Fire simulation tutorial using Fire Dynamics Simulator (FDS)

January 20, 2017

A short summary what is Fire Dynamics Simulator (FDS) and Smokeview. We give a brief overview its possibilities and do one example step by step. Simulation is run on on the high performance computing center of consortium of University of Tartu - Rocket. It is recommended to read Fire Dynamics Simulator User's Guide, examine sample projects (within FDS folder) and other tutorials for example `https://en.wikibooks.org/wiki/Fire_Simulation_for_Engineers` **[2], which is work-in-progress. Code and scripts can be found in github [9].**

1 FDS and Smokeview

Fire Dynamics Simulator (FDS), is a computational fluid dynamics (CFD) model of fire-driven fluid flow. FDS solves numerically a form of the Navier-Stokes equations appropriate for low-speed (Ma ¡ 0.3), thermally-driven flow with an emphasis on smoke and heat transport from fires. Smokeview is a separate visualization program that is used to display the results of an FDS simulation. [1]

2 Starting to use FDS/Smokeview

2.1 Installation

There are quite detailed instructions made for installation, which are easy to use and tested by myself for Windows and Linux environments. From wikibooks [2] covers Fedora Linux, Ubuntu Linux and Windows XP. From official website [3] you can find

instructions for Windows, Linux and OS X. Installation instructions will direct you to FDS github repository.

2.2 Running example

After successful installation it is possible to use command prompt to use FDS. To test it open command prompt terminal and type

fds jobName.fds

This command launches a single FDS process that will calculate one or more meshes in your simulation. This command will automatically invoke multiple cores of your computer via the OpenMP functionality built into FDS. At best, OpenMP will speed up your job by a factor of 2. For faster speed up, you need to invoke the MPI functionality. [5]

3 Step by step example

First choose your preferred text editor for example Notepad and create a new document jobName.fds. In first line we mark our job ID and title.

&HEAD CHID='rehi', TITLE='Common old Estonian house' /

As we can see that every command starts with & and ends with /. You may write commands in every line or after / as you prefer. In next line we define time how long simulation should run in seconds.

&TIME T_END=900.0 /

3.1 Mesh

In this section we define our world dimensions into which we build our model. We try to build 16,6x7,7x,3,4m space, where every x,y,z is separated into 0,1m sections.

&MESH IJK=166,77,34, XB=0.0,16.6,-1.0,6.7,0.0,3.4 /

In XB we define x,y,z lengths and positions: XB=x1,x2,y1,y2,z1,z2. With IJK we give how many sections we have in a line. For example in our case x is 16,6m long and we wanted one section to be 0,1m, which means we need to divide 16,6 into 166 parts.

3.2 Objects properties

Lets first define materials we will use: brick and wood. All following properies about wood, brick are taken from in reference[10].

&MATL ID = 'WOOD'
CONDUCTIVITY = 0.13
DENSITY = 461.9
EMISSIVITY = 0.92
SPECIFIC_HEAT = 1.250 /
&MATL ID = 'BRICK'
CONDUCTIVITY = 6.4
DENSITY = 1490.
SPECIFIC_HEAT = 0.98
EMISSIVITY = 0.9 /

All material properties needs to be defined by user. Basic properties can be found from the web with some research.

Now we define surface properties. There we will define material it is made of and surface properties.

&SURF ID = 'WALL'
RGB = 200,200,200
MATL_ID = 'WOOD'
THICKNESS = 0.021 /
&SURF ID = 'WOODEN DOOR'
RGB = 255,0,0
MATL_ID = 'WOOD'
THICKNESS = 0.010 /
&SURF ID = 'STONE'
MATL_ID = 'BRICK'
THICKNESS = 0.05
BACKING = 'VOID'
RGB = 127,127,127 /

To make life faster and easier we will define default surface what an obstacle will use when no SURF_ID is provided

&MISC SURF_DEFAULT='WALL' /

3.3 Objects

Now lets place objects into our world. They are called Obstructions (OBST). XB follows same logic as before in defining dimensions for the obstruction: XB=x1,x2,y1,y2,z1,z2.

