FLUENT TUTORIAL 03
SIMULATION OF WIND AROUND BUILDINGS

FLUENT TUTORIAL 03: SIMULATION OF WIND AROUND BUILDINGS

Purpose

Simulation of wind conditions around buildings is important due to the need to analyze pedestrian wind comfort and wind resources around buildings. For high rise buildings many urban authorities require wind comfort studies (by wind tunnel or CFD) before a building permit is granted. The purpose of this tutorial is to provide guidelines and recommendations for setting up and solving a case of air flow around buildings by means of Fluent. User Defined Function (UDF) is used to specify a parabolic wind profile at the inlet. Grid adaption is used to generate mesh of different densities in order to investigate grid independence of the solution.

Prerequisites

This tutorial assumes that you are familiar with the FLUENT interface and that you have good understanding of basic setup and solution procedures. The mesh generated in Gambit tutorial 05 –Wind around buildings will be used as input. It is strongly recommended that you go through Gambit tutorials 05 before this tutorial.

In this tutorial, you will use UDF function and a turbulence model, so you should have some experience with them. This tutorial does not cover the mechanics of these models, but focuses on the applications of this model to solve air flow around buildings.

Problem description

The problem considered is wind conditions around the Canal buildings in Gladsaxe, Denmark (Fig. 1). The apartment buildings are located in the municipality of Gladsaxe, which is a part of the suburbs of Copenhagen. As shown in Fig. 1, there are 22 buildings with a building height of max. 50 m. For the sake of simplicity, small buildings, trees and the bushes are not considered in the model. The 12 buildings to be included in the model are given in Fig. 2.

The investigation for the structural purposes of the wind flow is to see whether there will be an increase of velocities from one building to another, due to channeling effects or other effects the buildings might create. The investigation for the comfort purposes is to investigate the spots with higher velocities, which will not give any effect to the structural part, but will give disturbance due to noise or vibrations.

Creation of geometry and mesh of the model have been described in Gambit Tutorial 05-Wind around buildings. Readers who have not finished the Gambit tutorial are advised to do so before starting this tutorial.

Fig. 1 Canal Buildings in Gladsaxe.

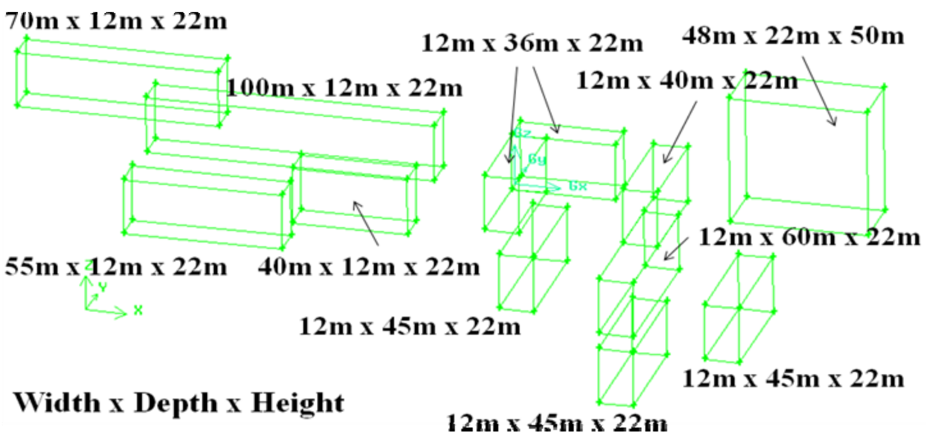


Fig. 2 Buildings considered

Setup and Solution

1 Step 1: Grid

1.1 Read the mesh file, ?????.msh

File → Read → Case

?????.msh is the mesh file exported in the Gambit Tutorial 05-Wind around buildings.

1.2 Check the grid

Grid → Check

The grid has to be checked first by the menu Grid/Check. An error message will be shown in the Text User Interface (TUI) if there is anything wrong with the grid. Otherwise the grid is ok if there is no error message. It is useful to always check the domain extents to avoid any mistakes in model scale. The size of the computational domain is shown in the TUI in meters.

1.3 Display the grid (see Fig. 3)

Display → Grid

(a) Under **Edge Type**, select **Feature**.
(b) Click Display to view the grid.
Feature *means that only the feature lines will be shown.* ***All*** *means that the grid lines and the features line will be shown. Notice that it is computer demanding and slow to show grid lines, therefore only enable All if necessary.*

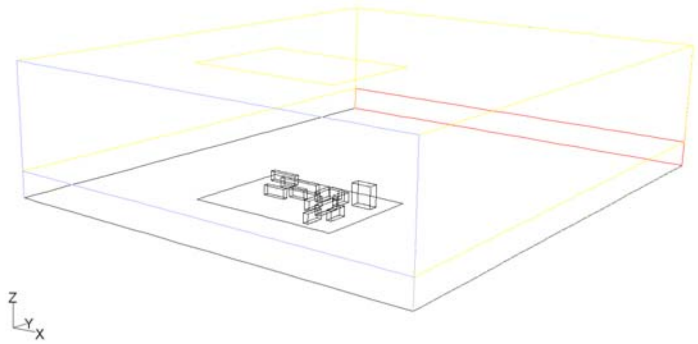


Fig. 3 Graphic display of the grid (Features)

2 Step 2: Models

2.1 Define the solver settings.

Define → Models → Solver

(a) Under **Gradient Option**, select **Node-based**.
This provides improved resolution of gradients, particularly with coarse or skewed cells.
2.2 Enable k-epsilon turbulence model.

Define → Models → Viscou

The flow conditions with regard to laminar or turbulent flow has to be determined or estimated by Reynolds number. The air flow around buildings is estimated to be turbulent, therefore an appropriate turbulence model is required. In this investigation, the k-epsilon turbulence model is chosen.
(a) Under Model, enable k-epsilon (2 eqn).
(b) Under k-epsilon model, enable RNG.

3 Step 3: Materials and operating conditions

Define → Materials

In this step you will specify fluid properties for air. Since the interest of the investigation is air flow around buildings, the solid properties of the ground and the buildings are not considered. You don't need to create new materials in the setup. As default, air is used as the fluid.

