Suggested answers in exercise on turbulence

This is not a complete answer to the exercise, - merely a calculation of the values and some discussion.

Three different sizes were used for the simulation, including a super fine grid. This grid was not included in the assignment due to the long calculation time, about 12 days. The coarse grid is 64, 24, 24, the fine grid is 128, 48, 48 and the super fine grid is 256, 96, 96.

Table 1 Grid used in simulation, D/dx ratio and calculation time*

Grid	**Grid size [cm]**	**Grid cells**	**Total number of cells**	**D*/dx**	**Calculation time using 4 OpenMP threads (using 2 threads) [hours]**
Coarse	10	64 x 24 x 24	36,864	3.8	0.89 (1.29)
Fine	5	128 x 48 x 48	294,912	7.6	15.6 (17.97)
Super fine	2.5	256 x 96 x 96	2,359,296	15.2	286 (-)

It can be seen in table 1 that the ratio D*/dx is varying between 3.8 for the coarse grid to 15.2 for the fine grid. This very close to the recommended values of D*/dx, which should be between 4 for a poor resolution of the turbulence and 16 for a good resolution of the turbulence (please keep in mind Kevin McGrattan's comment about these recommendations, or rather his advice not to use them).

Further, it can be seen from the last column that the ratio between the calculations time is approximately 18 when using all four core in the computer utilizing OpenMp. This is slightly above the expected value of 16, which is due to the doubling the grid cells in each direction and halving the time step to keep the same Courant number in the simulation. That would theoretical would result in 2^3 x 2 = 16 time longer calculation time when halving the grid size. Using only 2 of the available cores (the number in parenthesis) on the pc scales better with a difference of about a factor of 14 (17.97/1.29), but is overall slower as less cores are used.

It can been seen in Figure 1 and in Figure 2 and Figure 3 that the velocities in particular for the door opening takes some time to reach the steady state value.