&OBST XB= 0.00, 16.6, 0.00, 0.20, 0.00, 2.40 / Front wall
&OBST XB= 0.00, 16.6, 5.50, 5.70, 0.00, 2.40 / Back wall
&OBST XB= 0.00, 0.20, 0.00, 5.70, 0.00, 2.40 / Left wall

&OBST XB= 16.4, 16.6, 0.00, 5.70, 0.00, 2.40 / Right wall

&OBST XB= 0.00, 16.6, 0.00, 5.70, 2.40, 2.60 / Roof

&OBST XB= 6.80, 7.00, 0.00, 5.70, 0.00, 2.40 / Left separator wall

&OBST XB= 11.55, 11.75, 1.00, 5.70, 0.00, 2.40 / Right separator wall

In next line we make a stone pile on which we create a fire.

&OBST XB= 10.00, 11.55, 2.00, 4.85, 0.00, 0.50, SURF_ID='STONE' / Stone pile on which wood is placed

Now we have walls, roof, but there are no doors nor windows. For that we use HOLE.

&HOLE XB= 6.80, 7.00, 1.90, 2.85, 0.00, 1.80 / left separator wall door

&HOLE XB= 2.30, 4.40, -0.10, 0.20, 0.00, 1.80 / left room front door

&HOLE XB= 9.10, 9.80, -0.10, 0.20, 1.00, 1.60 / Middle room window

&HOLE XB= 14.00, 14.60, -0.10, 0.20, 1.00, 1.60 / Right room front window

&HOLE XB= 16.40, 16.70, 3.60, 4.20, 1.00, 1.60 / Right room right wall window

&HOLE XB= 11.05, 12.25, -0.10, 0.20, 0.00, 1.80 / right door

&HOLE XB= 11.10, 11.30, 5.50, 5.80, 2.00, 2.20 / Smoke hole in the middle room

Next we will define windows and doors, where they are needed. For simplicity both of them are made of a single piece of wood. In real life there are many components for a door and a window.

&OBST XB= 11.05, 12.25, -0.10, 0.20, 0.00, 1.80, PERMIT_HOLE=.FALSE., SURF_ID='WOODEN DOOR' / right door

&OBST XB= 16.40, 16.70, 3.60, 4.20, 1.00, 1.60, PERMIT_HOLE=.FALSE., SURF_ID='WOODEN DOOR' / Right room right wall window

&OBST XB= 2.30, 4.40, -0.10, 0.20, 0.00, 1.80, PERMIT_HOLE=.FALSE., SURF_ID='WOODEN DOOR' / left room front door

&OBST XB= 14.00, 14.60, -0.10, 0.20, 1.00, 1.60, , PERMIT_HOLE=.FALSE., SURF_ID='WOODEN DOOR' / Right room front window

PERMIT_HOLE=.FALSE., allows us to put OBST into HOLE. With SURF_ID we define what kind of material a surface is made of.

3.4 Fire

We need to give manually a chemical compositions of wood and how it burns. The chemical reaction for the wood burning, the soot production and the heat of combustion are taken from reference [10] [11].

&REAC ID='WOOD' C=1. H=1.7 O=0.74 N=0.002 SOOT_YIELD=0.015 CO_YIELD=0.004 HEAT_OF_COMBUSTION=12600./

Next a surface where reaction takes place and what properties it has, for example power output.

&SURF ID='BURNER', HRRPUA=55., MATL_ID='WOOD', THICKNESS=5e-2, COLOR='RASPBERRY' /

With HRRPUA we define kW/m2 output. Now we create a surface where reaction takes place.

3.5 Vents

Fire or heat source is created using VENT. VENT is also used to control air flow for example to create manual or mechanical air intake/outtake. In our example we will create 2 manual air holes.