Define → Operating conditions

As there is no large pressure different across the domain, air flow around buildings can be considered as incompressible, therefore the default operating pressure is correct. As the air flow can be considered isothermal, there is no need to specify the gravitational force.

4 Step 4: Boundary conditions

4.1 Wind profile, turbulent kinetic energy and turbulence dissipation rate at the inlet.

As shown in Fig. 4, wind velocity at the inlet is not uniform. The wind is lower close to the ground and higher away from the ground. The wind profile can be represented by:

$$U(y) = \frac{U_f}{\kappa} \ln\left(\frac{y + y_0}{y_0}\right)$$

where U(y) is wind speed at the height y, m/s.

κ is the Von Karman constant (= 0.42).

y_0 is the aerodynamic roughness length of the ground, m. Values for different types of landscapes can be read from the table in the lecture slides.

U_f is the friction "velocity", m/s, which can be calculated by the following equation.

$$U_f = \frac{\kappa U_{10}}{\ln\left(\frac{10 + y_0}{y_0}\right)}$$

where U_{10} is wind speed at 10 m height, m/s. y_0 is the aerodynamic roughness length, m.

For example, $U_{10} = 10$ m/s, $y_0 = 0.5$ m → $U_f = 1.38$ m/s.

Please note that the wind speed in the DRY (Danish Reference Year) is the meteorological wind speed measured at 10 m height in an open area.

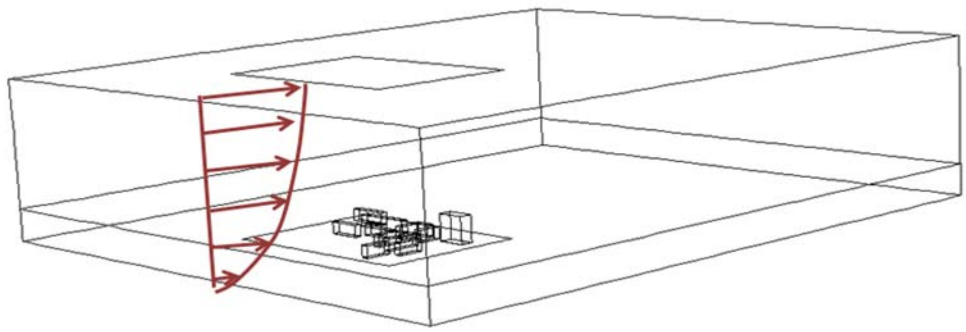


Fig. 4 Non-uniform wind velocity at the inlet.

The turbulent kinetic energy and the turbulence dissipation rate at the inlet can be estimated by:

$$k = 3.33 U_f^{\,2} \quad \text{and} \quad \varepsilon = \frac{U_f^{\,3}}{\kappa(y + y_0)}$$

The turbulent kinetic energy and the turbulence dissipation rate are shown in the Fig. 5. The red curve shows the turbulent kinetic energy *k*. It can be seen that it changes with the height. The turbulent kinetic energy is simplified with a constant value $3.33U_f^2$.

In order to have the right wind velocity, turbulent kinetic energy and turbulence dissipation rate at the inlet, UDF functions are needed. The UDF code (in C program language) can be programmed in any text editor. The UDF code is given in the appendix. You can copy the code to any text editor and save it as inletprof.c . Alternatively the file can be downloaded from the CampusNet. The path of the folder where the file is saved shall not be too deep. It is recommended to use D:\works\. A deep path may cause problems when interpreting the UDF code.

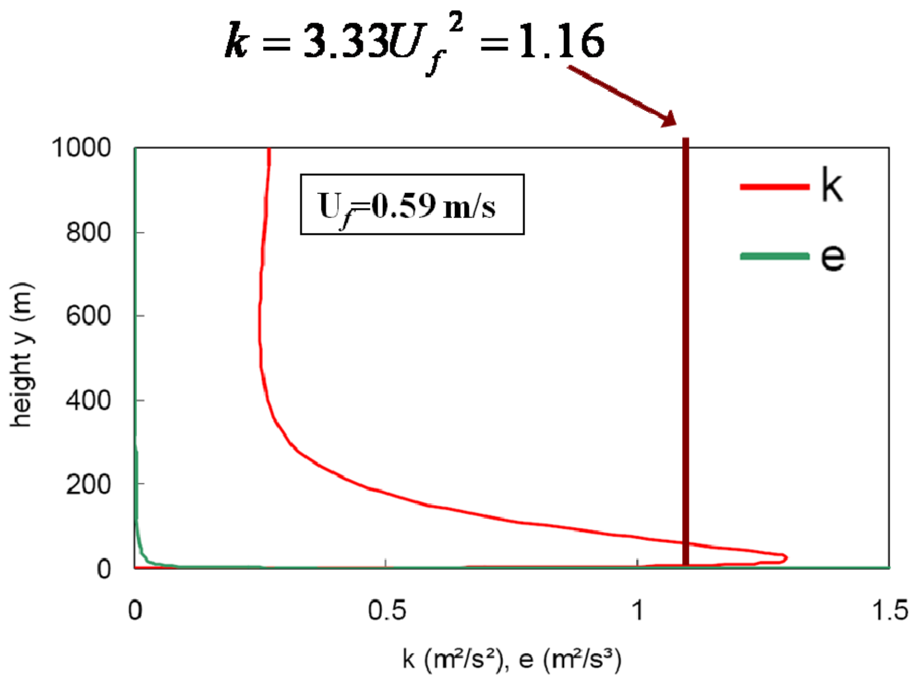


Fig. 5 Turbulent kinetic energy k and turbulence dissipation rate ε.

1. Interpret the UDF code.

User defined → Functions → Interpreted

(a) Under **Source file name**, use Browse to find the source file (inletprof.c).

(b) Interpret and close.

Check the TUI (Text User Interface) for error message. An error message will be shown in TUI if there is anything wrong with the interpretation. A file udfconfig.h will be created in the same folder as the case file.