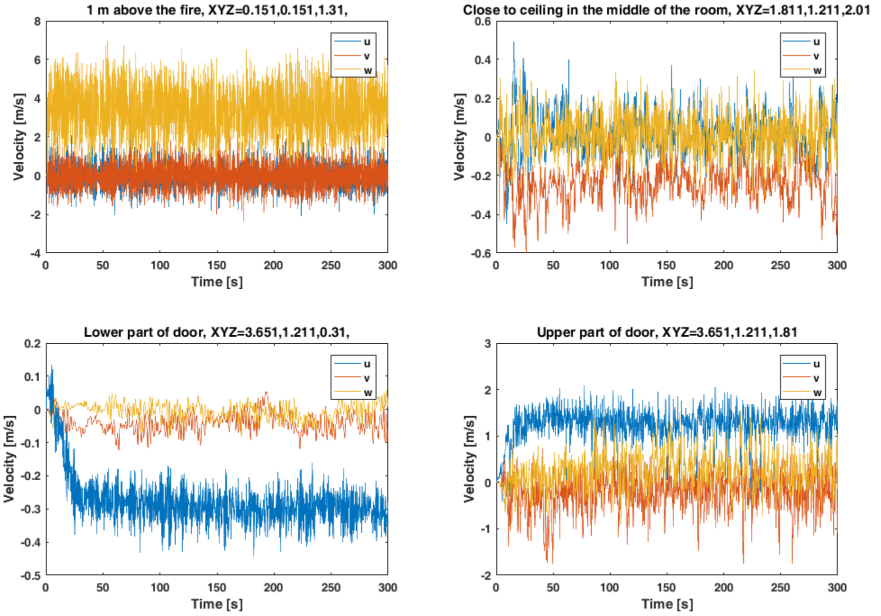

Figure 1 Velocity at four different point for coarse grid

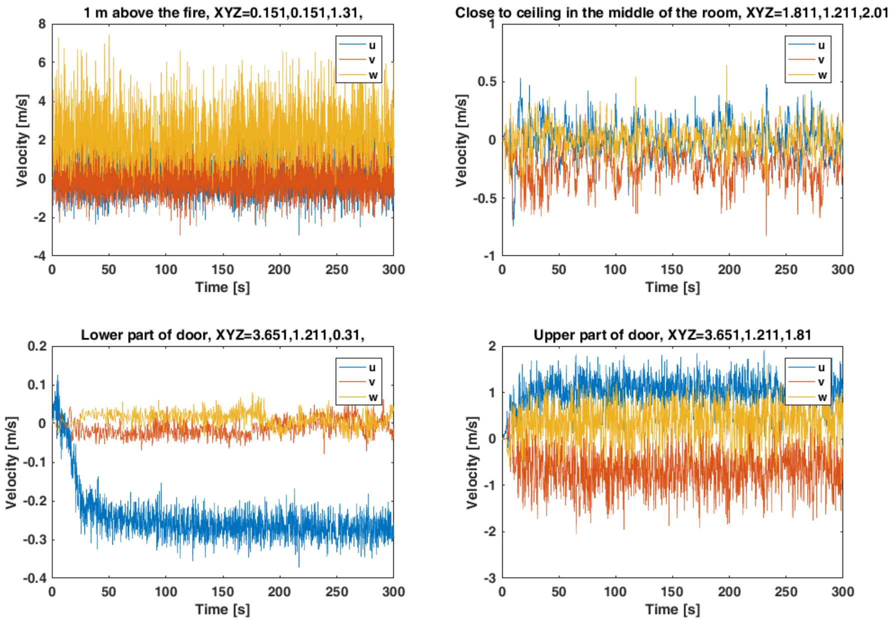

Figure 2 Velocity at four different point for fine grid

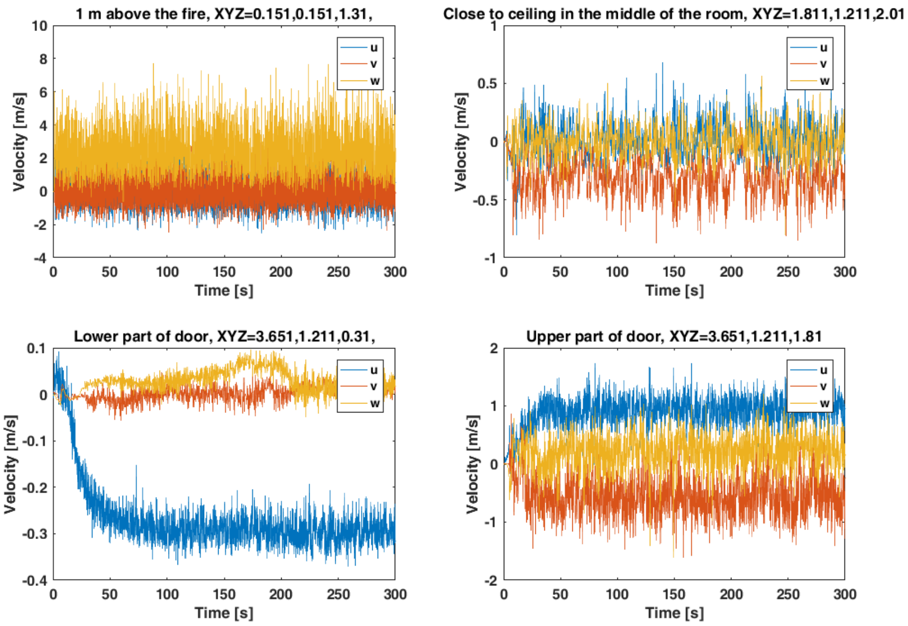

Figure 3 Velocity at four different point for super fine grid

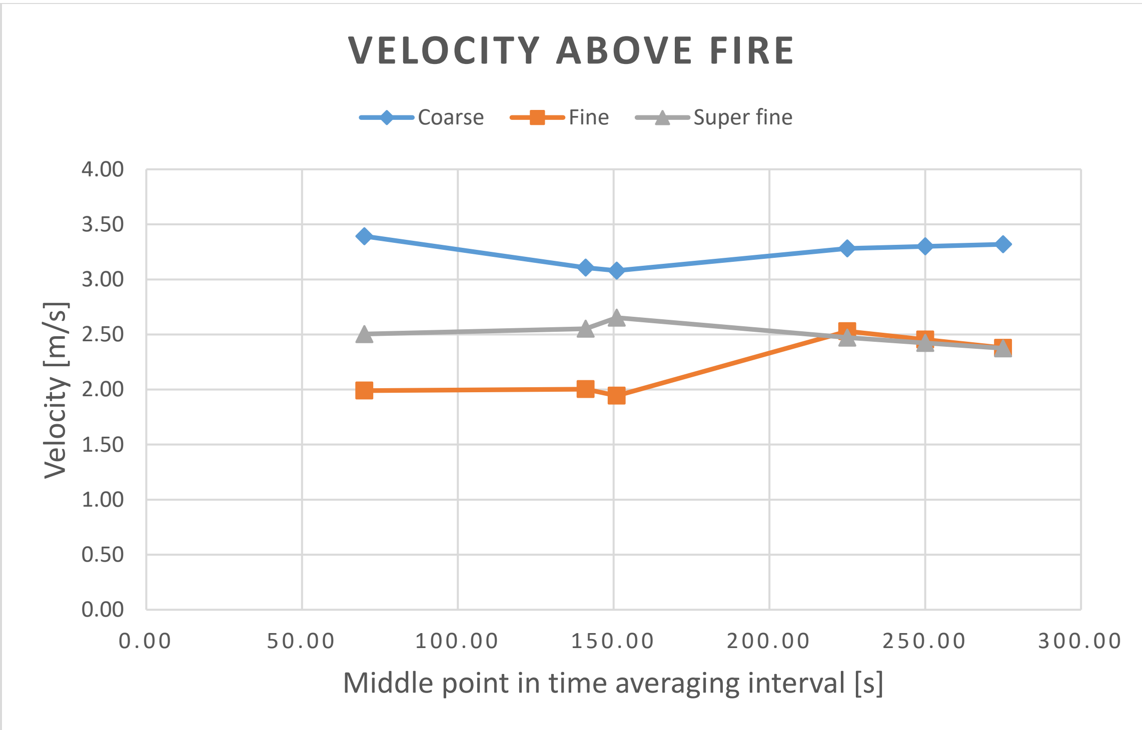

Figure 4 Velocity at four different point for fine grid

In order to evaluate the turbulent properties they need to have stabilised at steady state level. Looking at Figure 4, which shows the average calculated velocity above the fire is seems that values have first stabilised after 200 seconds were the curve for the fine grid and the super fine grid converges at 225 seconds. Further, it can be seen in figure 1 to 3 that for the lower door opening these that the v and w velocities converges about at the same time. For the coarse grid this happens at about 150 s, the fine grid at 190 s and the super fine grid at 220 s. So refinement of the grid leads to longer time for steady state conditions.

The values have been averaged from 200 seconds to 300 seconds.

Table 2 Mean velocity [m/s]

Grid	**Above fire**	**Middle of room**	**Lower door**	**Upper door**
Coarse	3.30	0.26	0.31	1.42
Fine	2.45	0.29	0.27	1.37
Super fine	2.42	0.37	0.30	1.15

Table 3 Turbulent intensity [%]

Grid	**Above fire**	**Middle of room**	**Lower door**	**Upper door**
Coarse	24.0	38.4	10.6	26.6
Fine	34.9	47.4	8.3	24.2
