# Computational Fluid Dynamics as an advanced module of ESP-r Part 1: The numerical grid - defining resources and accuracy

**Jordan A. Denev**

**Abstract: The present paper is a first one from a series of papers oriented to engineers who are not experienced in the field of the Computational Fluid Dynamics (CFD). The paper considers the important process of constructing the numerical grid. Four rules (constrains) are formulated which allow engineers to construct qualitative numerical grids. The rules are explained and demonstrated on an example of a ventilated room using the ESP-r (Energy Simulation Program - research) software**

## Introduction

In the world of complex modern buildings the integrated modelling becomes an important tool for optimising the work environment. Tools like the ESP-r software offer modelling capabilities at different levels. The level considered in the following concerns the Computational Fluid Dynamics (CFD) module of ESP-r. Using this module engineers can study thermal comfort in air-conditioned rooms, can define pressure distribution on the facades of buildings.

CFD remains a complex area which requires from the users the knowledge of a considerable amount of input parameters. Those input parameters should be well balanced between each other in order to achieve reliable and accurate results. This paper targets the simple explanation of important rules during the grid generation process.

All rules and examples in the paper are (to a big extent) general and hence are valid to the majority of today's CFD-tools for analysing air-conditioned rooms.

The paper is organised as follows. First we define the role of creating the computational grid within the overall modelling process. In order to achieve both accurate and quick solutions we introduce in section 2 four gridding rules. On a worked example in section 3 we demonstrate how to carry out the gridding process based on the four rules. Finally we consider the question how fine should be the numerical grids for engineering computations.

## 1. The role of the gridding process in room airflow modelling

The gridding process is the most time-consuming process in modelling. It is usually decisive:

- for the accuracy of the results;
- for the convergence behaviour of the computations (therefore also for the CPU-time of the computations).

What makes gridding complicated is that it is connected to many other aspects of the numerical modelling. For instance, turbulence models of the low-Reynolds-number type require very fine grids near walls, while boundary conditions of the wall-functions-type should keep the opposite (and not so easy-to-check) constraint of "positioning the first numerical point outside the viscous sub-layer". Very often such requirements are not very informative and therefore of a little help for the non-experienced user.

What further makes gridding complicated is that it requires some preliminary knowledge about the expected solution. Simply speaking - regions of large gradients of velocities and/or temperatures should be covered by (as much as possible) finer grids. And finally, let us note that many gridding rules are contradictory, making from the whole gridding process a complicated optimisation problem.

## 2. Four rules for creating numerical grids

To make a grid means to divide the physical space of the room into smaller regular volumes which are of a parallelogram shape – they are called "control volumes" or "grid cells". The target of gridding is to define the coordinates of the surfaces/faces of the control volumes (when projected on two-dimensional plots the surfaces/faces become "grid lines", see Fig. 1).

The rules we will consider in the following are not always based on stringent theory; some of them simply had proven themselves useful in practice. Our further considerations will confine here to orthogonal structured (i.e. regular) grids as they are best suitable and widely used for room airflow problems. Non-orthogonal grids and their specific features are presented e.g. in Peric (1985) and Denev and Stankov (2000).

*Definition: "grid aspect ratio"* is the ratio of the *x*-distance (similar for $\Delta y$ or $\Delta z$) of two neighbour control volumes: $\Delta x_i / \Delta x_{i+1}$ (Fig. 1).

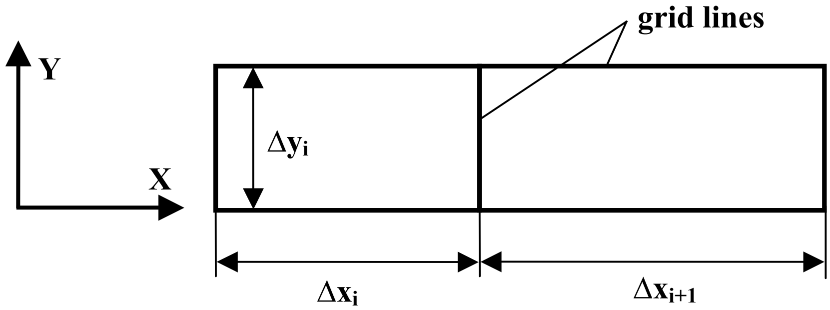


**Fig. 1. Two neighbour control volumes (grid cells) projected on the *XY* plane**

*First rule:* the grid aspect ratio should be kept possibly below *1.2* (or $1.2^{-1}$ - if the grid decreases in size) for all spatial directions (*x, y* or *z*). If it is not possible to keep the ratio below 1.2 one should try not to exceed the ratio *1.4* (or $1.4^{-1}$).

*Second rule:* The largest and the smallest size of any two grid cells in one spatial direction should not exceed the ratio *10.* If this is not possible, one should try not to exceed the ratio *20* (see also FLOVENT on-line documentation, 2001).