2. Define boundary conditions at the inlet.

Define → Boundary conditions

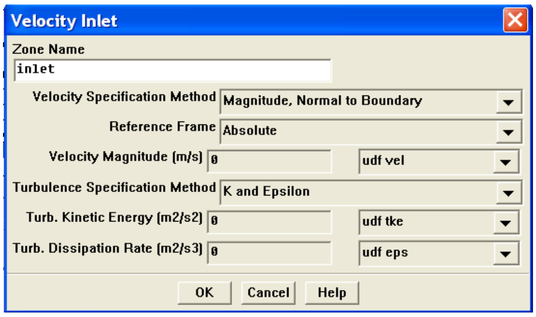
(a) Under Zone, select 'inlet'; Check boundary type is 'velocity-inlet'

(b) Click 'set'

(c) Select 'udf vel' for velocity magnitude instead of 'constant'; Select 'udf ke' for turbulent kinetic energy and 'udf eps' for turbulence dissipation rate. Click ok. (see Fig. 6)

(d) On the Boundary conditions panel, click 'copy'. Select the zone 'inlet' under **From Zone**. (see Fig.7) The other zones with the same type as 'inlet' will be shown under **To Zones**. Select 'inlet:025' and click **copy** and close.

In steps (a) and (d), please note that the velocity inlet boundary condition shall be applied to all the surfaces where the wind comes into the domain. The serial number in the name of the surfaces 'inlet:025' might be different from case to case.



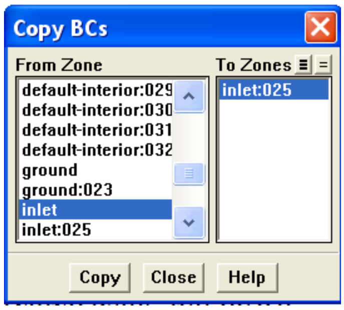


Fig. 6 Specification of boundary conditions for inlet.

Fig. 7 Copy of boundary conditions

4.2 Boundary conditions for the ground surfaces.

In Fluent, the roughness of a wall is implemented for sand-roughened surfaces with a corresponding roughness height k_s. The relationship between roughness height k_s (m), roughness length y_0 (m) and the roughness constant C_s (-) is:

$$k_s = \frac{9.793 y_0}{C_s}$$

The roughness length y_0 for different types of landscapes can be found in the lecture slides.

Note that Fluent 6.2 does not allow roughness height k_s to be larger than the distance between the centre point of the wall adjacent cell and the ground, y_p. Any k_s with a value larger than y_p will be automatically set equal to y_p without notice. In that case, k_s is taken equal to y_p and C_s is chose to satisfy the above equation. However C_s is limited in the interval [0,1].

For this case, the distance between the centre point of the wall adjacent cell and the ground is 0.4 m. The area around the buildings can be considered as open area with a roughness length of 0.03 m. C_s is calculated to be 0.73.

It shall be mentioned that special attention shall be paid to the depth of the first row when creating mesh and assigning boundary layer mesh to the ground surface. The depth of the first row shall be twice the roughness height of the ground.

Please note that if the first grid distance y_p is too small and if the roughness length is too large, it is impossible to find a value Cs in the range of [0,1]. Therefore the roughness effect will be underestimated. Calculation of a highly rough terrain may lead to unintended acceleration of the wind.

1. Specify boundary conditions for the ground surfaces.

Define → Boundary conditions

(a) Under Zone, select 'ground'; Check boundary type is 'wall'

(b) Click 'set'

(c) Under **Momentum**, key in a roughness height of 0.4 and roughness constant of 0.7 (see Fig. 8).

(d) On the Boundary conditions panel, click 'copy'. Select the zone 'ground' under **From Zone**. The other zones with the same type as 'ground' will be shown under **To Zones**. Select 'ground:023' and click **copy** and close (see Fig.9).

In steps (a) and (d), please note that the wall boundary condition shall be applied to all the ground surfaces. The serial number in the name of the surfaces 'ground:023' might be different from case to case.

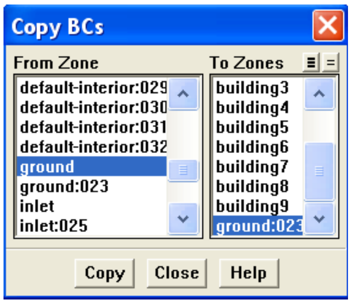


Fig. 8 Wall boundary conditions for the ground surface. Fig. 9 Copy of boundary conditions

4.3 Boundary conditions for outlet, top and lateral surfaces of the domain.

Define → Boundary conditions

Check the boundary type 'pressure outlet' for zone 'outlet' and 'outlet:024'.

Check the boundary type 'symmetry' for zone 'symmetry', 'symmetry:001' and 'symmetry:021'.

You can keep the default settings of the boundaries.

Note that the boundary conditions are predefined in Gambit and imported in Fluent. You can change the boundary type in Fluent if it is not correct.

The serial number in the zone names might be different from case to case.

5 Step 5: Solution

5.1Define the solution control parameters.

Solve → Controls → Solution

(a) Specify the Under-Relaxation Factors for Pressure, Momentum to 0.3 and 0.7 respectively.

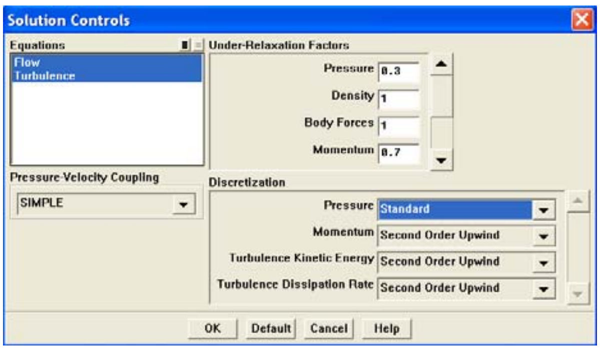
The default relaxation factors can be used as a start. This will provide optimum convergence rates for most cases. If there is a problem of convergence, the relaxation factors can be decreased, for instance to 0.2 and 0.4 for Pressure and Momentum respectively.

5.2 Reset limits.

Solve → Controls → Limits

(a) Key in a value of 1E08 for the Maximum Turbulent Viscosity Ratio.

A warning will be given in the TUI window if these limits are exceeded. This will lift the threshold for warning of high turbulent viscosity ratio.