Super fine	36.2	40.2	6.3	21.9

Table 4 Turbulent intensity including sub-grid turbulent kinetic energy [%]

Grid	**Above fire**	**Middle of room**	**Lower door**	**Upper door**
Coarse	28.2	40.9	11.6	31.2
Fine	38.2	49.7	8.9	26.3
Super fine	38.2	41.7	6.7	23.1

Table 5 Measure of turbulent resolution [-]

Grid	**Above fire**	**Middle of room**	**Lower door**	**Upper door**
Coarse	0.28	0.12	0.16	0.27
Fine	0.17	0.09	0.13	0.15
Super fine	0.10	0.07	0.10	0.10

Table 6 Measure of total turbulent energy [m^2/s^2]

Grid	**Above fire**	**Middle of room**	**Lower door**	**Upper door**
Coarse	1.30	0.02	0.0019	0.29
Fine	1.32	0.03	0.0009	0.19
Super fine	1.29	0.04	0.0006	0.11

It can be seen in Table 2 that the velocity above the fire is dependent on the grid size. A finer grid gives a lower average velocity due to the increase mixing (entrainment) in the fire plume. For the position in the middle of the room and in the door opening the values are very close to each other.

The turbulent intensity increased with finer grid for the position above the fire and in the middle of the room (Table 3). For the door opening the intensity decreases. The overall values are very high with a level of more than 20% for all positions, but the lower door position. The reason for the low intensity at the lower door position is that there is no/little disturbance in the air outside the room, where the air is drawn from for the lower door position.