*Example:* assume that in Fig. 1 control volume *No i* has a size in *x*-direction which is equal to $\Delta x_i$ = *0.36 [m]*. In order to keep the first rule neither of its neighbour cells (*No i-1* and *No i+1*) should be bigger than *0.432 [m]* or less than *0.30 [m].* Assume that control volume *No i+1* is the largest one between all volumes in x-direction (let $\Delta x_{i+1}$=*0.40[m]*) - then according to the second rule no one of the control volumes in the domain should have its $\Delta x$-size ($\Delta x$-distance) smaller than *0.04 [m].*

*Definition: "cell aspect ratio"* is the ratio between any two from the three dimensions of a single grid cell: $\Delta x_i / \Delta y_i$ (see Fig. 1) or $\Delta x_i / \Delta z_i$ or $\Delta y_i / \Delta z_i$ .

*Third rule:* the cell aspect ratio for each one control volume should be kept below *10*; in case that this is not possible one should try to keep the ratio below *20*.

*Example:* Let us consider again control volume *No i (Fig.1)* which has e.g. the size $\Delta x_i$ = *0.36 [m]*. According to the third rule for this control volume we have:

*0.036 [m]* < $\Delta y_i$ < *3.6 [m]* and *0.036 [m]* < $\Delta z_i$ < *3.6 [m].*

In a three-dimensional grid all combinations between the cell distances exist (each $\Delta x_i$ is combined with each $\Delta y_j$ and with each $\Delta z_k$ ). This automatically means that the smallest dimension (between all control volumes) and the largest grid-cell dimension (between all control volumes) are restricted by the ratio of *10*. This could be just another formulation of the third rule.

The first three rules guarantee that the resulting system of linear algebraic equations will stably converge within the iterative solver. The "ideal" numerical grid is consisting of cubical control volumes of equal size; the three rules actually do not allow the real grid to deviate much from this "ideal" situation. However, such grids with almost cubical and equal-size control volumes are usually very fine. This refinement is in some contradiction with the fourth rule.

*Fourth rule*: - the overall number of control volumes should be well controlled so that the CPU-time for one simulation does not exceed *1-2* days.

This fourth rule comes from the understanding that modelling usually requires repeated solutions with parameter variations. We would recommend as a good practice to solve the problem first on a very coarse numerical grid, e.g. below *10 000* grid points (please note, that on such preliminary grids it is difficult and not necessary to keep exactly the above given rules). On such a grid the user could make a series of quick runs in order to test the consistency of all other parts of the modelling process - boundary conditions, numerical parameters and physical models. The additional time spend to create an additional coarse grid usually is paid back due to shortening the overall time of the modelling process. How to proceed after this and create the fine grids we comment in the section 4.

## 3. A worked example - gridding a room with the ESP-r

The process of constructing numerical grids according to the above rules will be demonstrated on the example of the room from Figure 2 (the geometry is presented by the graphical tool of ESP-r). It is an example of an office room with dimensions *4 x 4 x 3 [m]*. An inflow opening supplies fresh air from the back wall; the air leaves the room trough the outflow opening in the lower part of the door.

As a first step we choose the origin of the grid and direction of each axis – e.g. let the origin of the Cartesian coordinate system be at vertex *"v3"* and the *x*-axis face vortex *"v4"*, the *y*-axis face vortex *"v2"* and the *z* axes be directed vertically upwards (see Fig. 2).

Now the gridding process can start independently for every spatial direction. However, prescribing the coordinate of each gridding line manually is quite a difficult task. Usually the process starts with finding out the coordinates of the most important grid lines. Those important grid lines form the boundaries of the grid regions; all other grid lines are then calculated by the gridding software using the coordinates of the regions. By this process we just force some of the grid lines to pass exactly trough the boundaries of the regions; the other grid lines remain more "free". When the final grid is created there is no difference between the grid lines at the boundaries of regions and the regular grid lines.