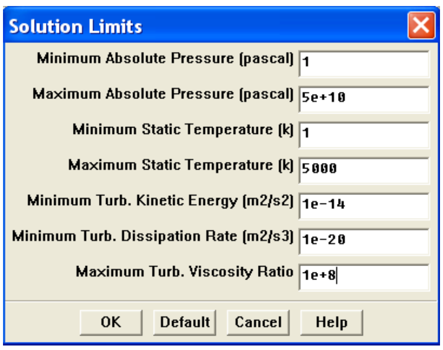


Fig. 10 Under-relaxation factors. Fig. 11 Solution limits

5.3 Initialize the solution.

Solve → Initialize → Initialize…

(a) Keep the default values for all parameters.

(b) Click Init to initialize the solution.

Initialization may take a while.

5.4 Create an Iso-surface by Sweep Surface.

Display → Sweep surface

We want to create an iso surface at the vertical middle plane of the domain. (see Fig. 13)The iso surface can be used to plot the results. It can be seen that the isosurface is vertical to axis X. The iso surface can be created in different ways, for example by sweeping surfaces. In Fig. 12, the value under Sweep Axis X is 1, which means the surface to be created is perpendicular to X axis.

Note: Fig. 13 can be shown by /Display/grid.

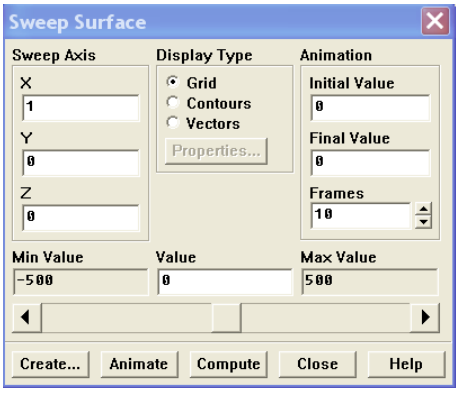
(a) Check that the value under Sweep Axis X is 1.

(b) Click Compute to calculate the range of X coordinate. *(The minimum and the maximum values of X coordinate will be shown under Min Value and Max Value respectively.)*

(c) Click Create and type in Name of the surface as 'Middleplane'.

Display → Grid

(d) Under Edge type, select Feature and under Faces select the faces to display (including the newly created isosurface 'Middleplane').



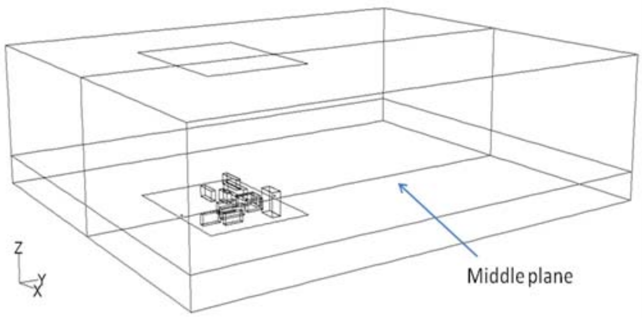


Fig. 12 Creation of the isosurface. Fig. 13 Position of the isosurface 'Middle plane'.

5.5 Define residual and surface monitors.

Solve → Monitors → Residual

(a) Check the box beside **Plot** and click ok.

(b) Change Convergence criteria for continuity, x, y, z velocities k and epsilon to 1e-06. (see Fig. 14)

The convergence criteria are different for different cases. A decrease of the criterion will prolong the iteration.

Solve → Monitors → Surface

One criterion to judge convergence is that the solution does not change with more iteration. It is therefore important to plot the solution of every iteration. The monitored solution shall be essential and representative of the solution. Here, area weighted average velocity of the surface ‘middleplane’ is monitored.

(a) Increase **Surface Monitor** to 1.

(b) Enable **Plot**, **Print** and **Write options**.

(c) Click Define …

(d) In the Define Surface Monitor panel, set the following parameters as shown in Fig. 15.

5.6 Start the iteration.

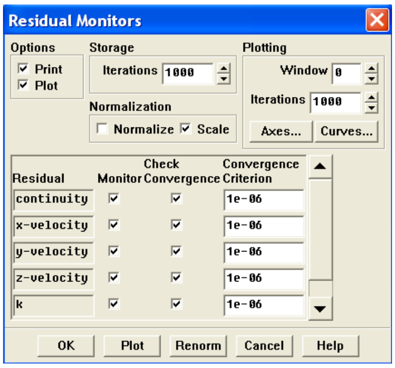
Solve → Iterate

(a) Beside the Number of iterations, key in a number of 3000.

A number of 3000 does not necessarily mean that the simulation has to run 3000 iterations. The iteration will be stopped if the convergence criteria are fulfilled.

(b) Click Iterate

There will be three windows respectively for Residual and Surfaces monitor, see Fig. 16.



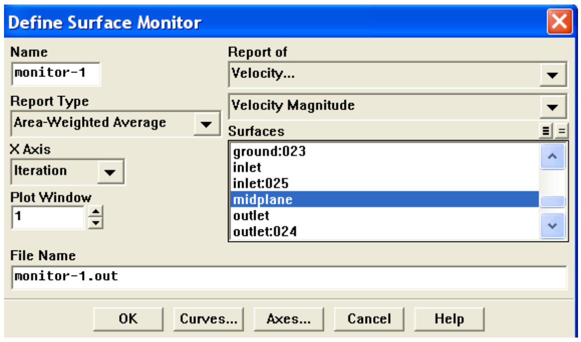


Fig. 14 Modification of convergence criteria.   Fig. 15 Definition of surface monitor

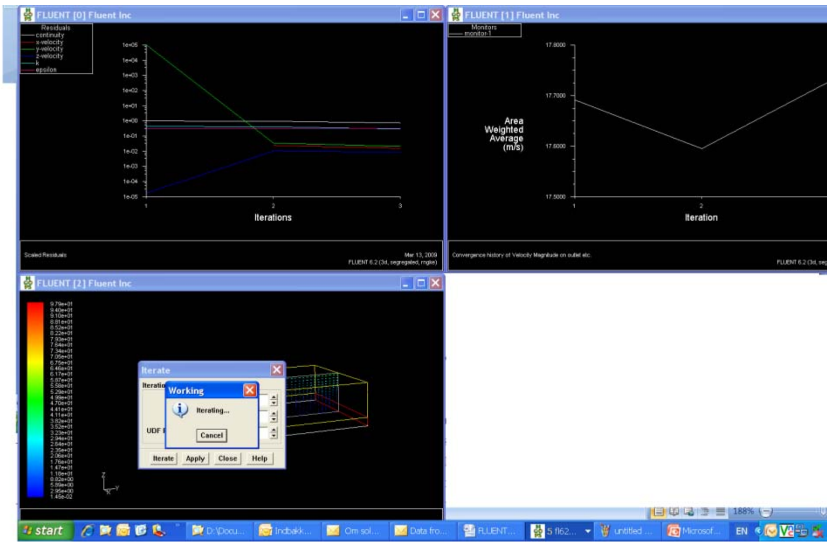


Fig. 16 Monitoring windows during iterations.