Including the sub-grid turbulent kinetic energy in the calculation of the turbulent intensity increased the intensity with about 2-4 percentage points as seen when comparing Table 3 and Table 4.

The measure of turbulent resolution is above the recommended value of 0.2 above the fire and in the upper door opening as seen in Table 5 for the coarse grid. For the fine grid the levels at all 4 positions are below the recommended value of 0.2.

Further it is interesting to note that the total turbulent energy changes with grid size as seen in Table 6. Above the fire it is the same, but for the middle of the room the total energy is very low and it increases with the fine grid to about the double value. For the upper door opening the total turbulent energy is higher than in the middle of the room and the level decreases with finer grid size. These energy level correlates well with the velocity in the same position, - high velocity and leads to higher level of turbulent energy. This is also expected.

These values have been evaluated in Matlab. The csv files have been imported into matlab using an auto generated script, as seen on page 8 and page 9. The interesting part is shown on page 10, where the graphs are printed and the values are calculated.

FFT Analysis

The python script provided by Lukas Arnold was used for the calculation.

In Figure 5 the distribution of energy of the frequencies are shown for the point 11, which Is the W velocity 1 m above the fire. The data should ideally be sampled with a fixed time step, so that could be the reason for the scatter in the graphs. The main features are shown, - the finer grid contains more energy at the higher frequencies, which is also expected.

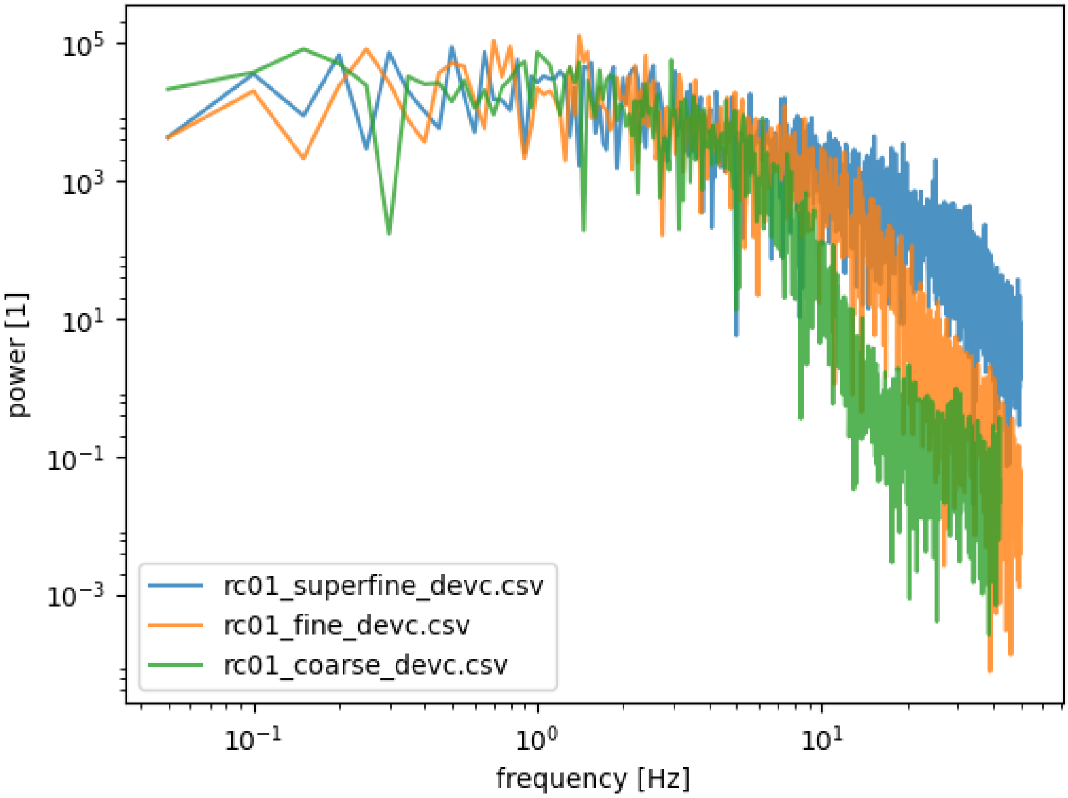

Figure 5 Spectrum of turbulent energy for all three grid sizes

Also interesting to note is the distribution of the fluctuations as seen in Figure 6 . Finer grid give more variation in the velocity.