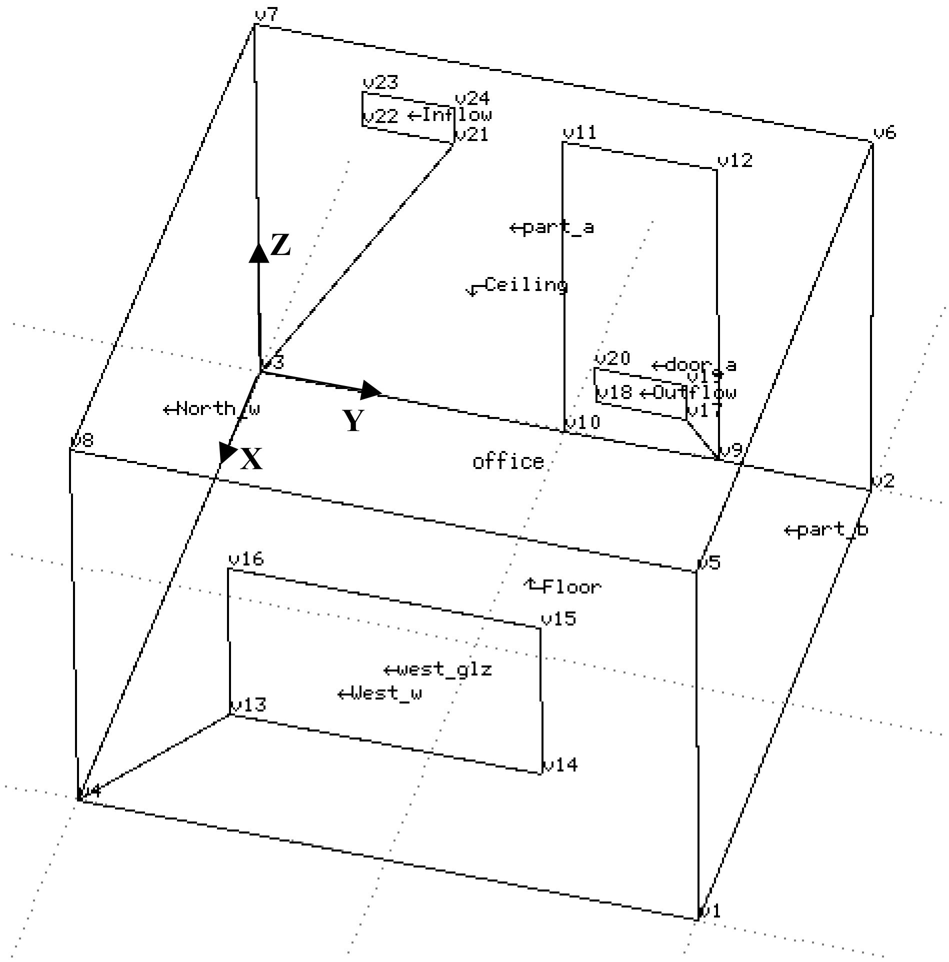


**Fig. 2. Geometry of the office room with the window, the door and the two ventilation openings. Vertex numbers and names of the surfaces are also shown**

Generally regions are formed by the boundaries of objects inside the room (e.g. computers or furniture) – on the surfaces/edges of such objects we need to position the corresponding grid lines. Other regions are due to boundary conditions. In this case objects at the surrounding walls (as e.g. windows) have different boundary conditions (e.g. temperature) and therefore grid lines should pass exactly their edges.

Along the *x*-axis we divide the grid in three regions/sections (see also Fig.6). In the example considered here there is another natural choice for the first region in *x*-direction: this is the area close to the wall with the inflow-jet (*x=0 [m]*). In that area the grid should be fine enough to catch the large gradients of the jet close to the inflow opening. As the largest gradients are close to the inflow opening we would like the grid step to enlarge with increasing the distance from the wall. The length of the first region depends on the jet development (i.e. on the solution), but probably a good choice is a length of *0.5 [m]*. The second section (region) is the largest one and is positioned in the middle of the room. The third section is near the wall with the window – the grid there should again be refined (decreasing the grid step) to catch the gradients near this outer wall of the room. We choose the length of this region to be again *0.5 [m]*.

Associating appropriate number of grid points (denoted by “cells”) in each region leads to the menus given in Fig. 3 – the resulting *x*-grid (variant 1) has now in total *16* grid cells.

```
x-axis gridding

 Number of regions:    3
 Total domain length:       4.000
 Current defined length:    4.000
 ------------------------------------
   Region  Cells  Length  P-law coeff  sym
a      1      4    0.500    1.200       no
b      2      9    3.000    1.000       no
c      3      3    0.500    0.800       no
 ------------------------------------
+ add/delete region
? help
- exit
```

**a) variant 1**

```
x-axis gridding

 Number of regions:    3
 Total domain length:       4.000
 Current defined length:    4.000
 ------------------------------------
   Region  Cells  Length  P-law coeff  sym
a      1      4    0.550    1.140       no
b      2     13    2.950    1.140      yes
c      3      3    0.500    0.950       no
 ------------------------------------
+ add/delete region
? help
- exit
```

**b) variant 2**

**Fig. 3. The menu for building regions along the *x*-axis. Note that the power-law coefficient with a value of 1.2 (denoted "P-law coeff" in the menu) gives results which are similar but not the same as a "grid aspect ratio" equal to 1.2**

Within each region the user can control the grid aspect ratio by means of the power-law coefficient – values larger than 1.0 create grids with consequently increasing cell-sizes and vice-versa. With this approach problems usually are encountered at boundaries between two regions. Therefore there is a tool which allows to monitor the grid aspect ratio - the screen output from this tool is given in figure 4 (variant 1).

Grid in the X axis...

| Cell index, | X start & end, | | distance, | aspect ratio |
|---|---|---|---|---|
| 1 | 0.0000 | 0.0947 | 0.0947 | 1.0000 |
| 2 | 0.0947 | 0.2176 | 0.1229 | 1.2974 |
| 3 | 0.2176 | 0.3540 | 0.1364 | 1.1098 |
| 4 | 0.3540 | 0.5000 | 0.1460 | 1.0702 |
| 5 | 0.5000 | 0.8333 | 0.3333 | 2.2836 |
| 6 | 0.8333 | 1.1667 | 0.3333 | 1.0000 |
| 7 | 1.1667 | 1.5000 | 0.3333 | 1.0000 |
| 8 | 1.5000 | 1.8333 | 0.3333 | 1.0000 |
| 9 | 1.8333 | 2.1667 | 0.3333 | 1.0000 |
| 10 | 2.1667 | 2.5000 | 0.3333 | 1.0000 |
| 11 | 2.5000 | 2.8333 | 0.3333 | 1.0000 |
| 12 | 2.8333 | 3.1667 | 0.3333 | 1.0000 |
| 13 | 3.1667 | 3.5000 | 0.3333 | 1.0000 |
| 14 | 3.5000 | 3.7076 | 0.2076 | 1.6055 |
| 15 | 3.7076 | 3.8615 | 0.1539 | 1.3493 |
| 16 | 3.8615 | 4.0000 | 0.1385 | 1.1109 |