5.7 Convergence.

The result is useful only if it is convergent. To be convergent, the solutions shall get stable and the residuals shall be decreased to a certain level, therefore monitoring of residuals and the solutions are of great importance. The iteration will stop if the convergence criteria are realized. In Fig. 17, it can be seen that the residuals fall gradually with iterations. The residual of continuity drops by 5.E-06 while the residuals of momentum and turbulence equations drop by 1E-08. This means that the solution tends to be convergent after 1700 iterations. The average air velocity of the middle plane is shown in Fig. 18. It can be seen that the average air velocity gets stabilized after 800 iterations. It is concluded that the solution is convergent.

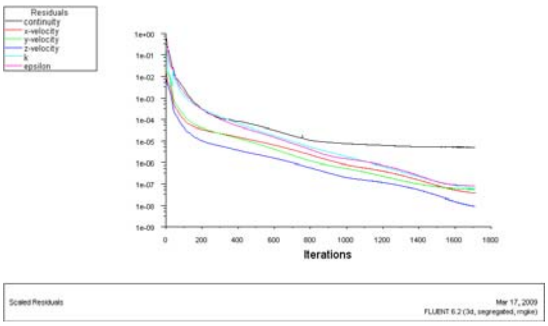


Fig. 17 Residuals.

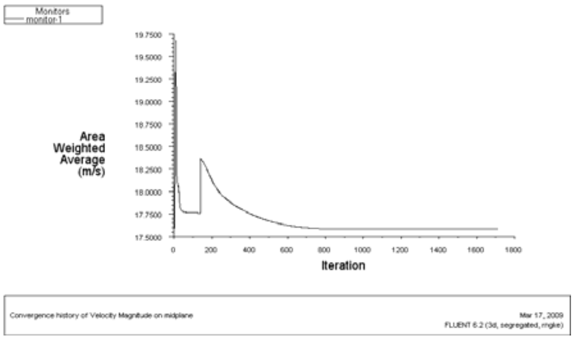


Fig. 18 Monitoring of average air velocity of the middle plane.

6. Step 6: Post-processing

6.1 Create iso-surfaces at the pedestrian level (1.5 m from the ground).

Surface → Iso-surface

(a) On the **Iso-surface** panel, select Grid in the drop-down list under **Surface of Constant**; Select Z coordinate.
(b) Click **Compute**.
The extent of the Z coordinate will then be shown.
(c) Key in -9.5 under **Iso-Values** and type in name 'pedestrian' under **New Surface Name** (see Fig. 19).
Since the ground is at Z=-11, the isosurface 1.5 m from the ground is at Z=-9.5. Note that the Z coordinate could be different depending on position of the ground surfaces created in Gambit.
(d) Click **Create**.
(e) On the **Iso-Surface** panel, select Z coordinate; Key in 39 under **Iso-Values**; Type in name 'level50m' under **New Surface Name**; Click **Create**.
This will create a surface at the top of the tallest building, 50 m from the ground.

6.2 Display contours of velocity magnitude in the pedestrian level.

Display → Contours

(a) On the **Contours** panel, enable **Filled** and **Draw Grid** under **Options**.
This will fill the contours with colour.
(b) Select **Velocity** and **Velocity magnitude** under **Contours of**; Select 'midplane' under **Surfaces** (see Fig. 20).
(c) Click **Display**.

(d) On the display window, right mouse button click the window bar and click **Page Setup**.
(e) On the Page Setup panel, Select Color and click OK.
Grey Scale or Monochrome can be selected if a grey scale or black and white images are preferred.
(f) On the display window, right mouse button click the window bar and click **Copy To Clipboard**.
Now the contours is copied to the clipboard which can be pasted to other document editing programs (see Fig. 21). Note that for a large model it might take a minute to copy the image to the clipboard.

6.3 Display Vectors in the middle plane.

Display → Vectors

(a) On the **Vectors** panel, enable **Draw Grid** under **Options**; Key in 4 for **Skip** (see Fig. 22).
For image with dense vectors, the skip function can be used. For example, a value of 4 means that every one in four vectors will be shown.
(b) Select 'midplane' under **Surfaces** (see Fig. 22).
(c) Click **Display**.
(d) On the display window, right mouse button click the window bar and click **Copy To Clipboard**.
Now the vectors is copied to the clipboard which can be pasted to other programs (see Fig. 23).

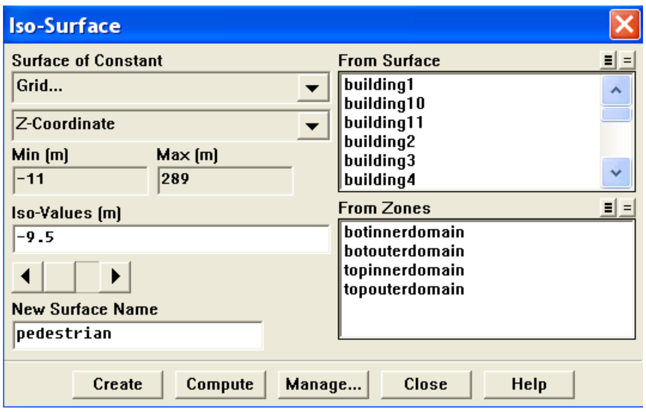


Fig. 19 Creation of Iso-Surface at 1.5 m above the ground.

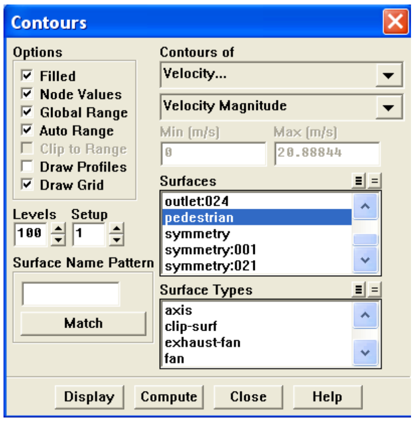


Fig.20 Contours of velocity magnitude.