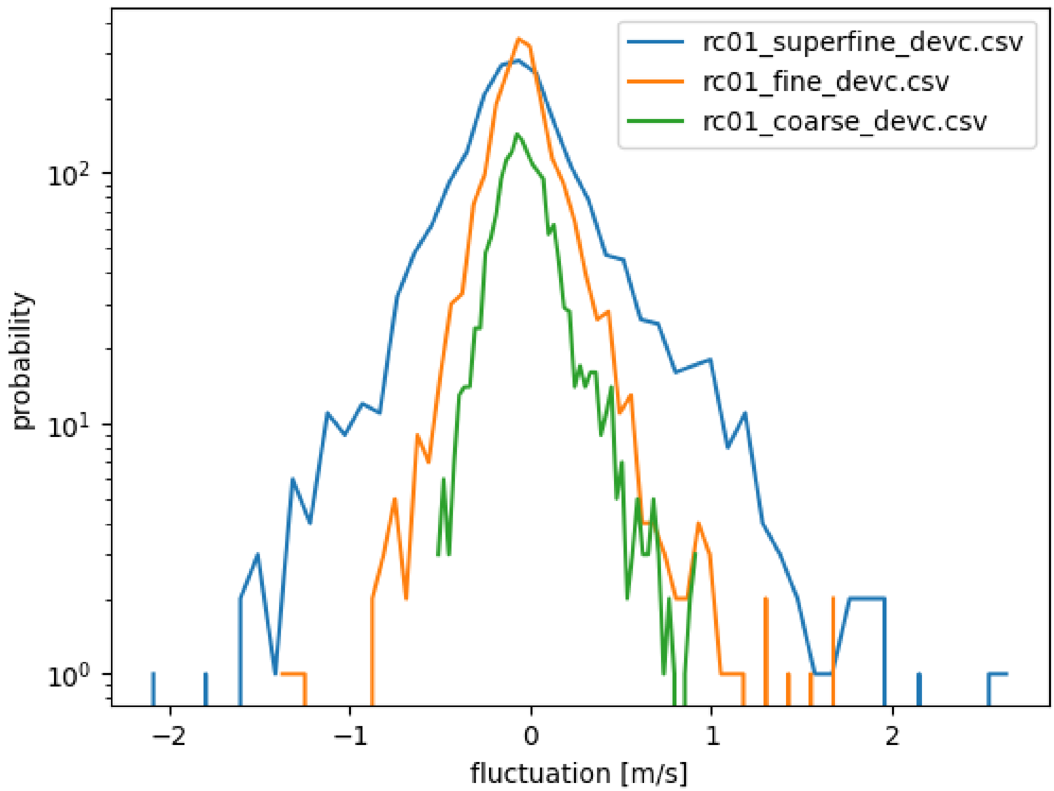

Figure 6 Fluctuations and their probability

```
%% Import data from text file.
% Script for importing data from the following text file:
%
%    D:\firemodels\FDS6\calc\rc01_2016_mean\rc01_answer_devc.csv
%
% To extend the code to different selected data or a different text file,
% generate a function instead of a script.

% Auto-generated by MATLAB on 2016/12/09 11:11:45

%% Initialize variables.
%filename ='D:\firemodels\FDS6\calc\rc01_2016_mean\rc01_answer_devc.csv'; %
coarse grid
filename =
'D:\firemodels\FDS6\calc\rc01_2016_mean_fine\rc01_fine_answer_devc.csv'; %
fine grid
delimiter = ',';
startRow = 3;

%% Format string for each line of text:
%   column1: double (%f)
%   column2: double (%f)
%   column3: double (%f)
%   column4: double (%f)
%   column5: double (%f)
%   column6: double (%f)
%   column7: double (%f)
%   column8: double (%f)
%   column9: double (%f)
%   column10: double (%f)
%   column11: double (%f)
%   column12: double (%f)
%   column13: double (%f)
%   column14: double (%f)
%   column15: double (%f)
%   column16: double (%f)
%   column17: double (%f)
%   column18: double (%f)
%   column19: double (%f)
%   column20: double (%f)
%   column21: double (%f)
%   column22: double (%f)
%   column23: double (%f)
%   column24: double (%f)
%   column25: double (%f)
%   column26: double (%f)
%   column27: double (%f)
%   column28: double (%f)
%   column29: double (%f)
% For more information, see the TEXTSCAN documentation.
formatSpec =
'%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%f%*s%*s%*s%*s%[^\
n\r]';

%% Open the text file.
fileID = fopen(filename,'r');

%% Read columns of data according to format string.
% This call is based on the structure of the file used to generate this
% code. If an error occurs for a different file, try regenerating the code
```

```
% from the Import Tool.
dataArray = textscan(fileID, formatSpec, 'Delimiter', delimiter,
'EmptyValue' ,NaN,'HeaderLines' ,startRow-1, 'ReturnOnError', false);

%% Close the text file.
fclose(fileID);

%% Post processing for unimportable data.
% No unimportable data rules were applied during the import, so no post
% processing code is included. To generate code which works for
% unimportable data, select unimportable cells in a file and regenerate the
% script.

%% Allocate imported array to column variable names
Time = dataArray{:, 1};
T1 = dataArray{:, 2};
T2 = dataArray{:, 3};
T3 = dataArray{:, 4};
