smoke layer temperature. The computational grid would probably not be acceptable (and would have to be refined) if more detailed information was needed, for instance heat fluxes on a flammable object located inside the compartment and that may experience secondary ignition. This discussion suggests that the evaluation of the quality of a computational grid used in a given simulation is also related to the objectives of that particular simulation.

Note also that the estimates of the quality of the grid presented above should be considered as good starting points and that these starting points need to be typically supported by what is called a grid convergence study. A grid convergence study consists in simulating the same problem with increasingly refined computational grids (typically by a factor of at least 4 in each direction) and showing that beyond a certain resolution level, the solution becomes independent of changes in the grid. While highly desirable, the task of performing a grid convergence study is generally difficult and computationally expensive: when one increases the number of grid cells by a factor 2 in each direction, the computational cost is increased by a factor of approximately 16 (see Section 1.2 below). Thus, a grid convergence study that considers a first-level grid with given choices of $(\Delta x, \Delta y, \Delta z)$, a second-level grid with $(\Delta x/2, \Delta y/2, \Delta z/2)$, and a third-level grid with $(\Delta x/4, \Delta y/4, \Delta z/4)$ will include 3 simulations and the third-level grid simulation will be approximately 256 times more expensive that the first-level grid simulation.

1.2 Computational Cost

The computational cost of a simulation scales approximately linearly with the total number of grid cells, noted N_{GC}, $N_{GC} = (N_{\Delta x} \times N_{\Delta y} \times N_{\Delta z})$, where $N_{\Delta x}$ ($N_{\Delta y}$, $N_{\Delta z}$) is the number of cells in the *x*- (*y*-, *z*-) direction. The computational cost also scales approximately linearly with the total number of steps, noted $N_{\Delta t}$, used to advance the numerical solution in time. Because of numerical stability constraints, temporal resolution is directly controlled by spatial resolution: a grid that is twice finer ($N_{\Delta x}$, $N_{\Delta y}$ and $N_{\Delta z}$ are multiplied by a factor 2) will require time steps that are twice smaller ($N_{\Delta t}$ is also multiplied by a factor 2).

We can write in a loose fashion: $\text{CPU cost} = C \times (N_{\Delta x} \times N_{\Delta y} \times N_{\Delta z}) \times N_{\Delta t}$, where C is a computer-machine-dependent coefficient. When one increases the number of grid cells by a factor 2 in each direction, N_{GC} is increased by a factor 8; $N_{\Delta t}$ is increased by a factor 2; and the computational cost is increased by a factor of (approximately) 16.

1.3 Prescribed Fuel Mass Loss Rate

```
&OBST XB=2.2,2.8,1.7,2.3,0.0,0.1,
SURF_IDS='BURNER','INERT','INERT' /
&SURF ID='BURNER', HRRPUA = 555.55, RAMP_Q='fireramp' /
&RAMP ID='fireramp', T= 0.0,  F=0.0  /
&RAMP ID='fireramp', T= 10.0, F=0.026 /
…
```

&RAMP ID='fireramp', T=440.0, F=0.004 /

&RAMP ID='fireramp', T=450.0, F=0.000 /

&RAMP ID='fireramp', T=600.0, F=0.0 /

In the *FirstTry* example, the fuel mass loss rate is user-prescribed using the HRRPUA statement. We now present a brief description of the FDS set-up. The fuel source is a floor-level square-shaped vent with a surface area $A_f = (0.6 \times 0.6)$ m^2. The fuel is methane. The combustion model in FDS uses a global combustion equation and the heat of combustion per unit mass of fuel is $\Delta H_f = 49.6746591$ MJ/kg (this value can be found in the FDS output file called *FirstTry.out*; open this file and look for the METHANE keyword). By definition, we have: HRRPUA $= \dot{m}_f'' \times \Delta H_f$, where HRRPUA is the "Heat Release Rate Per Unit Area" (in units of kW/m^2) and $\dot{m}_f''$ the fuel mass loss rate per unit fuel source area (in units of kg/s/m^2); this expression shows that the HRRPUA statement in the FDS input file is simply a way to prescribe the fuel mass loss rate. We can write the fuel mass loss rate (integrated across the fuel source area) as: $\dot{m}_f = \dot{m}_f'' \times A_f = (\text{HRRPUA}/\Delta H_f) \times A_f$. If the compartment is well-ventilated, all the fuel released by the fuel source gets consumed by the flame and the fire size (*i.e.*, the heat release rate) is: $\dot{Q} = \dot{m}_f \times \Delta H_f = \text{HRRPUA} \times A_f$.

In the *FirstTry* example, we have:

$$\dot{m}_f = \frac{\text{HRRPUA}}{\Delta H_f} \times A_f = \frac{555.55 \times 10^3}{49.6746591 \times 10^6} \times (0.6 \times 0.6) = 4.026 \times 10^{-3} \text{ kg/s}$$

and

$$\dot{Q} = \text{HRRPUA} \times A_f = (555.55 \times 10^3) \times (0.6 \times 0.6) = 200 \text{ kW}$$

These values correspond to the fully-developed phase of the fire. This fully-developed phase is preceded by a growth phase and followed by a decay phase during which the fuel mass loss rate is time-dependent. In FDS, the time variations of $\dot{m}_f$ and $\dot{m}_f''$ are specified through the RAMP functionality. See the FDS User's Guide for details on the RAMP functionality.

1.4 Solid Wall Temperatures

&SURF ID = 'SIDE_WALL',

MATL_ID = 'CONCRETE',

THICKNESS = 0.15 /

&SURF ID = 'FLOOR',

MATL_ID = 'CONCRETE',

THICKNESS = 0.15 /

&SURF ID = 'CEILING',

MATL_ID = 'CONCRETE',

THICKNESS = 0.15 /

&MATL ID = 'CONCRETE',

CONDUCTIVITY = 0.125

DENSITY = 525.

SPECIFIC_HEAT = 1.05 / from CFAST database

These statements define the thermal properties of the solid walls. These properties are needed in a conjugate heat transfer calculation, as explained below.

An important component in fire dynamics is the heat transfer that occurs between the flame, the smoke layer and the solid walls inside the fire compartment. Heat transfer into, or out of the solid walls is controlled in part by the wall surface temperatures. In CFD modeling, the wall surface temperatures are calculated as solutions of a coupled gas-solid heat transfer calculation (called a conjugate heat transfer calculation) in which gas-to-solid heat fluxes drive heat accumulation inside the solid walls and solid conduction describe heat transport inside the walls. The simulation of solid heat conduction requires the values of the heat conductivity, mass density and heat capacity of the walls.

1.5 Spatial Coordinates and Computational Grid Resolution

&OBST XB=2.2,2.8,1.7,2.3,0.0,0.1,

SURF_IDS='BURNER','INERT','INERT' /

&VENT XB=0.0,0.0,1.6,2.4,0.0,2.0, SURF_ID='OPEN' / Doorway

It is important to emphasize that the computational grid resolution defines the prism through which a CFD solver will interpret the geometry of a given fire scenario. In the *FirstTry* simulation, the grid resolution is 10 cm with grid lines uniformly distributed every 10 cm from the boundaries of the computational domain. The vents and obstructions that are introduced in the FDS input file need to conform to the grid lines: in other words, the spatial coordinates of the boundaries of vents and obstructions need to match grid line locations. This explains why the spatial coordinates of the VENT and OBST statements in *FirstTry.fds* are specified as multiples of 0.1 m. Note that if a vent or an obstruction is specified with coordinates that are not grid-conforming, FDS will introduce a correction and adjust the dimension and/or location of the corresponding vent or obstruction, and thereby turn it into a grid-conforming object. This modification may change the exact dimensions of the corresponding vent or obstruction.