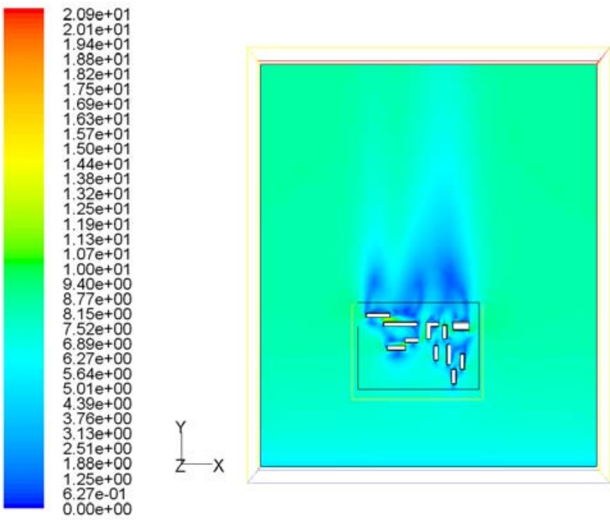


Fig. 21 Temperature contours at the middle plane.

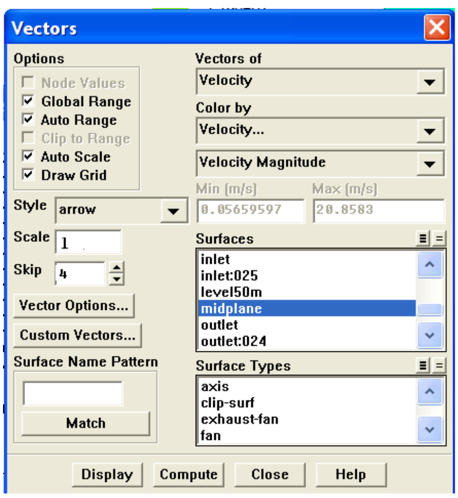


Fig.22 Contours of static temperature.

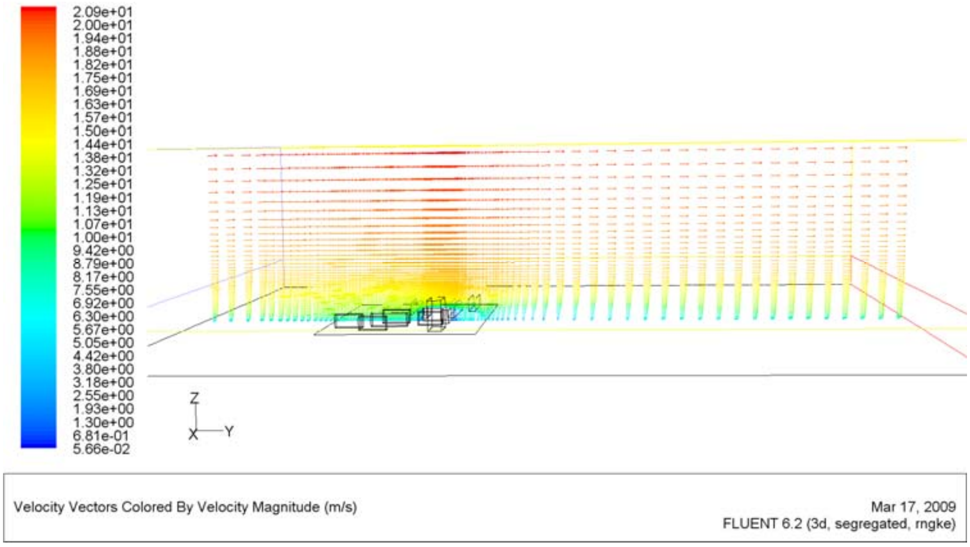


Fig. 23 Vectors at the middle plane.

6.4 Display Vectors and Contours of Static Temperature on the other iso-surfaces.

Repeat the steps in Chapter 6.2 and Chapter 6.3 and plot velocity magnitude contours and vectors at the iso-surfaces 'level50m' and 'pedestrian'. Copy the images to Paint and save the image files in bmp format.

Create a folder under your group in the CampusNet and upload the images.

6.5 Create a line in the domain and show Y velocity component.

Surface → Iso-surface

(a) On the **Iso-surface** panel, select the face 'midplane' under **From Surfaces**; select Grid in the drop-down list under **Surface of Constant**; Select Y coordinate.

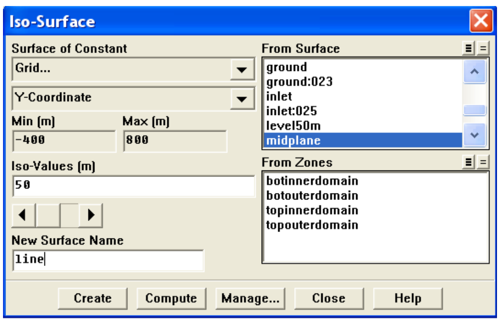
(b) Click **Compute**.

The extent of the Y coordinate will then be shown.

(c) Key in 50 under **Iso-Values** and type in name 'line' under **New Surface Name** (see Fig. 24).

(d) Click **Create**.

This will create a line at Y=50 at the middle plane as shown in Fig. 25.



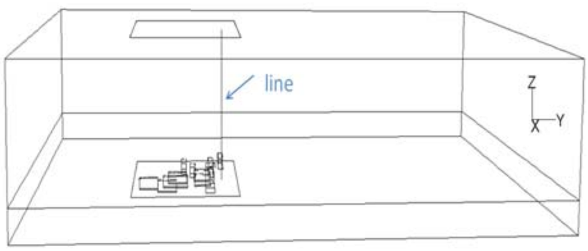


Fig. 24 Creation of a line by iso-surface.

Fig. 25 The iso-surface 'line'.

Plot → XY plot

(a) On the **Solution XY plot** panel, key in 1 for Z under **Plot Direction**; select Velocity and Y velocity under **Y axis Function** (see Fig. 26).

A value of 1 for Z means that the plot is along the Z direction.

(b) Select 'line' under **Surfaces**.

(c) Click Plot.