**a) variant 1**

Grid in the X axis...

| Cell index, | X start & end, | | distance, | aspect ratio |
|---|---|---|---|---|
| 1 | 0.0000 | 0.1132 | 0.1132 | 1.0000 |
| 2 | 0.1132 | 0.2496 | 0.1363 | 1.2038 |
| 3 | 0.2496 | 0.3962 | 0.1466 | 1.0757 |
| 4 | 0.3962 | 0.5500 | 0.1538 | 1.0487 |
| 5 | 0.5500 | 0.7246 | 0.1746 | 1.1354 |
| 6 | 0.7246 | 0.9348 | 0.2102 | 1.2038 |
| 7 | 0.9348 | 1.1609 | 0.2261 | 1.0757 |
| 8 | 1.1609 | 1.3980 | 0.2371 | 1.0487 |
| 9 | 1.3980 | 1.6437 | 0.2457 | 1.0360 |
| 10 | 1.6437 | 1.8964 | 0.2527 | 1.0286 |
| 11 | 1.8964 | 2.1536 | 0.2573 | 1.0182 |
| 12 | 2.1536 | 2.4063 | 0.2527 | 1.0182 |
| 13 | 2.4063 | 2.6520 | 0.2457 | 1.0286 |
| 14 | 2.6520 | 2.8891 | 0.2371 | 1.0360 |
| 15 | 2.8891 | 3.1152 | 0.2261 | 1.0487 |
| 16 | 3.1152 | 3.3254 | 0.2102 | 1.0757 |
| 17 | 3.3254 | 3.5000 | 0.1746 | 1.2038 |
| 18 | 3.5000 | 3.6761 | 0.1761 | 1.0084 |
| 19 | 3.6761 | 3.8402 | 0.1641 | 1.0731 |
| 20 | 3.8402 | 4.0000 | 0.1598 | 1.0265 |

**b) variant 2**

**Fig. 4. Screen output in a tabular format of the grid aspect ratio for the two variants of the x-grid.**

Looking at the screen output we can notice that exactly at the boundaries between the regions (cell index *5* and *14* in Fig. 4a) there are unacceptably high grid aspect ratios. Therefore a second variant of the grid should be created which avoids the sudden jumps in the *x*-distance between neighbour control volumes (see the same figure). This is achieved on the expense of increasing the number of control volumes from *16* to *20* and restricting (at the same time) the power-law coefficient to be closer to unity (compare Figs 3a and 3b). The resulting grid aspect ratio for this second variant is shown in Fig. 4b – all aspect ratios are now kept below or equal to the desired ratio of *1.2*.

Fig. 5 shows the consequent menus/submenus used for the gridding process in the ESP-r. The so far created *x*-grid is then presented in Fig. 6.

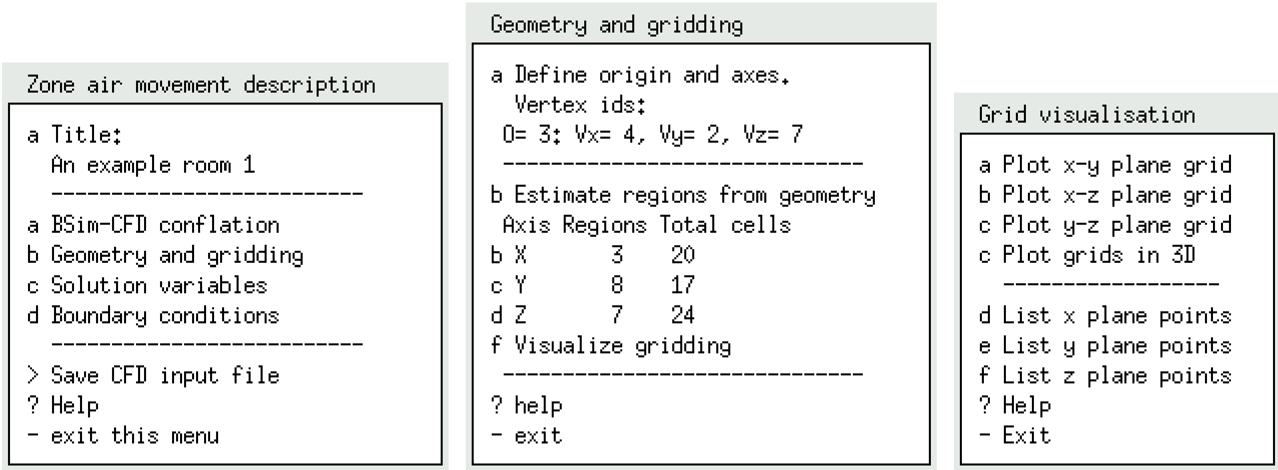


**Fig. 5. Three consequent menus governing the gridding process. Fig 3. is a result of option "b" in the middle menu while Fig. 4 is produced by option "d" in the last menu**