T4 = dataArray{:, 5};
SKE1 = dataArray{:, 6};
SKE2 = dataArray{:, 7};
SKE3 = dataArray{:, 8};
SKE4 = dataArray{:, 9};
U1 = dataArray{:, 10};
V1 = dataArray{:, 11};
W1 = dataArray{:, 12};
U2 = dataArray{:, 13};
V2 = dataArray{:, 14};
W2 = dataArray{:, 15};
U3 = dataArray{:, 16};
V3 = dataArray{:, 17};
W3 = dataArray{:, 18};
U4 = dataArray{:, 19};
V4 = dataArray{:, 20};
W4 = dataArray{:, 21};
U1rms = dataArray{:, 22};
V1rms = dataArray{:, 23};
W1rms = dataArray{:, 24};
U2rms = dataArray{:, 25};
V2rms = dataArray{:, 26};
W2rms = dataArray{:, 27};
U3rms = dataArray{:, 28};
V3rms = dataArray{:, 29};


%% Clear temporary variables
clearvars filename delimiter startRow formatSpec fileID dataArray ans;

%%
```

```
%
% Plot the velocity over time
%

for i=1:4
    subplot(2,2,i), plot(Time, eval(['U' num2str(i)]),Time, eval(['V'
num2str(i)]), Time,eval(['W' num2str(i)]))
    xlabel('Time [s]')
    ylabel('Velocity [m/s]')
    legend('u', 'v', 'w')
    switch i
        case 1
          titlename = '1 m above the fire, XYZ=0.151,0.151,1.31,';
        case 2
          titlename = 'Close to ceiling in the middle of the room,
XYZ=1.811,1.211,2.01';
        case 3
          titlename = 'Lower part of door, XYZ=3.651,1.211,0.31,';
        case 4
          titlename = 'Upper part of door, XYZ=3.651,1.211,1.81';
    end
    title(titlename)
end


%
%  Calculate turbulent intensity from 200 s to 300 s
%
mean_velocity=zeros(1,4);
intensity=zeros(1,4);
total_turbulent_energy =zeros(1,4);
turbulent_resolution=zeros(1,4);

mst=find(Time>200,1, 'first' ); % minimum time step, the first time step
after 200 seconds
for i=1:4   % Step through the 4 positions
    u=eval(['U' num2str(i) '(' num2str(mst) ':end)']); % u for position i
    v=eval(['V' num2str(i) '(' num2str(mst) ':end)']); % v for position i
    w=eval(['W' num2str(i) '(' num2str(mst) ':end)']); % w for position i
    ske = eval(['SKE' num2str(i) '(' num2str(mst) ':end)']);% ske pos. i
    velocity = sqrt(u.^2 + v.^2 + w.^2); % Calculate the velocity vector
    mean_velocity(i)=mean(velocity) ; % Average velocity for position i
    intensity(i)=sqrt(1/3 * (var(u)+var(v)+var(w)))/mean_velocity(i)*100;
    total_turbulent_energy(i) = (1/2 * (var(u)+var(v)+var(w))+ mean(ske));
% [m^2/s^2]
    turbulent_resolution(i)= mean(ske)/total_turbulent_energy(i);
end
% display calculated values on screeen
display(mean_velocity)
display(intensity)
display(turbulent_resolution)
display(total_turbulent_energy)
```