The Y velocity vs. Z coordinate is shown in Fig. 27.

(d) On the **Solution XY plot** panel, enable **Write to File**; Click **Write**; Type in a name 'original.txt' and write.

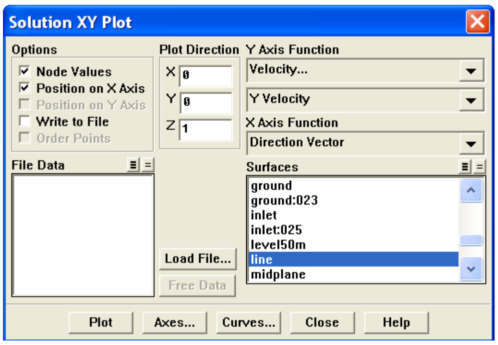


Fig. 26 Plot of Y velocity at the ***surface*** 'line'.

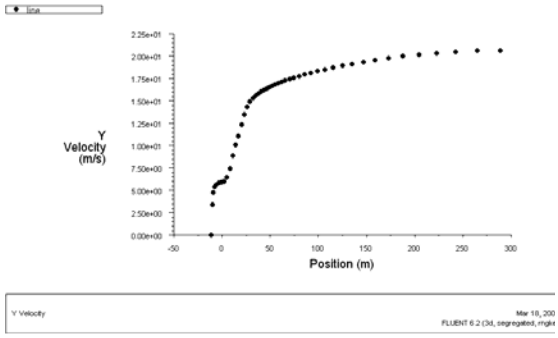


Fig. 27 Y velocity vs. Z coordinate.

7. Step 7: Grid adaption

7.1 Adapt the grid using pressure gradient.

Adapt → Gradient

(a) On the **Grid adaption** panel, select **Gradient** under **Method**; select **Pressure** and **Static Pressure** under **Gradients of**; Disable **Coarsen** under **Options** (See Fig. 28).

As we don't want to coarsen the grid, the Coarsen option is disabled.

(b) Click **Compute**.

The minimum and the maximum gradients of static pressure are under 'min' and 'max' respectively. The threshold for refinement is set to 3 which means that any place with a gradient higher than 3 will be refined with a denser mesh.

(c) Click **Mark** in the **Gradient Adaption** panel.

The cells are marked for mesh refinement. In the TUI window, the number of cells marked for refinement will be shown.

(d) Click **Manage** in the **Gradient Adaption** panel.

This allows us to have a look at the cells that are marked for refinement.

(e) On the **Manage adaption registers** panel (Fig. 29), select 'gradient-r0' under **Registers**; Click **Options** on the right bottom corner and Enable **Draw Grid**; Click Display.

*Every time you click **Mark**, the adaption is registered as gradient-r **x (x is serial number)**. If there are more than one registered adaptions, you shall keep the correct one and delete the rest.*

The marked cells is shown in Fig. 30.

(f) Click **Adapt**; Click Ok to confirm the adaption.

In the TUI window, a summary of the Grid size of the original and the adapted mesh will be shown.

As a rule of thumb, you shall try to adapt the grid so that the total number of cells increases by 20-30%.

The number of cells depends on RAM member of the computer.

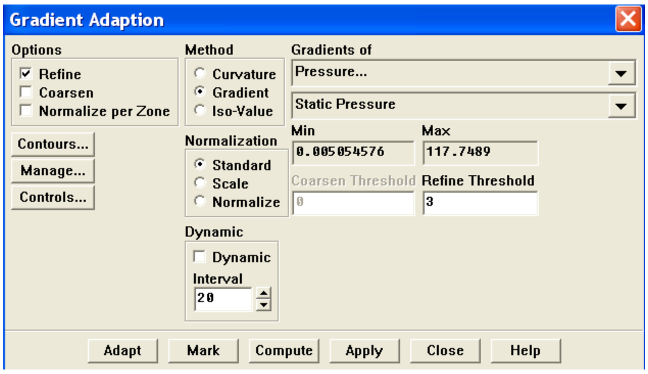


Fig. 28 Grid adaption using pressure gradient.

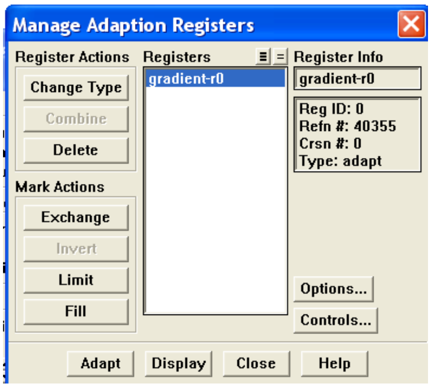


Fig. 29 Display the marked cells.

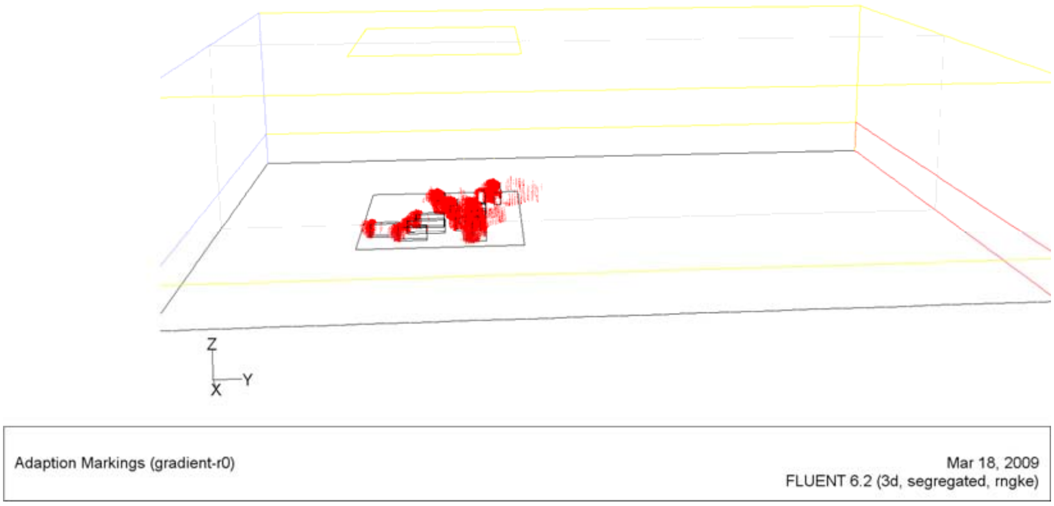


Fig. 30 The cells to be refined marked in red.