1.6 Importance of Order of Statements

&VENT XB=0.0,0.0,1.6,2.4,0.0,2.0, SURF_ID='OPEN' / Doorway

&VENT MB='XMIN', SURF_ID = 'SIDE_WALL' /

The order of the statements in an FDS input file is sometimes important. This is the case when FDS finds conflicting statements. For instance, the first statement above produces an open boundary in the planar (doorway) region defined by $x = 0$, $1.6 \leq y \leq 2.4$ m and $0 \leq z \leq 2$ m. The second statement produces a wall boundary across the entire $x = 0$ plane. These two statements disagree in the doorway region and because of the conflict, FDS resorts to a default rule for conflict resolution; the default rule in FDS is to give preference to the first statement. Note that if the order of the two statements above was reversed, the doorway region would be treated (incorrectly) as a closed wall.

1.7 Diagnostics

&DEVC XB=0.0,0.0,1.6,2.4,0.0,2.0, QUANTITY='MASS FLOW +', ID='MFRp' /

&DEVC XB=0.0,0.0,1.6,2.4,0.0,2.0, QUANTITY='MASS FLOW -', ID='MFRm' /

One of the main responsibilities of CFD users is to include relevant diagnostics in the input file so that useful information may be extracted from a given simulation. In FDS, this is generally accomplished by using the DEVC functionality. (Note that Smokeview provides animations and

graphics that provide valuable visualization and insight but remain essentially qualitative. In contrast, the DEVC functionality allows for quantitative diagnostics.) The DEVC functionality is illustrated in the *FirstTry* example by introducing two flow rate diagnostics that record the time variations of the mass inflow rate and mass outflow rate across the doorway vent. See the FDS User's Guide for details on the DEVC functionality.

1.8 Results from FirstTry

We present below a sample of the results extracted from the simulation of *FirstTry.fds*. Figure 1.1 presents the time variations of the fuel mass loss rate (MLR). Since MLR is user-prescribed (see Section 1.3), the curve presented in Fig. 1.1 is only a consistency check, *i.e.*, a check that the MLR variations have been correctly specified in the FDS input file. Figure 1.1 shows the prescribed sequence of the MLR evolution characterized by a growth phase (up to approximately 80 s), a fully-developed phase (from 80 s until 300 s) and a decay phase (with burnout occurring at approximately 430 s).

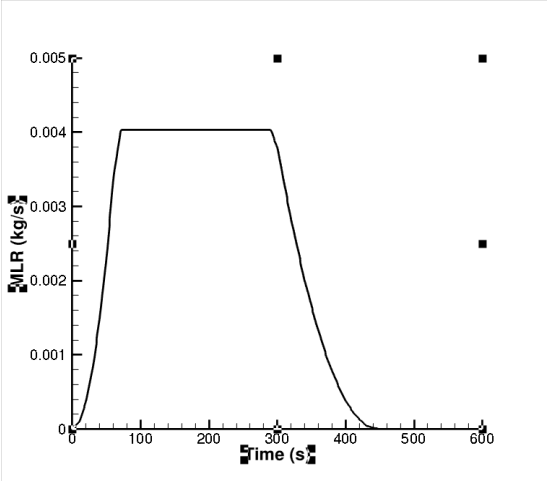

Figure 1.1: Time variations of the fuel mass loss rate. This plot is extracted from the output file called FirstTry_hrr.csv.

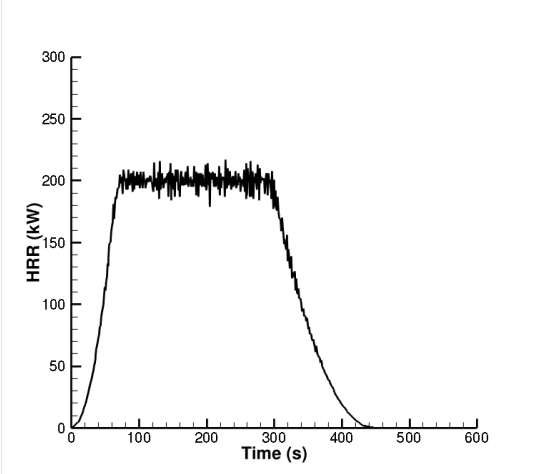

Figure 1.2: Time variations of the heat release rate. This plot is extracted from the output file called FirstTry_hrr.csv.

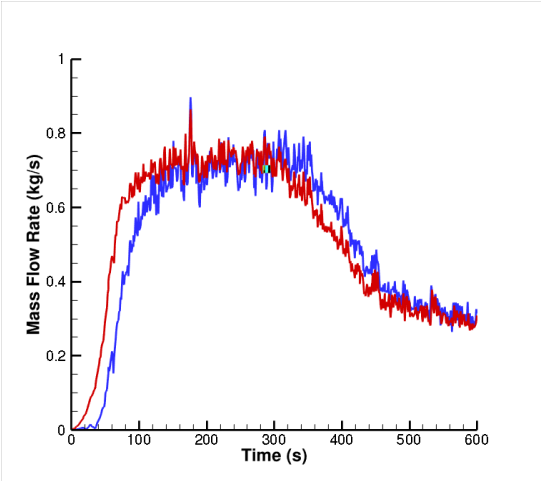

Figure 1.3: Time variations of the mass inflow (blue) and outflow (red) rates at the doorway. This plot is extracted from the output file called FirstTry_devc.csv.

Figure 1.2 presents the corresponding time variations of the heat release rate (HRR). As expected from the discussion in Section 1.3, during the fully-developed phase, the fire size reaches an average value of 200 kW. The (small) fluctuations in HRR are due to unsteadiness associated with turbulent motions.

Figure 1.3 presents the time variations of the mass inflow rate, $\dot{m}_{in}$, and mass outflow rate, $\dot{m}_{out}$, at the doorway. Figure 1.3 shows that after an initial transient phase, a steady-state regime is achieved during which $\dot{m}_{in} \approx \dot{m}_{out}$. The doorway flow rates are driven by the heat release process and therefore the variations of $\dot{m}_{in}$ and $\dot{m}_{out}$ follow those of the fire size.

2 Verification Test of Mass Conservation: *TestMass.fds*

We now turn to a discussion of a test case called *TestMass* aimed at verifying global mass conservation in FDS and aimed also at providing an example of a grid convergence study. The FDS input file is an ASCII text file called *TestMass.fds* (this file will be provided at the Workshop). We review below the main features of *TestMass.fds*.

2.1 Main Features

The example case *TestMass* is similar to the *FirstTry* case presented in the previous section. The configuration features a reduced-scale cubic-shaped compartment of size 1 m exposed to a 200-kW (methane-fueled) fire and featuring a vertical vent ($W_O = H_O = 0.8$ m). The simulation is run over a duration of 30 s with different levels of computational mesh resolution:

&MESH IJK=40,40,40, XB=0.0,1.0,0.0,1.0,0.0,1.0 / $\Delta x = \Delta y = \Delta z = 0.025$ m

or

&MESH IJK=20,20,20, XB=0.0,1.0,0.0,1.0,0.0,1.0 / $\Delta x = \Delta y = \Delta z = 0.05$ m

&MESH IJK=10,10,10, XB=0.0,1.0,0.0,1.0,0.0,1.0 / $\Delta x = \Delta y = \Delta z = 0.1$ m

&MESH IJK=5,5,5, XB=0.0,1.0,0.0,1.0,0.0,1.0 / $\Delta x = \Delta y = \Delta z = 0.2$ m

The main diagnostics are the mass inflow and outflow rates across the vertical vent:

&DEVC XB=0.0,0.0,0.1,0.9,0.0,0.8, QUANTITY='MASS FLOW +', ID='MFRp'/

&DEVC XB=0.0,0.0,0.1,0.9,0.0,0.8, QUANTITY='MASS FLOW -', ID='MFRm'/

2.2 Results from TestMass

We present below a sample of the results extracted from the simulation of *TestMass.fds*. Figure 2.1 presents the time variations of the mass inflow rate, $\dot{m}_{in}$, and mass outflow rate, $\dot{m}_{out}$, at the vertical vent. These results were obtained for $\Delta x = 0.025$ m . Figure 2.1 shows that after an initial transient phase, a steady-state regime is achieved during which $\dot{m}_{in} \approx \dot{m}_{out}$. Note that this statement of mass conservation is written in a time-average sense and that it assumes that the fuel mass loss rate remains small (in all rigor, at steady state, we write $\overline{\dot{m}_{in}} + \overline{\dot{m}_f} \approx \overline{\dot{m}_{out}}$, where the overbar denotes a time-average quantity and where we have $\overline{\dot{m}_f} \ll \overline{\dot{m}_{in}}$).