&VENT XB= 10.05, 11.50, 2.10, 4.75, 0.50, 0.50, SURF_ID='BURNER' / Ignition source on stone

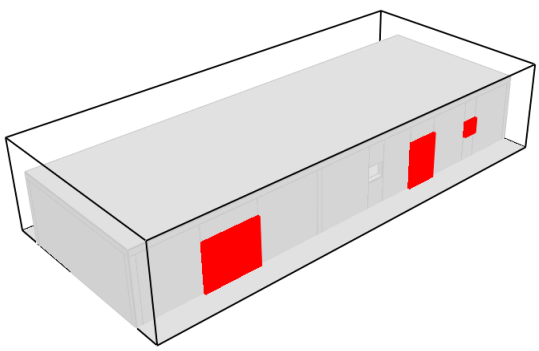
In this case total energy output would be surface area*HRRPUA. 3,8425 m2 * 55 kW/m2 = 211,3375 kW.

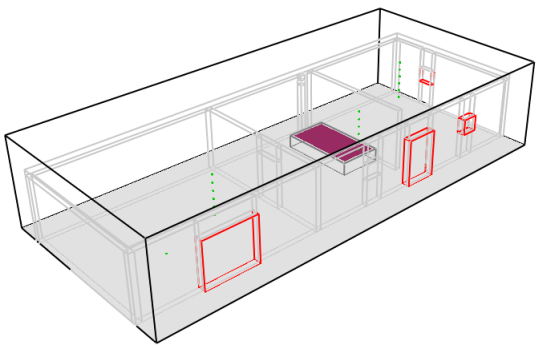
We have currently 2 open holes: small smoke hole in the middle room and middle room window. In our case boundaries where air flows should be opened. In default they are closed.

&VENT MB='XMIN' SURF_ID='OPEN'
&VENT MB='XMAX' SURF_ID='OPEN' /
&VENT MB='YMIN' SURF_ID='OPEN' /
&VENT MB='YMAX' SURF_ID='OPEN' /

With this all x and y surfaces are open. Wherever there is a hole the smoke/heat transfers.

Now we have made a basic model for our FDS. However we do not see any data yet, but it is easy to fix and you have free hands where, what data you want to collect.





4 Analyzing

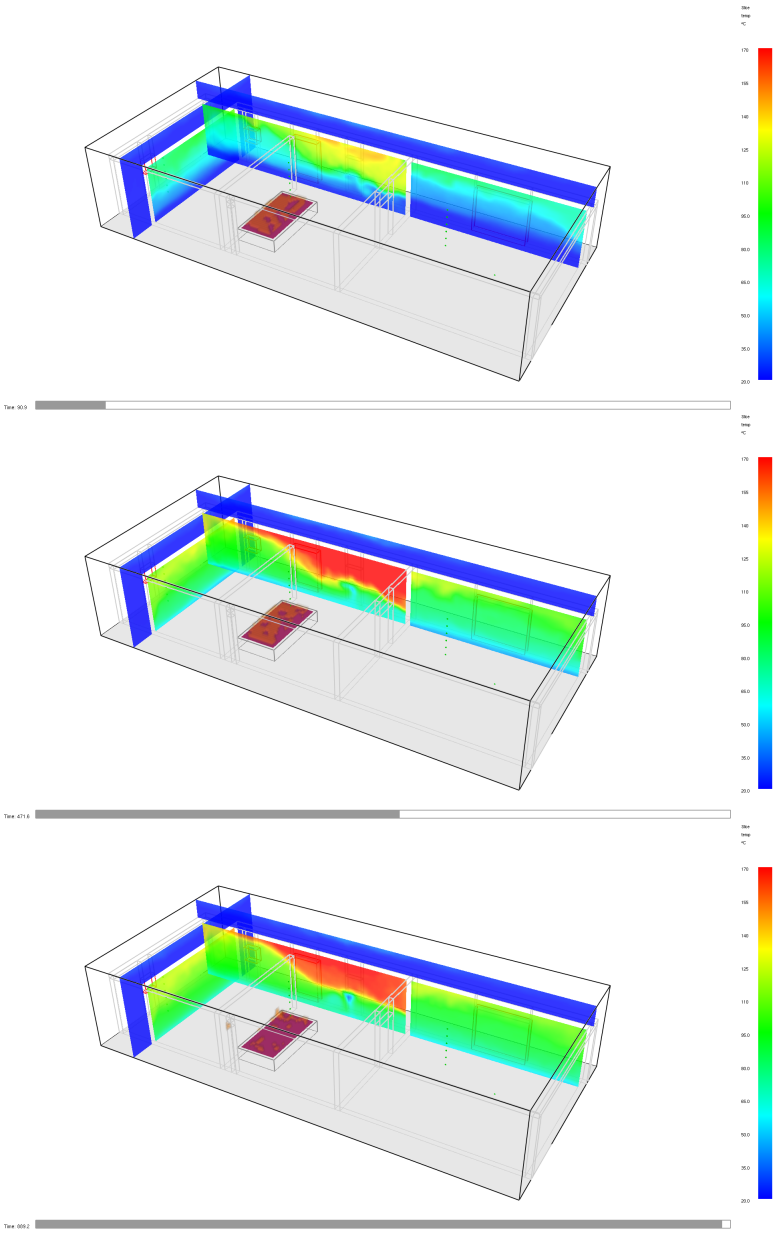
The BNDF (boundary file) namelist group parameters allows you to record surface quantities at all solid obstructions. As with the slice (SLCF) group, each quantity is prescribed with a separate BNDF line, and the output files are of the form CHID_n.bf. No physical coordinates need be specified, however, just QUANTITY. In our example we will observe temperature, wall temperature, burning rate, gauge heat flux (amount of energy that would be absorbed if the surface were cold), HRRPUV (Heat Release Rate Per Unit Volume), mixture fraction (describe combustion).
Following BNDF were calculated