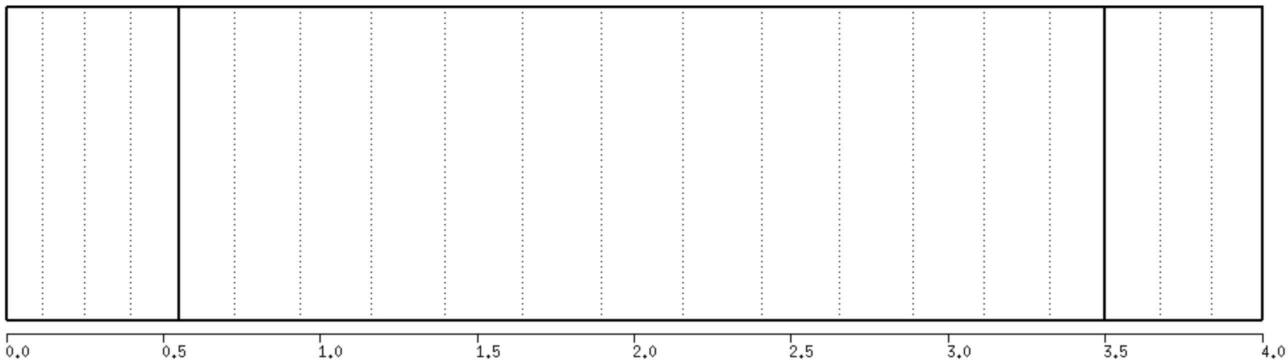


**Fig. 6. The constructed grid for *x*-direction (variant 2) containing 20 control volumes. All vertical lines (called "grid lines") show the boundaries of the control volumes. The thick lines present the boundaries of the regions used to prepare the grid. In the final grid there is no difference between the thick and the thin lines**

For gridding the two other directions (*y* and *z*) it is good first to start with a vertical view of the room and to have a look at the elements which define the boundaries of the regions, Fig. 7. What concerns the gridding process, there is no principal difference between the *y* and *z* directions; therefore we will continue the explanations with the vertical (i.e. *z*) direction.

Description of the regions and their *z*-dimensions are presented in Table 1. Looking at the table one could notice that two of the regions (*5* and *7* – the shadowed ones) are quite small. Such small-size regions could affect the whole numerical grid (trough the first rule) making it unnecessarily fine. However, very often such regions could be omitted by just moving their boundaries until they coincide with the boundaries of neighbour regions. Of course, the whole process is allowed only if the effect on the expected solution is be "small" or "acceptable". In our case one should keep on any price the dimensions and position of the inflow opening. Therefore we choose to move the upper part of the window *5 [cm]* up (see the arrow in Fig. 7) in order to align it with the lower part of the inflow opening (alternatively one could decide to move the whole window *5 [cm*] up). By this movement region *5* is included (lumped) into region *4*. Similarly, we could move also the upper part of the door by *10 [cm]* up to coincide with the upper part of the inflow opening; however, this movement seems not necessary as the upper part of the door is “within” the inflow opening which should anyway be covered by a very fine grid. As a final result the number of regions in *z*-direction is now 7 (see Fig. 8b).