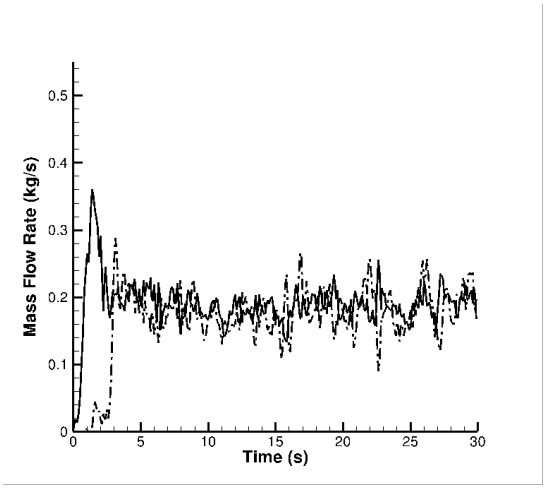

Figure 2.1: Time variations of the mass inflow (dashed line) and outflow (solid line) rates at the doorway. This plot is extracted from the output file called TestMass_devc.csv. Simulation with $\Delta x = 0.025$ *m.*

Figure 2.2 compares the values of $\dot{m}_{in}$ and $\dot{m}_{out}$ obtained in simulations with different levels of grid resolution. Using the concepts presented in Section 1.1, the relevant length scales of the problem are: the burner size D = 0.2 m; the flame length $L_f = (0.235 \times \dot{Q}^{2/5} - 1.02 \times D_f) \approx$ 1.73 m (note that in the present scenario, the flame length is larger than the ceiling height and the flame will therefore impinge on the ceiling); and the vent width and height, $W_O = H_O = 0.8$ m. Because we focus here on the flow rates at the vent, the most relevant length scales from this list are the dimensions W_O and H_O, and we expect an accurate simulation of the vent flow provided that $(W_O/\Delta y) \geq 10$ and $(H_O/\Delta z) \geq 10$, or equivalently $\Delta y \leq 0.08$ m and $\Delta z \leq 0.08$ m. This discussion suggests that simulations with $\Delta x = 0.025$ m and $\Delta x = 0.05$ m are adequately resolved, whereas simulations with $\Delta x = 0.1$ m and $\Delta x = 0.2$ m are under-resolved. This analysis is consistent with the results displayed in Figure 2.2: while the numerical results obtained with $\Delta x =$ 0.1 or 0.2 m are sensitive to changes in grid resolution (which suggests that the corresponding solutions are strongly affected by numerical errors), the results obtained with $\Delta x = 0.025$ or 0.05 m seem to become independent of further grid refinement and the corresponding solutions can be considered as "grid-converged".

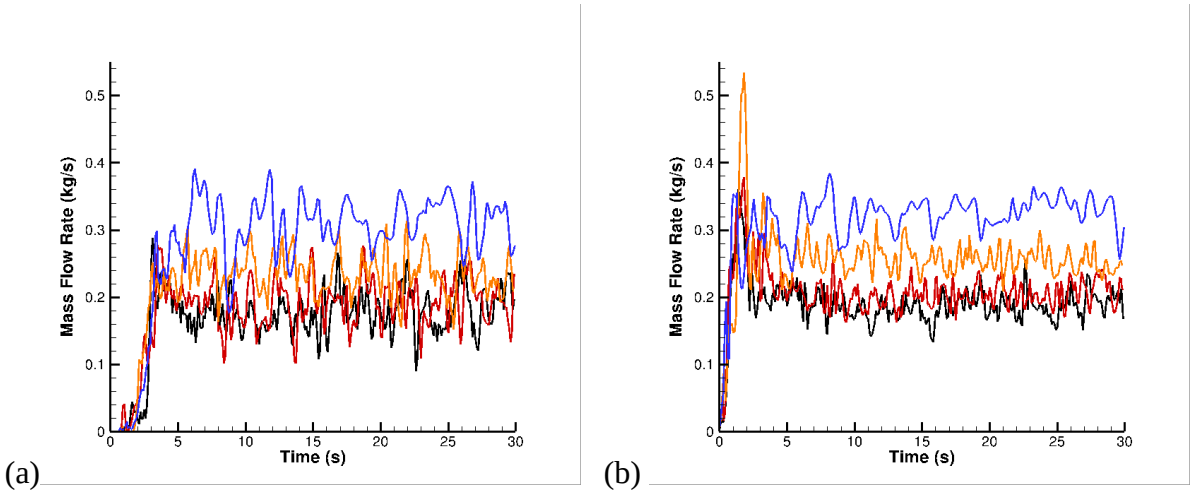

Figure 2.2: Time variations of: (a) the mass inflow rate; (b) the mass outflow rate at the doorway. This plot is extracted from the output files called TestMass_devc.csv. Comparison between simulations performed with different levels of grid resolution: $\Delta x = 0.2$ m (blue); $\Delta x = 0.1$ m (orange); $\Delta x = 0.05$ m (red); $\Delta x = 0.025$ m (black).

This example provides a simple illustration of the importance of grid resolution on the accuracy of a CFD simulation as well as a brief discussion of the classical approach for the assessment of grid quality based on a length scale analysis and a grid convergence study.

3 Verification Test of Momentum Conservation: *Poiseuille.fds*

We now turn to a discussion of a test case called *Poiseuille* aimed at verifying global momentum conservation in FDS in the context of a laminar flow. The FDS input file is an ASCII text file called *Poiseuille.fds* (this file will be provided at the Workshop). We review below the main features of *Poiseuille.fds*.

3.1 Main Features

The example case *Poiseuille* is a simple channel flow configuration. The configuration features laminar Poiseuille flow between two infinite parallel plates. The separation distance between the plate is $H = 0.1$ m; the volume flow rate is prescribed and equal to $\dot{Q}_{flow} = 6.67 \times 10^{-5}$ m^3/s (a small value selected so that the flow is laminar). The simulation is run over a duration of 200 s.

The main relevant length scale in this problem is the separation distance H and we choose $\Delta z = 0.005$ m or $(H/\Delta z) = 20$. The streamwise length of the channel is 2 m (a large value selected so that the computational domain is long enough to include both an entry length region and a large region with fully-developed Poiseuille flow behavior).

The inlet boundary condition at $x = 0$ corresponds to a plug flow at the prescribed flow rate $\dot{Q}_{flow}$:

&SURF ID = 'INLET', VOLUME_FLOW=-0.0000667 /

The minus sign in front of the value of $\dot{Q}_{flow}$ is a sign convention that indicates the direction of the flow (the sign convention is to count outflow as positive and inflow as negative; see the FDS User's Guide for details).

The front and rear planes at $y = 0$ and $y = 0.1$ m correspond to "mirror" boundary conditions:

&VENT MB='YMIN', SURF_ID = 'MIRROR' /

&VENT MB='YMAX', SURF_ID = 'MIRROR' /

These boundary conditions are special symmetry conditions that impose zero normal velocity ($v = 0$) and zero normal gradients for tangent velocity components ($\partial u/\partial y = \partial w/\partial y = 0$) and that help maintain two-dimensional flow.