&BNDF QUANTITY='GAUGE HEAT FLUX' /
&BNDF QUANTITY='WALL TEMPERATURE' /
&BNDF QUANTITY='BURNING RATE' /

Now we define different slices (SLCF). There we need to tell from which axis X,Y or Z we want a slice.

&SLCF PBX=15.00, QUANTITY='TEMPERATURE' /
&SLCF PBX=5.00, QUANTITY='TEMPERATURE' /
&SLCF PBX=11.00, QUANTITY='TEMPERATURE' /
&SLCF PBX=15.00, QUANTITY='HRRPUV' / Heat Release Rate per Unit Volume
&SLCF PBX=11.00, QUANTITY='HRRPUV' / Heat Release Rate per Unit Volume
&SLCF PBX=5.00, QUANTITY='HRRPUV' / Heat Release Rate per Unit Volume
&SLCF PBX=15.00, QUANTITY='MIXTURE FRACTION' /
&SLCF PBX=11.00, QUANTITY='MIXTURE FRACTION' /
&SLCF PBX=5.00, QUANTITY='MIXTURE FRACTION' /
&SLCF PBY=3.45, QUANTITY='TEMPERATURE' /
&SLCF PBY=0.50, QUANTITY='TEMPERATURE' /
&SLCF PBY=3.45, QUANTITY='HRRPUV' / Heat Release Rate per Unit Volume
&SLCF PBY=0.50, QUANTITY='HRRPUV' / Heat Release Rate per Unit Volume
&SLCF PBY=3.45, QUANTITY='MIXTURE FRACTION' /
&SLCF PBY=0.50, QUANTITY='MIXTURE FRACTION' /

Below we can see three different temperatures in different times.



If there is points what is interesting to observe then these can be defined using DEVC (device). In different cases they can be used for example as smoke alarms, springers activators.