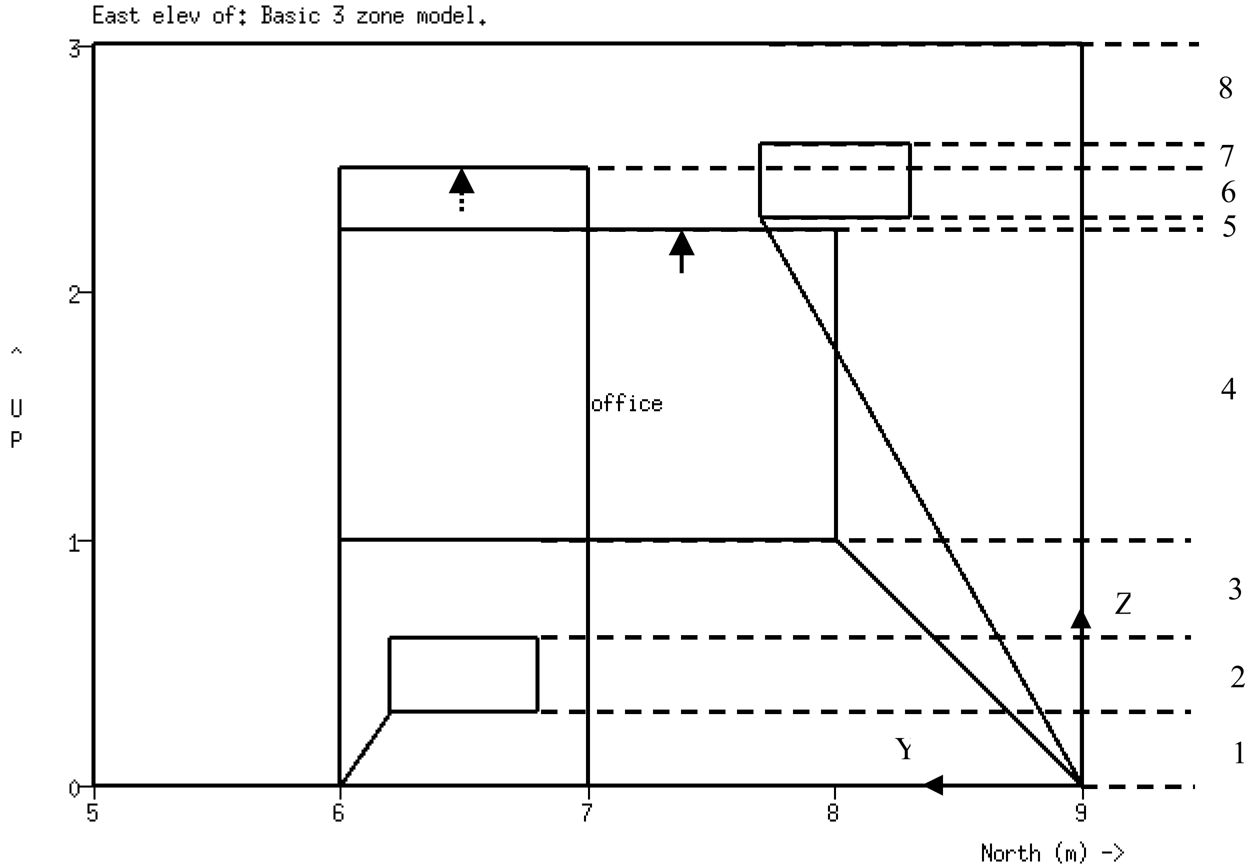


**Fig.7. Vertical view trough the room - dashed lines denote boundaries of regions for gridding (see also Table 1)**

**Note: The *y*- and *z*-coordinates shown here in *[m]* correspond to the whole-building model in the ESP-r. *Z*-coordinates are the same for both the CFD-model and the whole-building model while y-coordinates differ: for the CFD model the *y*-direction goes from right to the left while for the whole-building model the *y*-direction is exactly the opposite**

The regions and cell distances for the *y*-direction are presented in Fig. 8a. The *yz*-projection of the completed grid is given in Fig. 9.

*Comment:* The process of combining regions together is especially recommendable in rooms with many pieces of furniture and/or technical equipment. In such rooms quite a lot of regions exist; at the same time moving pieces of furniture about *5-10 [cm]* away from their original position usually has almost no influence on the accuracy of the final results.

Up to now the gridding process regarding the first rule is completed (for all axes). Lets check in the following the grid for consistency with the other rules. In order to do this we collect the necessary information about grid cell distances from figures 4b and 8a,b and summarize it in Table 2. The third column of the table shows that the ratio is within the bounds, prescribed by the second rule.

**Table 1. Information required to create the grid in the vertical (*z*) direction**

| Region No | "Begin ÷ end" boundaries of the region | Vertices [m] | z coordinates [m] | length of region [m] |
|---|---|---|---|---|
| 1 | floor ÷ outflow_1 | 2 ÷17 | 0÷0.3 | 0.3 |
| 2 | outflow_1 ÷ outflow_2 | 17÷19 | 0.3÷0.6 | 0.3 |
| 3 | outflow_2 ÷ window_1 | 19÷14 | 0.6÷1.0 | 0.4 |
| 4 | window_1 ÷ window_2 | 14÷15 | 1.00÷2.25 | 1.25 |
| 5 | window_2 ÷ inflow_1 | 15÷21 | 2.25÷2.30 | 0.05 |
| 6 | inflow_1 ÷ door_2 | 21÷12 | 2.30÷2.50 | 0.2 |
| 7 | door_2 ÷ inflow_2 | 12÷24 | 2.50÷2.60 | 0.1 |
| 8 | inflow_2 ÷ ceiling | 24÷6 | 2.60÷3.00 | 0.4 |

**Note: Outflow_1 means the lower end of the outflow opening and outflow_2 – its higher end an so on. The numbers of regions correspond to the numbers on the right hand side of figure 7**

```
y-axis gridding

 Number of regions:    8
 Total domain length:        4.000
 Current defined length:     4.000
 ------------------------------------
   Region  Cells  Length  P-law coeff  sym
a      1      3    0.700    1.100      no
b      2      1    0.300    1.000      no
c      3      1    0.300    1.000      no
d      4      3    0.700    0.900      no
e      5      1    0.200    1.000      no
f      6      3    0.600    1.000      no
g      7      1    0.200    1.000      no
h      8      4    1.000    1.100      no
 ------------------------------------
+ add/delete region
? help
- exit
```

```
Grid in the Y axis...
Cell index, Y start & end,   distance, aspect ratio
      1    0.0000    0.2091    0.2091    1.0000
      2    0.2091    0.4481    0.2391    1.1435
      3    0.4481    0.7000    0.2519    1.0536
      4    0.7000    1.0000    0.3000    1.1911
      5    1.0000    1.3000    0.3000    1.0000
      6    1.3000    1.5604    0.2604    1.1519
      7    1.5604    1.7860    0.2255    1.1546
      8    1.7860    2.0000    0.2140    1.0539
      9    2.0000    2.2000    0.2000    1.0701
     10    2.2000    2.4000    0.2000    1.0000
     11    2.4000    2.6000    0.2000    1.0000
     12    2.6000    2.8000    0.2000    1.0000
     13    2.8000    3.0000    0.2000    1.0000
     14    3.0000    3.2176    0.2176    1.0882
     15    3.2176    3.4665    0.2489    1.1435
     16    3.4665    3.7287    0.2622    1.0536
     17    3.7287    4.0000    0.2713    1.0345
```