The main diagnostics are the streamwise variations of pressure along the channel:

&DEVC XYZ=0.0,0.05,0.05, QUANTITY='PRESSURE', ID='localP0' /

&DEVC XYZ=0.2,0.05,0.05, QUANTITY='PRESSURE', ID='localP1' /

…

&DEVC XYZ=2.0,0.05,0.05, QUANTITY='PRESSURE', ID='localP10' /

Note that in FDS, "PRESSURE" denotes the aerodynamic component of pressure (in FDS, pressure is decomposed into the sum of a background component and a flow-induced perturbation, and "PRESSURE" refers to the flow-induced perturbation). There is no gravity in the present problem, the background component is therefore uniform and equal to its atmospheric value; under such conditions, "PRESSURE" indicates gauge pressure. See the FDS User's Guide for details on the meaning of "PRESSURE" and the Technical Reference Guide for details on the pressure decomposition.

3.2 Results from Poiseuille

We present below a sample of the results extracted from the simulation of *Poiseuille.fds*. Figure 3.1 presents the simulated streamwise variations of gauge pressure along the channel centerline. Figure 3.1 also compares the simulated variations to the predicted variations given by the classical Poiseuille flow solution:

$$\Delta p(x) = \frac{12\mu\dot{Q}_{flow}}{H^3 W} \times (x_{exit} - x) = \frac{12 \times (1.82 \times 10^{-5}) \times (6.67 \times 10^{-5})}{(0.1)^3 \times 0.1} \times (2 - x)$$

where μ is the dynamic viscosity of ambient air, $\mu = 1.82 \times 10^{-5}$ kg/m/s, W the y-dimension of the computational domain, $W = 0.1$ m, and x_{exit} the x-length of the channel, $x_{exit} = 2$ m. The comparison is excellent (slight variations in the entry length region, at $x \leq 0.3$ m, are explained by the transition from plug flow to Poiseuille flow).

This case provides a simple example of pressure losses in a channel flow and verifies the ability of FDS to correctly simulate the coupling between flow velocity and pressure.

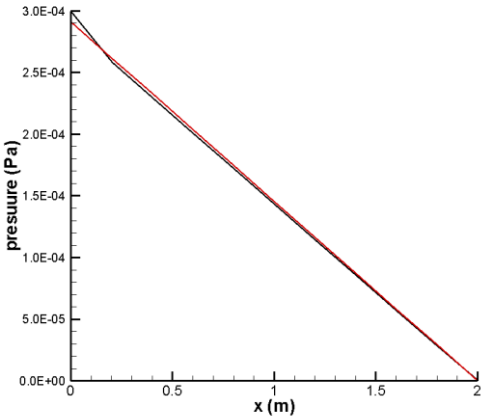

Figure 3.1: Spatial variations of pressure along the channel centerline. This plot is extracted from the output file called Poiseuille_devc.csv. Comparison between FDS results (black) and the analytical Poiseuille flow solution (red).

4 Verification Test of Momentum Conservation: *Turbulent_Channel.fds*

We now turn to a discussion of a test case called *Turbulent_Channel* aimed at verifying global momentum conservation in FDS in the context of a turbulent flow. The FDS input file is an ASCII text file called *Turbulent_Channel.fds* (this file will be provided at the Workshop). We review below the main features of *Turbulent_Channel.fds*.

4.1 Main Features

The example case *Turbulent_Channel* is similar to the *Poiseuille* case presented in the previous section. The configuration features turbulent flow in a square channel. The size of the square channel is $H = 0.1$ m; the volume flow rate is prescribed and equal to $\dot{Q}_{flow} = 0.2\ \mathrm{m}^3/\mathrm{s}$ (a large value selected so that the flow is turbulent; the Reynolds number of the channel flow is $Re_H = (\bar{u}H/\nu) = (\dot{Q}_{flow}/(W\nu)) = (0.2/(0.1 \times (1.5 \times 10^{-5}))) \approx 133333$, where ν is the kinematic viscosity of ambient air, $\nu = 1.5 \times 10^{-5}\ \mathrm{m}^2/\mathrm{s}$). The streamwise length of the channel is increased to $x_{exit} = 4$ m (a large value selected so that the computational domain is long enough to include both an entry length region and a large region with fully-developed turbulent flow behavior); grid resolution is unchanged. The simulation is run over a duration of 5 s (because of the large values of the flow velocity, $\bar{u} = 20\ \mathrm{m/s}$, and associated rates of turbulent mixing, transition to steady state is fast in this case and the simulation does not need to be run for a long time; the characteristic time scale of the problem is the mean flow-through time $(x_{exit}/\bar{u}) = (4/20) = 0.2$ s).

4.2 Results from Turbulent_Channel

We present below a sample of the results extracted from the simulation of *Turbulent_Channel.fds*. Figure 4.1 presents the simulated streamwise variations of mean (time-averaged) gauge pressure along the channel centerline.

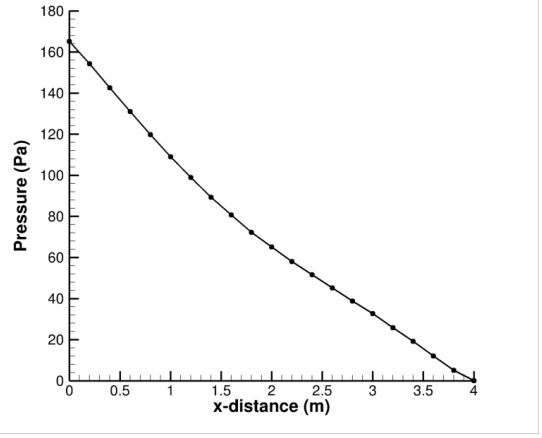

Figure 4.1: Spatial variations of mean (time-averaged) pressure along the channel centerline. This plot is extracted from the output file called Turbulent_Channel_devc.csv.

Figure 4.1 shows that pressure variations in this case are much larger than those found in the previous laminar case (Figure 3.1). This is consistent with expectations since the mean flow velocity has been increased by a factor of 3000 (compare the values of $\dot{Q}_{flow}$) and since pressure losses increase with flow velocity. The change in slope of the pressure gradient observed at $x \approx 1.5$ m in Figure 4.1 is due to the transition from the entry length region to the fully-developed turbulent flow region.

The data presented in Figure 4.1 can be used to calculate the simulated pressure gradient in the fully-developed turbulent region. Using data between $x = 2$ and 4 m, one finds: $(d\bar{p}/dx) \approx -32$ Pa/m. This value can be compared to the predictions given by the classical correlation due to Colebrook and White:

$$(d\bar{p}/dx) = -f \times (\frac{1}{2}\rho\bar{u}^2) \times \frac{1}{H}$$

where *f* is the Darcy friction factor and is calculated from the following implicit equation:

$$\frac{1}{\sqrt{f}} = -2 \times log_{10}(\frac{2.51}{Re_H\sqrt{f}})$$

We find: $f \approx 0.01695$ and $(d\bar{p}/dx) \approx -40$ Pa/m. Thus, the simulated pressure losses are approximately 20% lower than those calculated by the Colebrook and White correlation.