&DEVC XYZ=11.0,2.3,2.1, QUANTITY='TEMPERATURE' /
&DEVC XYZ=11.0,2.3,1.8, QUANTITY='TEMPERATURE' /
&DEVC XYZ=11.0,2.3,1.5, QUANTITY='TEMPERATURE' /

&DEVC XYZ=11.0,2.3,1.2, QUANTITY='TEMPERATURE' /
&DEVC XYZ=11.0,2.3,0.9, QUANTITY='TEMPERATURE' /
&DEVC XYZ=11.0,2.3,0.6, QUANTITY='TEMPERATURE' /

5 Parallel computing

Parallel Processing FDS employs OpenMP [4], a programming interface that exploits multiple processing units on a single computer. From version 6.1 FDS will employ OpenMP by default. For clusters of computers, FDS employs Message Passing Interface (MPI) [6]. Here, the computational domain must be divided into multiple meshes and typically each mesh is assigned its own process. These processes can be limited to a single computer, or they can be distributed over a network.
To run FDS in parallel using MPI (Message Passing Interface), you must break up the computational domain into multiple meshes so that the workload can be divided among the computers. Direct instructions to run can be hard as parallel computers can be configured very differently. However there are quite many instructions available in the web and guides. For example FDS user manual [1] has thorough explanations, wikibooks [2] has Fedora Linux and Ubuntu Linux instructions.
In this example code was run on the high performance computing center of consortium of University of Tartu - Rocket [7]. Same code was run in 5 times. In each case different size of meshes were used. For each mesh one node was assigned. MPI was used for a cluster and in each node 4 OpenMP threads were used. Following code is one example how to break mesh into four parts.

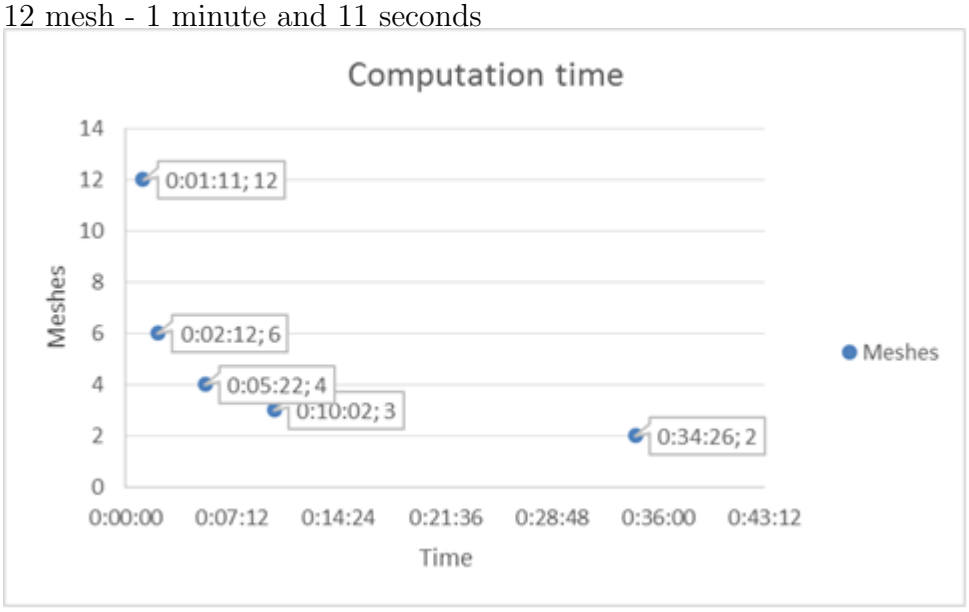
&MESH ID='mesh1' IJK=84,36,36, XB=0.0,8.4,-0.8,2.8,0.0,3.6, MPI_PROCESS=0/
&MESH ID='mesh2' IJK=84,36,36, XB=0.0,8.4,-0.8,2.8,0.0,3.6, MPI_PROCESS=1/
&MESH ID='mesh3' IJK=84,36,36, XB=8.4,16.8,2.8,6.4,0.0,3.6, MPI_PROCESS=2/
&MESH ID='mesh4' IJK=84,36,36, XB=8.4,16.8,2.8,6.4,0.0,3.6, MPI_PROCESS=3/