**a) results for *y*-gridding**

```
z-axis gridding

 Number of regions:    7
 Total domain length:        3.000
 Current defined length:     3.000
 ------------------------------------
   Region  Cells  Length  P-law coeff  sym
a      1      2    0.300    1.140      no
b      2      2    0.300    1.000      no
c      3      3    0.400    1.000      no
d      4     10    1.300    0.950      no
e      5      2    0.200    0.900      no
f      6      1    0.100    1.000      no
g      7      4    0.400    1.000      no
 ------------------------------------
+ add/delete region
? help
- exit
```

```
Grid in the Z axis...
Cell index, Z start & end,   distance, aspect ratio
      1    0.0000    0.1361    0.1361    1.0000
      2    0.1361    0.3000    0.1639    1.2038
      3    0.3000    0.4500    0.1500    1.0925
      4    0.4500    0.6000    0.1500    1.0000
      5    0.6000    0.7333    0.1333    1.1250
      6    0.7333    0.8667    0.1333    1.0000
      7    0.8667    1.0000    0.1333    1.0000
      8    1.0000    1.1459    0.1459    1.0940
      9    1.1459    1.2818    0.1359    1.0731
     10    1.2818    1.4142    0.1324    1.0265
     11    1.4142    1.5444    0.1302    1.0171
     12    1.5444    1.6729    0.1285    1.0127
     13    1.6729    1.8002    0.1273    1.0101
     14    1.8002    1.9264    0.1262    1.0084
     15    1.9264    2.0517    0.1253    1.0072
     16    2.0517    2.1762    0.1245    1.0063
     17    2.1762    2.3000    0.1238    1.0056
     18    2.3000    2.4072    0.1072    1.1553
     19    2.4072    2.5000    0.0928    1.1546
     20    2.5000    2.6000    0.1000    1.0773
     21    2.6000    2.7000    0.1000    1.0000
     22    2.7000    2.8000    0.1000    1.0000
     23    2.8000    2.9000    0.1000    1.0000
     24    2.9000    3.0000    0.1000    1.0000
```

**b) results for *z*-gridding**

**Fig. 8. The final grid in *y* and *z* directions**

**Table 2. Grid data for checking the second and the third rules (CV=Control Volume)**

| direction | maximum distance [m] | CV No | minimum distance [m] | CV No | ratio: max / min [ - ] |
|---|---|---|---|---|---|
| x | 0.2573 | 11 | 0.1132 | 1 | 2.27 |
| y | 0.30 | 4 and 5 | 0.20 | 9-13 | 1.50 |
| z | 0.1639 | 2 | 0.0928 | 19 | 1.76 |

The third rule requires that we check the ratio for each combination between the minimum and maximum distances from Table 2. Obviously, the worst case is presented by the largest number (*0.30*=$\Delta y_{max}$) and the smallest number (*0.0928*=$\Delta z_{min}$) in the table which gives a ratio of *3.23*. Therefore the third rule is also satisfied. This is usually the case when the room dimensions are close to each other.