This difference may be explained by the fact that the near-wall turbulent boundary layer dynamics are not captured in the present simulation but rather are modelled through the use of a wall function (FDS assumes the classical logarithmic law of the wall; see the FDS Technical Reference Guide for details). The quality of near-wall grid resolution can be evaluated by examining the distance of the first off-wall grid node for streamwise velocity *u*: $\Delta z_1 = (\Delta z/2) = 0.0025$ m (note that FDS uses a staggered grid and *x*-velocities are evaluated at half distances in the vertical direction, *i.e.*, at $z_{min} + (\Delta z/2)$, $z_{min} + (3 \times \Delta z/2)$, …, $z_{max} - (\Delta z/2)$). Following standard turbulent boundary layer analysis, the distance of the first off-wall grid node is re-cast in wall viscous units: $\Delta z_1^+ = (u_\tau \times \Delta z_1/\nu)$, where u_τ is the wall friction velocity, defined as $u_\tau = \sqrt{\tau_w/\rho}$, with τ_w the wall shear stress. The wall shear stress τ_w can be calculated from a momentum balance statement (at steady state, viscous drag is balanced by pressure losses); we write:

$$\tau_w = (-d\bar{p}/dx) \times (H/4) \approx (32) \times (0.1/4) \approx 0.8 \text{ Pa}$$

and from there:

$$u_\tau = \sqrt{\tau_w/\rho} = \sqrt{0.8/1.2} \approx 0.8 \text{ m/s} \Delta z_1 = (\Delta z/2) = 0.0025 \text{ m}$$

$$\Delta z_1^+ = (u_\tau \times \Delta z_1/\nu) = (0.8 \times 0.0025\ /(1.5 \times 10^{-5})) \approx 133$$

This calculation shows that the simulation does not capture the wall velocity gradients (this would require $\Delta z_1^+ \approx 1$) and therefore relies on a subgrid-scale wall model to reconstruct these gradients and calculate the drag forces. The simulations are said to be "wall-modelled" (rather than "wall-resolved"). The classical logarithmic law of the wall adopted by FDS is valid provided that $\Delta z_1^+ \leq (100 - 300)$ and accuracy is expected to increase for lower values of Δz_1^+. While not shown here, the 20% error on pressure losses observed in the present case is expected to decrease at higher resolution.

This case provides another example of pressure losses in a channel flow and contributes to verify the ability of FDS to correctly simulate the coupling between flow velocity and pressure. It also provides a simple illustration of the importance of near-wall grid resolution in problems dominated by wall dynamics as well as a brief discussion of the classical approach for the assessment of grid quality based on a length scale analysis applied to the wall boundary layer.

5 Verification Test of Energy Conservation: *test_energy.fds*

We now turn to a discussion of a test case called *test*_energy aimed at verifying global energy conservation in FDS. The FDS input file is an ASCII text file called *test_energy.fds* (this file will be provided at the Workshop). We review below the main features of *test_energy.fds*.

5.1 Main Features

The example case *test_energy* is similar to the *FirstTry* and *TestMass* cases presented in Sections 1 and 2. The configuration features a reduced-scale cubic-shaped compartment of size 1 m exposed to a 40-kW methane-fueled fire and featuring a vertical vent ($W_O = H_O = 0.6$ m). The simulation is run over a duration of 60 s with a 2-cm computational mesh resolution.

5.2 Results from test_energy

We present below a sample of the results extracted from the simulation of *test_energy.fds*. We focus here on the information contained in the output file called *test_energy_hrr.csv*: the first column in this file is time; the second column, called HRR, is the heat release rate (spatially-averaged over the volume of the computational domain); the third column, called Q_RADI, is the rate of energy loss through walls and vents due to radiative heat transfer (spatially-averaged over the surfaces of the boundaries of the computational domain); the fourth column, called Q_CONV, is the rate of energy loss through open vents due to convective transport (spatially-averaged over the surfaces of the vents); and the fifth column, called Q_COND, is the rate of energy loss through walls due to conduction (generally referred to as wall "convective" heat transfer) (spatially-averaged over the surfaces of the walls). Note that the sign of Q_RADI, Q_CONV and Q_COND is negative (the sign convention is to count gains as positive and losses as negative).

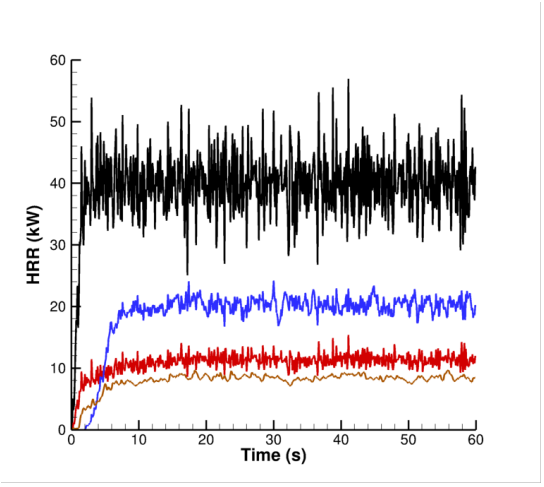

Figure 5.1: Temporal variations of the mean (spatially-averaged) values of: heat release rate, HRR (black); rate of radiation heat loss through walls and vents, Q_RADI (red); rate of convection heat loss through open vents, Q_CONV (blue); rate of conduction heat loss through walls, Q_COND (brown). This plot is extracted from the output file called test_energy_hrr.csv.

Figure 5.1 presents the simulated time variations of HRR, Q_RADI, Q_CONV and Q_COND. We find that the corresponding time-averaged values (averaged for $20 \leq t \leq 60$ s, *i.e.*, after the initial transient and once steady state has been reached) are: $\overline{HRR} \approx 40$ kW; $-\overline{Q_{RADI}} \approx 11.4$ kW; $-\overline{Q_{CONV}} \approx 20.3$ kW; $-\overline{Q_{CONV}} \approx 8.5$ kW. Thus, we find that at steady state, we can write: $\overline{HRR} \approx -(\overline{Q_{RADI}} + \overline{Q_{CONV}} + \overline{Q_{CONV}})$ (within 0.5%). This result is a check of global conservation of energy: the energy released by the combustion process is stored as heat inside the fire plume and the smoke layer and convected out of the fire room through Q_CONV (in the present configuration, this energy path accounts for approximately 50% of the combustion heat release) or is lost to the walls due to radiative heat transfer through Q_RADI or "convective" heat transfer through Q_CONV).

This case provides a simple example of the heat dynamics inside a fire room and contributes to verify the ability of FDS to correctly simulate the coupling between flow, combustion and radiation processes.

6 Verification Test of Flow Solver: *vortex_N.fds*

We now turn to a discussion of a test case called *vortex_N* (where *N* is an integer equal to 2, 5, 10, 20 or 40) aimed at providing another example of a grid convergence study in the context of a laminar flow problem that has a known analytical solution. The FDS input files are ASCII text files called *vortex_N.fds* (the file *vortex_10.fds* will be provided at the Workshop). We review below the main features of *vortex_N.fds*.

6.1 Main Features

The example case *vortex_N* corresponds to a basic laminar flow configuration known as the Lamb-Oseen vortex. The configuration features relevant flow dynamics through convection and viscous decay and allows a quantification of the numerical error through comparisons between the numerical and analytical solutions.

In a two-dimensional (x, z) rectangular Cartesian coordinate system, the analytical solution for the Lamb-Oseen vortex can be expressed as:

$$u = u_{\theta,max}\left(\frac{z - z_c}{r_c}\right)\exp\left(\frac{1}{2}(1 - \frac{(x - x_c)^2 + (z - z_c)^2}{r_c^2})\right) + u_{coflow}$$

$$w = -u_{\theta,max}\left(\frac{x - x_c}{r_c}\right)\exp\left(\frac{1}{2}(1 - \frac{(x - x_c)^2 + (z - z_c)^2}{r_c^2})\right)$$

where r_c and $u_{\theta,max}$ are the characteristic size and velocity of the vortex structure at a particular time, (x_c, z_c) the coordinates of the center of the vortex, and u_{coflow} the velocity of an assumed uniform co-flow in the *x*-direction. In these expressions, the variables r_c, $u_{\theta,max}$ and (x_c, z_c) are time-dependent and their time variations are given by:

$$r_c = (r_{c,0} + 2\nu t)^{1/2}$$

$$u_{\theta,max} = \frac{u_{\theta,max,0}}{(1 + \frac{2\nu t}{r_{c,0}^2})^{3/2}}$$

$$(x_c, z_c) = \left((x_{c,0} + u_{coflow} \times t), z_{c,0}\right)$$

where $r_{c,0}$, $u_{\theta,max,0}$ and $\left(x_{c,0}, z_{c,0}\right)$ are the prescribed values of r_c, $u_{\theta,max}$ and (x_c, z_c) at initial time, $t = 0$, and where ν is the kinematic viscosity of ambient air. In the following, we use: $r_{c,0} = 0.005$ m; $u_{\theta,max,0} = 0.5$ m/s; $u_{coflow} = 0.1$ m/s; $\left(x_{c,0}, z_{c,0}\right) = (0,0)$; and $\nu = 10^{-5}$ m^2/s.