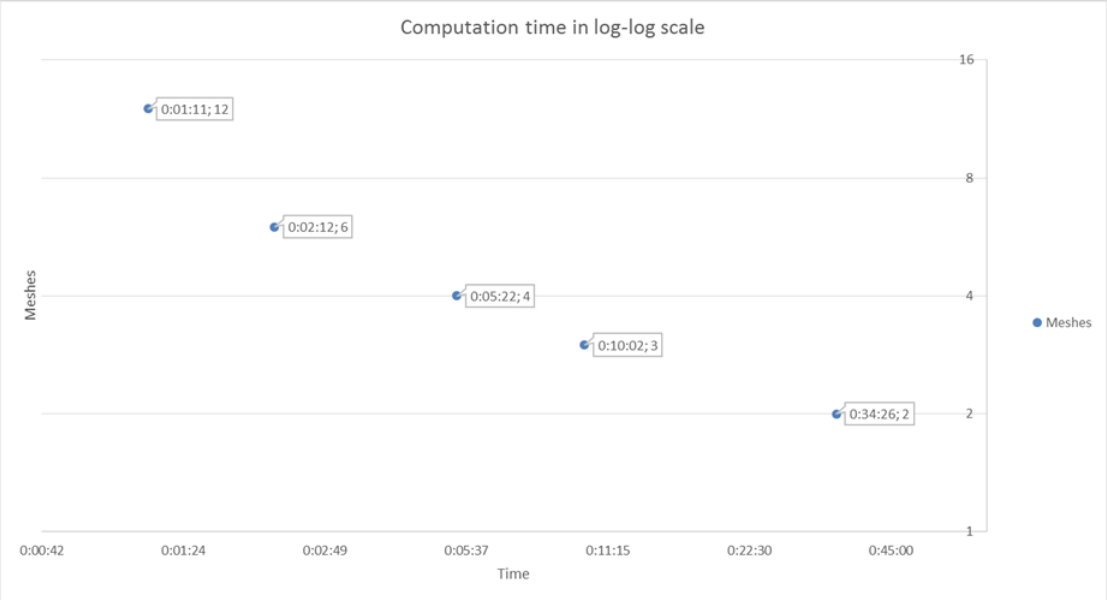
I run this project in 3 different mesh sizes. It is recommended to do a more with different sizes to analyze efficiency of parallel computing, so further investigation is needed.

6 mesh - 24 minutes and 20 seconds
12 mesh - 15 minutes and 12 seconds
24 mesh - 8 minutes and 24 seconds

Below we can see time it took to run simulation with a small problem size.

2 mesh - 34 minute and 26 seconds
3 mesh - 10 minute and 2 seconds
4 mesh - 5 minute and 22 seconds
6 mesh - 2 minute and 12 seconds
12 mesh - 1 minute and 11 seconds





6 Acknowledgment

Would like to acknowledge Benson Muite for his good suggestions. Project is sponsored by IT Akadeemia [8].

References

[1] FDS User Guide `1)https://drive.google.com/file/d/0B4528gcI9D3HUENXVXBWSTN1eWc/view` hecked on 15 January 2017

[2] FDS wikibooks `https://en.wikibooks.org/wiki/Fire_Simulation_for_Engineers/FDS` Checked on 15 January 2017

[3] FDS download from official homepage `https://pages.nist.gov/fds-smv/downloads.html` Checked on 15 January 2017

[4] B. Chapman, G. Jost, and R. van der Pas. Using OpenMP Portable Shared Memory Parallel Programming. MIT Press, Cambridge, Massachusetts, 2007.

[5] FDS github `https://github.com/firemodels/fds/wiki/Installing-and-Running-FDS-on-a-Windows-PCl` Checked on 15 January 2017

[6] 6) W. Gropp, E. Lusk, and A. Skjellum. Using MPI Portable Parallel Programming with the MessagePassing Interface. MIT Press, Cambridge, Massachusetts, 2nd edition, 1999.

[7] UT Rocket `http://www.hpc.ut.ee/rocket_cluster` Checked on 15 January 2017

[8] IT Akadeemia `http://www.hitsa.ee/ikt-haridus/ita` Checked on 15 January 2017

[9] Project code in github `https://github.com/raunonaksi/FDS-introduction`

[10] Properies values `https://arxiv.org/ftp/arxiv/papers/1404/1404.6774.pdf`

[11] James G. Quintiere, Fundamentals of Fire Phenomena, Wiley, 2011