7.2 Start iteration.

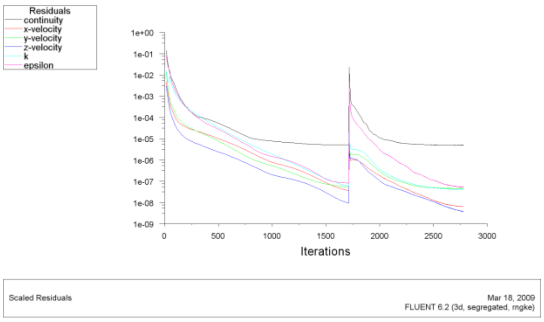
Solve → Iterate

(a) Beside the Number of iterations, key in a number of 2000.

The iteration will be stopped if the convergence criteria are fulfilled.

(b) Click Iterate

The solution has to be recalculated after the grid adaption. It is useful to start the iteration based on the solution of the original grid. It means that you do not initialize the solution. The residuals of the iterations after adaption are shown in Fig. 31. It can be seen that the residuals have a sharp increase and gradually drop again. The residual shall drop to a level similar to the case with the original grid. The solution is also monitored to check convergence (Fig. 32). The average velocity at the middle plane is shown after the 1716th iteration. It is obvious that the average velocity levels off after the 2200th iteration. It can be concluded that the solution is convergent.



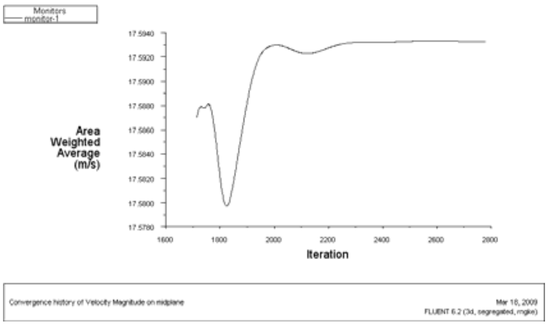


Fig. 31 Residuals of the iteration after adaption. Fig. 32 Average velocity on the middle plane.

7.3 Plot of the Y velocity at the surface ‘line’ before and after the adaption.

Plot	→ XY plot

(a) On the **Solution XY plot** panel, key in 1 for Z under **Plot Direction**; select Velocity and Y velocity under **Y axis Function** (see Fig. 26).

A value of 1 for Z means that the plot is along the Z direction.

(b) Select ‘line’ under **Surfaces**.

(c) Click **Load file** and navigate to find the file ‘original.txt’ created in Step 6.5.

(d) Click **plot**.

The Y velocity before and after the adaption will be plotted along Z axis (See Fig. 33).

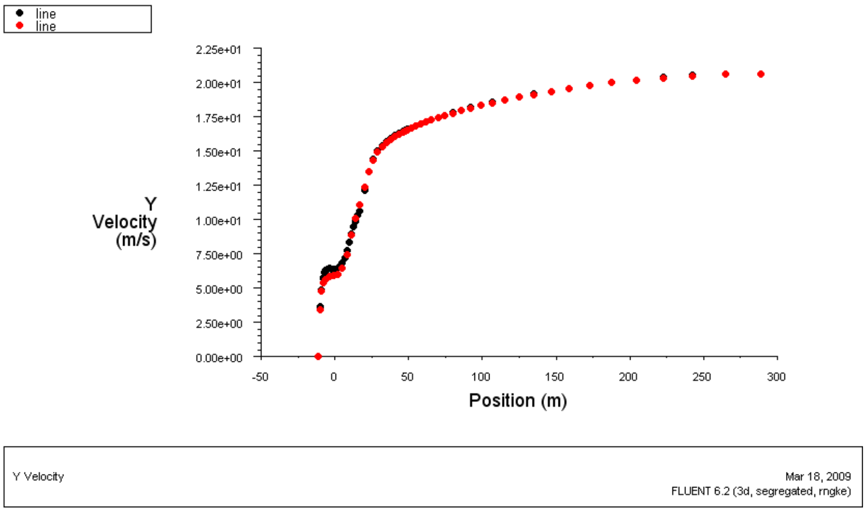


Fig. 33 Y velocity vs. Z coordinate before and after the adaption.

To finish the tutorial, please:

1, Create a folder under your group in the CampusNet.

2, Save the final case and data file and upload them, as well as the images files created in bmp format, to the CampusNet.

Appendix

```
#include "udf.h"
/* BOUNDARY PROFILES FOR WIND SPEED       Uref = 10 m/s */
/* ROUGHESS LENGTH              yo= 0.5 m */
real v = 10;
real uf = 1.38;
real yo = 0.5;
/* CALCULATION OF THE PROFILE FOR HORIZONTAL WIND SPEED */
DEFINE_PROFILE(vel, thread, nv) /* function name, thread and variable number */
{                       face_t f;
                        real x[ND_ND];
                        begin_f_loop (f,thread)
                        {
                          F_CENTROID(x,f,thread);
            F_PROFILE(f,thread,nv) = (uf/(0.42))*log((x[2]+11+yo)/yo);
                        }
                        end_f_loop(f,thread)
}
/* CALCULATION OF THE PROFILE FOR TURBULENT KINETIC ENERGY */
DEFINE_PROFILE(tke, thread, nv) /* function name, thread and variable number */
{
                        face_t f;
                        real x[ND_ND];
                        begin_f_loop (f,thread)
                        {
                          F_CENTROID(x,f,thread);
            F_PROFILE(f,thread,nv) = 3.33*(uf*uf);
                        }
                        end_f_loop(f,thread)
}
/* CALCULATION OF THE PROFILE FOR TURBULENCE DISSIPATION RATE */
DEFINE_PROFILE(eps, thread, nv) /* function name, thread and variable number */
{
                        face_t f;
                        real x[ND_ND];
                        begin_f_loop (f,thread)
                        {
                          F_CENTROID(x,f,thread);
            F_PROFILE(f,thread,nv) = (uf*uf*uf)/(0.42*(yo+x[2]+11));
                        }
                        end_f_loop(f,thread)
}
```