The initial flow velocity field is generated using a MATLAB script file called *Initialize_vortex.m* (this file will be provided at the Workshop). The MATLAB program generates the initial velocity field in a two-dimensional computational domain of size (0.1×0.1) m^2 and using a uniform grid with $(10 \times N) \times (10 \times N)$ computational grid cells; we consider different levels of grid resolution: $(2 \times r_{c,0})/\Delta x = N$, with $N = 2, 5, 10, 20, 40$. Note that the MATLAB program pays close attention to the staggered grid format adopted in FDS: *x*-velocities, *u*, are evaluated at cell faces in the *x*-direction (*i.e.*, at $(x_{min} + i \times \Delta x)$, where *i* is a running index for the *x*-grid) and cell centers in the *z*-direction (*i.e.*, at $(z_{min} + k \times (\Delta z/2))$, where *k* is a running index for the *z*-grid), while *z*-velocities, *w*, are evaluated at cell centers in the *x*-direction (*i.e.*, at $(x_{min} + i \times (\Delta x/2))$, where *i* is a running index for the *x*-grid) and cell faces in the *z*-direction (*i.e.*, at $(z_{min} + k \times \Delta z)$, where *k* is a running index for the *z*-grid). Finally, the MATLAB program stores the velocity field in a

comma-separated values (CSV) file called *init_vortex_N_uvw.csv*. These files are then read by FDS and used as initial conditions for simulations.

Let us consider the case $N = 10$. The input file *vortex_10.fds* contains the following statement that instructs FDS to read the external file *init_vortex_10_uvw.csv* (this file has been previously copied to the run directory):

&CSVF UVWFILE = 'init_vortex_10_uvw.csv' /

The example case *vortex_10* features a Lamb-Oseen vortex structure convected by a coflow inside a two-dimensional computational domain of size 0.1 m with a grid resolution $\Delta x = 0.001$ m ($(2 \times r_{c,0})/\Delta x = 10$). The simulation is run over a duration of 3 s.

&MESH IJK = 100,1,100, XB = -0.05,0.05, -0.001,0.001, -0.05,0.05 /

&TIME T_END = 3.0 /

At time $t = 3$ s, the vortex has covered a distance of $(u_{coflow} \times t) = 0.3$ m, *i.e.*, 3 times the size of the computational domain. While this problem could be simulated using a computational domain large enough to capture the entire displacement of the vortex structure, it is more effective to simulate the problem using periodic boundary conditions:

&VENT MB='XMIN', SURF_ID = 'PERIODIC' /

&VENT MB='XMAX', SURF_ID = 'PERIODIC' /

The periodic boundary conditions applied at the west and east boundaries of the domain essentially simulate an infinite train of vortices along the *x*-direction separated by a 0.1-m distance: when the vortex leaves the domain at the east boundary, the same vortex re-enters at the west boundary. Thus, at time $t = 3$ s, the center of the vortex structure is back to its original position at $(x_c, z_c) = (0,0)$.

Furthermore, the viscosity of the gas (FDS assumes air as the default choice) is artificially modified with the following statement:

&SPEC ID = 'AIR', VISCOSITY = 1.199266e-5, BACKGROUND = .TRUE. /

where VISCOSITY is the dynamic viscosity: $\mu = 1.199266 \times 10^{-5}$ kg/m/s; and because the mass density of air (at normal temperature and pressure conditions) is $\rho = 1.199266$ kg/m^3, we have $\nu = (\mu/\rho) = 10^{-5}$ m^2/s.

The main diagnostic is the profile of the variations of *z*-velocity along the line $z = 0$:

&DEVC ID = 'W-Vel', XB = -0.0495,0.0495, 0.0,0.0, -0.0005,-0.0005,

QUANTITY = 'W-VELOCITY', POINTS = 100, TIME_AVERAGED = .FALSE. /

This DEVC statement creates an output file called *vortex_10_line.csv* that contains QUANTITY data along the line specified by the XB coordinates. The default time-averaging feature available with this diagnostic is turned off here. During the simulation, *vortex_10_line.csv* is continuously updated with instantaneous profiles and, at the end of the simulation, *vortex_10_line.csv* contains the last profile obtained at time $t = 3$ s. Note that in the DEVC statement, the XB coordinates are the coordinates of the cell centers along the horizontal line of interest (at $z = -(\Delta z/2)$) but, consistent with the staggered grid format, FDS will actually evaluate the W-VELOCITY at the centers of the north faces along this line (*i.e.*, at $z = 0$). In summary, the DEVC statement allows the extraction of the *z*-velocity profile along the line $z = 0$ and at the final time $t = 3$ s.

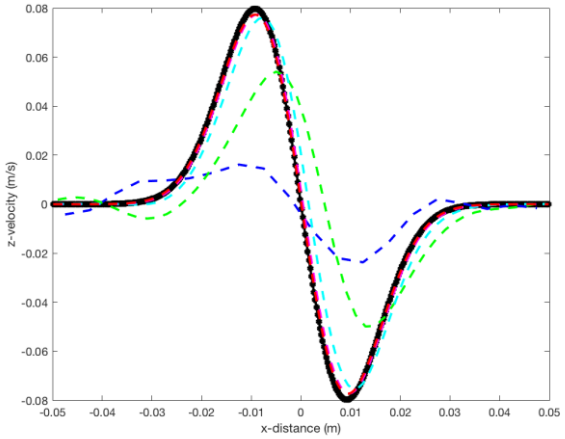

Figure 6.1: Spatial variations of z-velocity along the line $z = 0$ and at time $t = 3$ s. This plot is extracted from the output files called vortex_N_line.csv. Comparison between simulations performed with different levels of grid resolution: $\Delta x = 0.005$ m and $(2 \times r_{c,0})/\Delta x = 2$ (blue dashed line); $\Delta x = 0.002$ m and $(2 \times r_{c,0})/\Delta x = 5$ (green dashed line); $\Delta x = 0.001$ m and $(2 \times r_{c,0})/\Delta x = 10$ (cyan dashed line); $\Delta x = 0.0005$ m and $(2 \times r_{c,0})/\Delta x = 20$ (magenta dashed line); $\Delta x = 0.00025$ m and $(2 \times r_{c,0})/\Delta x = 40$ (red dashed line). The FDS results are also compared to the analytical Lamb-Oseen vortex flow solution (black solid line).

6.2 Results from vortex_N

We present below a sample of the results extracted from the simulation of *vortex_N.fds*. Figure 6.1 presents the simulated spatial variations of *z*-velocity along the line $z = 0$ and at time $t = 3$ s. The FDS results are also compared to the analytical Lamb-Oseen vortex flow solution presented in the previous section. In Figure 6.1, the difference between the FDS results and the analytical solution gives a valuable quantification of the numerical error. One finds that for $(2 \times r_{c,0})/\Delta x \geq 10$, the vortex structure is well-resolved and the numerical error remains small; in contrast, for $(2 \times r_{c,0})/\Delta x \leq 5$, the vortex structure is under-resolved and the numerical error is large. The numerical error takes the form of dissipative errors (also known as "numerical diffusion") that act to reduce the strength of the vortex and dispersive errors that affect the speed and location of the vortex center.

This example provides another illustration of the importance of grid resolution on the accuracy of a CFD simulation and confirms the rule of thumb for grid design discussed in Section 1 (the rule suggests using 10 grid cells per relevant physical length scale, here the size of the vortex, $L = (2 \times r_{c,0})$). The example also provides an illustration of how to set initial conditions in FDS and allows a discussion of the subtleties associated with the staggered grid format.