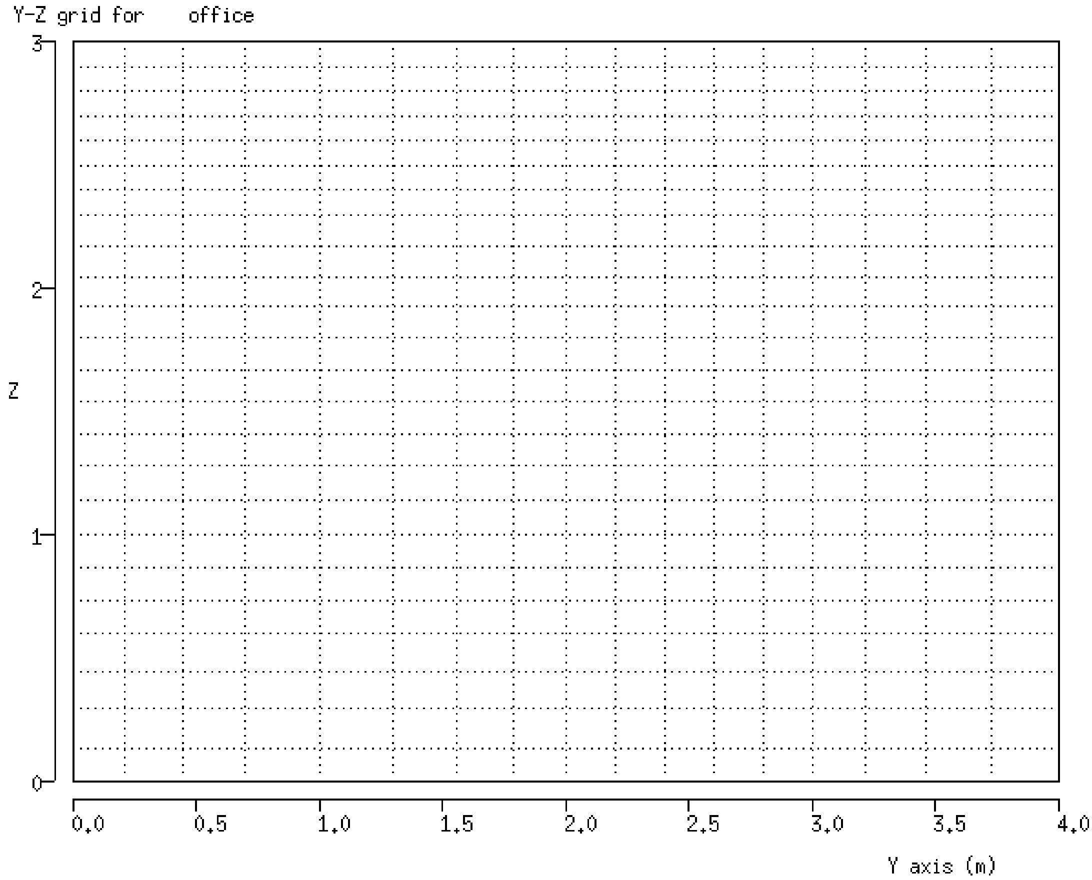


**Fig. 9. The graphical presentation of the grid lines in the final *yz*-grid**
**Note: This view is a mirror image of the view from Fig. 7**

As it concerns the fourth rule – our grid has now only *20 x 17 x 24 = 8 160* control volumes, i.e. we could expect quick solutions within minutes on a PC. This makes the grid suitable for testing of all other modelling features. Numerical grids with such small numbers of control volumes do not deliver grid-independent solutions. Therefore a considerably finer grid should be created for the final computations. This issue is addressed in the next section.

## 4. Final grid refinement

When planning the appropriate grid size, one should keep in mind that the CPU-time is proportional to $N^2$, where $N$ is the number of control volumes. Assume that in our example with *8 160* control volumes the computation need on a PC *10* minutes to converge. If one tries to increase the grid resolution twice in each spatial directions, the number of control volumes will then increase *8* times (the control volumes will then be *40 x 34 x 48 = 65 280*). The expected CPU-time for the refined grid now will be $8^2=64$ times longer, i.e *640* minutes, or more than *10* hours ! reasonable CPU-times on today's PCs are achieved usually with *100 000* to *800 000* control volumes.

Very often problems in ventilated rooms are stationary. When planning CPU-time for such simulations one should know that problems with buoyancy usually require at least *2-3* times longer iterations to achieve convergence.

With much finer grids the first control volume near the wall becomes smaller and smaller in size. However, as mentioned in the introduction, turbulence models with wall functions put an upper limit to the size of the control volume near the wall. This rises the following question - how fine is allowed to be the grid near the wall in case of turbulence models with wall-functions ?

For models with wall functions (as e.g. the most widely used standard $k$-$\varepsilon$ model) it is assumed that the viscous sub-layer of the boundary layer ends at a dimensionless distance $y^+ < 11.62$ (see Denev 1995); the first grid point should be outside this region. Assuming further that in ventilated rooms the velocities parallel to walls are usually in the interval $c_{par}=0.1\text{-}0.2\ [m/s]$ , we may calculate the minimum distance to the wall from the following equations:

$$(1) \qquad y^+ = \frac{y.u_\tau}{\nu}$$

$$(2) \qquad \frac{c_{par}}{u_\tau} = \frac{1}{\kappa} ln(y^+ \cdot E)$$

With constants $\kappa=0.41$, $E=9.0$ (for smooth walls), $y^+ = 11.62$ and the viscosity of air equal to $\nu=1.485.10^{-5}\ [m^2/s]$, equation (2) could be solved for the friction velocity $u_\tau$ . Then the dimensional wall distance $y$ can be obtained from equation (1). As a result we obtain that the viscous sub-layer is in the region $y = 0.0098 \div 0.01956\ [m]$. This is the distance between the first grid point (which is positioned in the middle of the first control volume) and the wall. Therefore we obtain finally that the first control volume should be larger than approx. *2 - 4 [cm]* in the direction normal to the wall.

## Conclusions

Both modern commercial software or freeware packages like the ESP-r (see URL://www.esru.strath.ac.uk) contribute to popularising the Computational Fluid Dynamics as a design tool for engineers. However, CFD remains a complex field which still requires a lot of specialized knowledge. This complexity usually leads to errors by novices in the field. Therefore the present paper has been devoted to give some popular understanding of the important gridding process. To support this target four simple rules has been formulated. The rules allow the design engineer to create numerical grids of good quality without going deeply in the theory of CFD. The application of the rules has been demonstrated on a step-by-step basis using an example of a ventilated room.

## References

**Denev, J. A. 1995.** Numerical Modeling of Tree-Dimensional Turbulent Flows in Ventilated Rooms. Ph.D. thesis. Faculty of Power and Mechanical Engineering. Technical University of Sofia, 168p.

**Denev, J. A. and P. Stankov 2000** `Non-Orthogonal Grid Generation Techniques Applied to Room Airflow Modeling' , *Proceedings of the XXVI Summer School 'Applications of Mathematics in Engineering and Economics', June 11-17, Sozopol, Bulgaria*, B.I. Cheshankov and M.D. Todorov, eds., Heron Press, Sofia, pp. 236-240

**FLOVENT on-line documentation** for software version 3.2, **2001**. (URL://www.flovent.com)

**Peric, M. 1985.** A finite volume method for the prediction of three-dimensional fluid flow in complex ducts. PhD Thesis, University of London, 299p.