7 Verification Test of Radiation Solver: *radiation_box_50_N.fds*

We now turn to a discussion of a test case called *radiation_box_50_N* (where *N* is an integer equal to 50, 100, 200, 500, 1000 or 2000) aimed at providing an example of angular convergence in the radiation solver. The radiation solver in FDS is based on a decomposition of angular space (4π steradians) into *N* elementary solid angles and a calculation of the radiation intensity (in units of W/m^2/sr) in each elementary solid angle using the radiative transfer equation (RTE) and a Discrete Ordinates Method (DOM). The default choice for the number of angles used in the decomposition of angular space is $N = 100$ and *N* is varied here in order to study angular convergence. The FDS input files are ASCII text files called *radiation_box_50_N.fds* (the file *radiation_box_50_500.fds* will be provided at the Workshop). We review below the main features of *radiation_box_50_N.fds*.

7.1 Main Features

The example case *radiation_box_50_N* corresponds to a basic radiation configuration in which an array of targets is exposed to black body radiation from a source of known temperature (the emissive power of the source is prescribed) and of known location and geometry (the view factor between each target and the source can be calculated analytically), and in which the targets and the source are separated by transparent gas. The analytical expression for the incoming radiative heat flux arriving at each target (also called the irradiation, in units of W/m^2) is: $G = \phi \times (\sigma T_s^4)$, where ϕ is the view factor, σ the Stefan-Boltzmann constant, $\sigma = 5.67 \times 10^{-8}\ W/m^2/K^4$, and T_s the temperature of the source. The values of ϕ for each target in the *radiation_box_50_N* case are pre-calculated and are provided as comments in the FDS input file. The configuration thereby allows a quantification of the numerical error through comparisons between the numerical and analytical solutions.

Let us consider the case $N = 500$. The example case *radiation_box_50_500* features a cubic-shaped computational domain of size 1 m with a grid resolution $\Delta x = 0.02$ m (the spatial resolution plays a minor role in the present test case):

&MESH XB=0.0,1.0,0.0,1.0,0.0,1.0, IJK=50,50,50 /

The source is a (1×1) m^2 hot solid plate located on the vertical plane $x = 0$ and the targets are located on the plane $x = 1$ m. The source is a black body ($\epsilon_s = 1$) surface at $T_s = (273.15 + 91.2717)$ K; its emissive power is $(\sigma T_s^4) = 1$ kW/m^2; possible exchange of heat between the hot plate and the gas through convective heat transfer is turned off ($h = 0$):

&VENT XB=0.0,0.0,0.0,1.0,0.0,1.0, SURF_ID='HOT' /

&SURF ID = 'HOT' TMP_FRONT = 91.2717 TAU_T = 0. EMISSIVITY=1.0,

HEAT_TRANSFER_COEFFICIENT=0. COLOR='RED'/

where TMP_FRONT designates the temperature of the source (in units of degree Celsius), EMISSIVITY the emissivity ϵ_s and HEAT_TRANSFER_COEFFICIENT the convective heat transfer coefficient h.

All other surfaces are treated as black body surfaces with a temperature of 1 K (these surfaces do not reflect and do not emit radiation):

&SURF ID = 'COLD' TMP_FRONT = -272.15 TAU_T = 0. EMISSIVITY=1.0,

HEAT_TRANSFER_COEFFICIENT=0. COLOR='BLUE'/

The gas between the source and the targets is dry air (and is therefore transparent):

&MISC Y_CO2_INFTY=0., HUMIDITY=0. /

The number of radiation angles is $N = 500$:

&RADI NUMBER_RADIATION_ANGLES=500

NUMBER_INITIAL_ITERATIONS=0

TIME_STEP_INCREMENT = 1

ANGLE_INCREMENT = 1 /

The statement "TIME_STEP_INCREMENT = 1" indicates that the radiation solver is called at every time step (*i.e.*, for every call to the gas phase solver). The statement "ANGLE_INCREMENT = 1" indicates that each elementary solid angle in the decomposition of angular space is updated during a call to the radiation solver. These two statements provide maximum accuracy in the radiation calculation.

The simulation is run over one time step (radiation transport is instantaneous and there is therefore no point in simulating this problem for a longer duration):

&TIME T_END=0.1, DT = 0.1 /

The main diagnostics are the heat fluxes measured at different locations along a straight line $(y - z) = 0$ in the plane $x = 1$ m:

&DEVC QUANTITY = 'INCIDENT HEAT FLUX' XYZ = 1.0,0.025,0.025 IOR=-1 /

&DEVC QUANTITY = 'INCIDENT HEAT FLUX' XYZ = 1.0,0.075,0.075 IOR=-1 /

…

&DEVC QUANTITY = 'INCIDENT HEAT FLUX' XYZ = 1.0,0.975,0.975 IOR=-1 /

Note that in FDS, "INCIDENT HEAT FLUX" denotes the incoming radiative heat flux (*i.e.*, the irradiation *G*).

7.2 Results from radiation_box_50_N

We present below a sample of the results extracted from the simulation of *radiation_box_50_N.fds*. Figure 7.1 presents the simulated spatial variations of the irradiation *G* along the line $(y - z) = 0$ at $x = 1\ m$ and compares the values of *G* obtained in simulations with different levels of angular resolution (from 50 to 2000 angles). The FDS results are also compared to the analytical solution presented in the previous section. In Figure 7.1, the difference between the FDS results and the analytical solution gives a valuable quantification of the numerical error. One finds that for $N \geq 200$, the radiation field at $x = 1\ m$ is well-resolved and the numerical error remains small; in contrast, for $N \leq 100$, the radiation field is under-resolved and the numerical error is large. The numerical error takes the form of unphysical oscillations (and may lead to the clearly incorrect result that the irradiation peaks at an off-center location).

The present conclusion that 200 angles provide adequate results is problem-dependent and cannot be generalized in that form. The design of a suitable discretization of angular space, $\Delta\Omega = (4\pi/N)$, may be understood as a process based on a comparison of solid angles. It starts first with the identification of physical solid angles associated with angular regions of interest, $\Delta\Omega_s$, in a given fire scenario (for instance, in the present case, the solid angle associated with the angular region occupied by the hot plate when viewed from one of the targets) and then finishes with a comparison of $\Delta\Omega_s$ to $\Delta\Omega$. The rule of thumb in the design process consists in providing 10 elementary solid angles per relevant angular region: $(\Delta\Omega_s/\Delta\Omega) \geq 10$. If time permits, a second FDS example will be presented during the workshop.

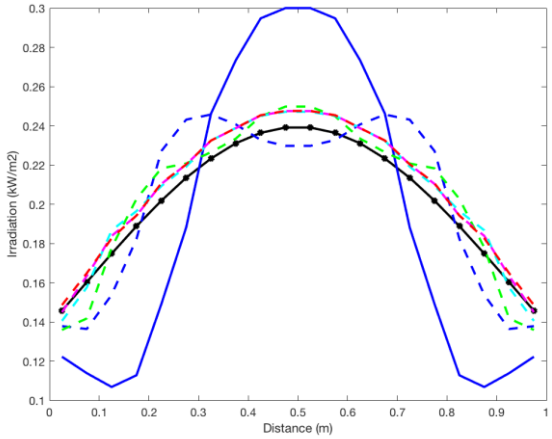

Figure 7.1: Spatial variations of the irradiation along the line $(y - z) = 0$ *at* $x = 1$ *m. This plot is extracted from the output files called radiation_box_50_N_devc.csv. Comparison between simulations performed with different levels of angular resolution:* $N = 50$ *(blue solid line);* $N = 100$ *(blue dashed line);* $N = 200$ *(green dashed line);* $N = 500$ *(cyan dashed line);* $N = 1000$ *(magenta dashed line);* $N = 2000$ *(red dashed line). The FDS results are also compared to the analytical solution (black solid line).*

This example provides an illustration of the importance of angular space discretization on the accuracy of the simulation of radiation transport as well as a brief discussion of the classical approach for the assessment of the quality of angular discretization based on a solid angle analysis and an angular convergence study.

8 Verification Test of Pyrolysis Solver: *wood_pyrolysis_100_N.fds*

We now turn to a discussion of a test case called *wood_pyrolysis_100_N* (where *N* is an integer equal to 100, 200, 400, 800, 1600 or 3200) aimed at providing an example of grid convergence in the (solid phase) pyrolysis solver. The pyrolysis solver in FDS is based on a one-dimensional treatment of solid heat conduction and mass conservation combined with pyrolysis chemistry. The spatial resolution adopted in the solid phase solver is made using a default implicit choice in FDS; this choice is varied here in order to study grid convergence (in the solid phase). The FDS input files are ASCII text files called *wood_pyrolysis_100_N.fds* (the file *wood_pyrolysis_100_100.fds* will be provided at the Workshop). We review below the main features of *wood_pyrolysis_100_N.fds*.

8.1 Main Features

The example case *wood_pyrolysis_100_N* corresponds to a basic cone calorimeter configuration in which a wood sample is exposed to an incoming radiative heat flux (irradiation) of 100 kW/m^2 (this value is high and is in fact higher than the values typically studied in a cone calorimeter; the choice of a high value is made here to facilitate the illustration of the problems associated with grid resolution in the solid phase) and the response of the sample is characterized through measurements of the mass loss rate.

Let us consider the case $N = 100$. The example case *wood_pyrolysis_100_100* features a 2.5-cm-thick wood sample exposed to 100 kW/m^2 irradiation and simulated as a one-dimensional solid phase problem with a uniform grid resolution of $\Delta x_s \approx 100$ μm (in FDS, the resolution used in the solid phase solver is not explicitly controlled in the input file. The relevant information can be

found in the FDS output file called *wood_pyrolysis_100_100.out*; open this file and look for the keywords "*Solid Phase Node, Layer, Coordinates(m)*").

The following statement indicates that the simulation is limited to the solid phase solver:

&MISC SOLID_PHASE_ONLY=.TRUE., Y_O2_INFTY=0.01 /

The cone calorimeter conditions are defined as follows:

&SURF ID = 'WALL'

MATL_ID = 'WOOD'

THICKNESS = 0.025

BACKING = 'INSULATED'

EXTERNAL_FLUX = 100.

RAMP_EF = 'External flux'

HEAT_TRANSFER_COEFFICIENT = 10.

STRETCH_FACTOR = 1

CELL_SIZE_FACTOR = 0.375 /

where MATL_ID defines the material of the sample, THICKNESS its thickness, BACKING the backside thermal boundary condition, EXTERNAL_FLUX and RAMP_EF the irradiation, and HEAT_TRANSFER_COEFFICIENT the convective heat transfer coefficient at the exposed surface. In addition, the statement "STRETCH_FACTOR = 1" indicates that the solid phase solver is using a uniform grid and the statement "CELL_SIZE_FACTOR = 0.375" indicates that the grid resolution is 0.375 times the default implicit choice made in FDS: this factor leads to a value $\Delta x_s \approx 100\ \mu$m.

The simulation is run over a duration 3500 s (a large value selected so that the simulation includes the burn out of the sample)

&TIME TWFIN=3500., WALL_INCREMENT=1 /

The statement "WALL_INCREMENT=1" indicates that the solid phase solver is called at every time step (*i.e.*, for every call to the gas phase solver). This statement provides maximum accuracy in the simulation of the gas-solid phase coupling.

The wood pyrolysis model is defined as follows:

&MATL ID = 'WOOD'

EMISSIVITY = 0.9

CONDUCTIVITY = 0.126

DENSITY = 663.

SPECIFIC_HEAT = 2.52

N_REACTIONS = 1

A = 5.25E+07

E = 1.256E+05

SPEC_ID = 'METHANE'

MATL_ID = 'CHAR'

NU_SPEC = 0.8

NU_MATL = 0.2

HEAT_OF_REACTION = 0 / Source: Novozhilov et al., Fire Safety J. 27 (1996) 69-84

where A and E are the pre-exponential factor and the activation energy associated with the single-step ("N_REACTIONS = 1") pyrolysis chemistry model (see the FDS Technical Reference Guide for details), NU_SPEC and NU_MATL are the (mass) yields of flammable fuel vapors (treated as 'METHANE') and solid residue (called 'CHAR'), and HEAT_OF_REACTION is the heat of pyrolysis assumed in the pyrolysis chemistry model. In this model, 80% of the virgin wood is transformed into fuel vapors and 20% is transformed into char; the transformation is neutral from an energy perspective.

The thermal properties of the char residue are:

&MATL ID = 'CHAR'

EMISSIVITY = 0.9

CONDUCTIVITY = 0.126

DENSITY = 133.

SPECIFIC_HEAT = 2.52 /

The main diagnostic is the fuel mass loss rate per unit exposed surface area (called BURNING RATE in FDS):

&DEVC XYZ=0.02,0.02,0.0, QUANTITY='BURNING RATE', IOR=+3, ID='MLRPUA_1' /

&DEVC XYZ=0.05,0.05,0.0, QUANTITY='BURNING RATE', IOR=+3, ID='MLRPUA_2' /

&DEVC XYZ=0.08,0.08,0.0, QUANTITY='BURNING RATE', IOR=+3, ID='MLRPUA_3' /

The mass loss rate is measured at three different locations in order to check that the problem is indeed one-dimensional. Results show that the three diagnostics provide identical values.

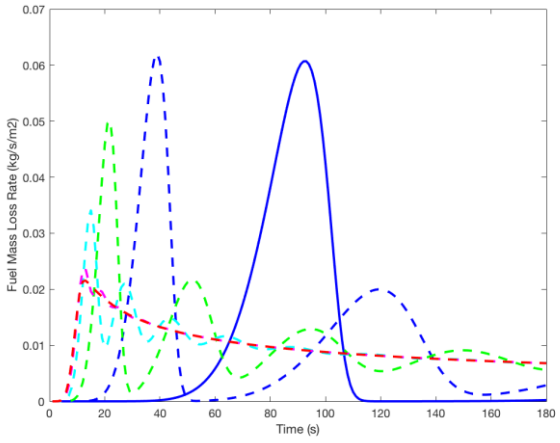

Figure 8.1: Temporal variations of the fuel mass loss rate per unit surface area, $0 \leq t \leq 180$ s. *This plot is extracted from the output files called wood_pyrolysis_devc.csv. Comparison between simulations performed with different levels of grid resolution:* $\Delta x_s \approx 3.2$ mm *(blue solid line);* $\Delta x_s \approx 1.6$ mm *(blue dashed line);* $\Delta x_s \approx 800$ μm *(green dashed line);* $\Delta x_s \approx 400$ μm *(cyan dashed line);* $\Delta x_s \approx 200$ μm *(magenta dashed line);* $\Delta x_s \approx 100$ μm *(red dashed line).*

8.2 Results from wood_pyrolysis_100_N

We present below a sample of the results extracted from the simulation of *wood_pyrolysis_100_N.fds*. Figure 8.1 presents the simulated temporal variations of the fuel mass loss rate per unit surface area (MLRPUA) for the early phase of the pyrolysis process, $0 \leq t \leq 180$ s, and compares the values of MLRPUA obtained in simulations with different levels of grid resolution (from 100 μm to 3.2 mm). One finds that for $\Delta x_s \leq 200\,\mu$m, the solution is grid-converged; in contrast, for $\Delta x_s \geq 400\,\mu$m, the solution is under-resolved and features large unphysical oscillations. The rule of thumb for grid design in the solid phase is to use a resolution of order 100 μm.

If time permits, a second FDS example will be presented during the workshop.

This example provides an illustration of the importance of spatial discretization in the solid phase on the accuracy of the simulation of pyrolysis processes as well as a discussion of the subtleties associated with providing user-control of spatial discretization inside the solid phase.

.