International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 06, June 2017
www.ijates.com
ijates
ISSN 2348 - 7550

Implementation of a deforming mesh for the flow past an oscillating square cylinder

Karthikeyan J

Mechanical Engineering Department, National Institute of Technology Kurukshetra (India)

ABSTRACT

The concept of a dynamically deforming mesh and the procedure for implementing it in ANSYS Fluent are explained highlighting the ways in which a mesh can be deformed. The user defined function necessary for the implementation was written in C language and the source code is provided. The mesh metrics required to keep track of the quality of the deforming mesh are explained.

Keywords: Ansys Fluent, Deforming Mesh, Oscillating Prism, Skewness, Smoothing Technique

I. INTRODUCTION

With the advent of high speed computers, the significance of computer simulations is fast catching up with that of the physical testing methods which are indispensable in ascertaining the quality of a product, in the development stages itself.The current article describes the implementation of a particular technique that is used in computer simuations of fluid flow problems.

II. WHAT IS A DYNAMIC MESH?

Simulations of flow past bluff bodies of various shapes generate a lot of interest as they allow us to estimate the forces acting on the bodies. But many times, it may be required to simulate the flow of a fluid past a cylinder or prism which is not stationary but is moving, the equation of the motion being known. However we are aware that a pre-processor of a CFD package wraps a mesh over the computational domain of specified dimensions. Since the cylinder lies inside the domain but its volume does not contribute to the simulations in any way, the cylinder surface is also wrapped by the mesh. The grid points contained in the mesh have unique co-ordinates which do not change with time. But in the case of flow past an oscillating cylinder, the mesh points on the cylinder also oscillate along with the cylinder. Consequently, for maintaining the topology of the grid, the nodes or grid points in the subsequent layers of elements should also move. For this to happen, the mesh must be allowed to deform with time. It is very important that the deformation of the mesh should follow the motion of the cylinder for the quality of the mesh to remain within allowable limits. As will be seen in the following paragraphs, the deformation of the mesh can be controlled in a number of ways – modelling the line elements as spring elements, solving a partial differential equation for the velocity of the nodes, defining a motion for each node individually or by specifying the geometry of the deforming zone.

International Journal of Advanced Technology in Engineering and Science
Vol. No.5, Issue No. 06, June 2017
www.ijates.com
ijates
ISSN 2348 - 7550

III. MOTION OF THE CYLINDER

For the simulation of flow past an oscillating cylinder, the motion of the cylinder must be specified. This can be done by explicitly mentioning the linear or angular velocity of the cylinder or by using a 6-DOF solver. In the first case, the material properties of the cylinder are not necessary and the solver moves the cylinder merely based on the linear or angular velocity which is usually a function of time. In ANSYS Fluent, this can be achieved using a user-defined function(UDF)[1]which uses the built-in macro *DEFINE_CG_MOTION(name, dt, vel, omega, time, dtime)*. The arguments of the macro are explained as follows:

- name: name of the function that appears in the GUI
- dt: pointer to the mesh zone to which the function is applied
- vel: an array to store the x, y, z components of the linear velocity
- omega: an array to store the x, y, z components of angular velocity
- time: a variable to get the current flow time from the solver
- dtime: a variable to get the current time step size from the solver

It is to be noted that the variable names can be any name accepted in C language but the order of the arguments must be maintained.

The second method of using the 6-DOF solver [2]allows us to specify the properties of the cylinder such as its mass, moment of inertia, etc. and the degrees of freedom allowable for the cylinder' motion. Using this and the information on fluid forces from the flow calculations, the solver estimates the cylinder motion and oscillates it. A schematic of the geometry for the simulation of flow past a square cylinder is shown in Fig 1.

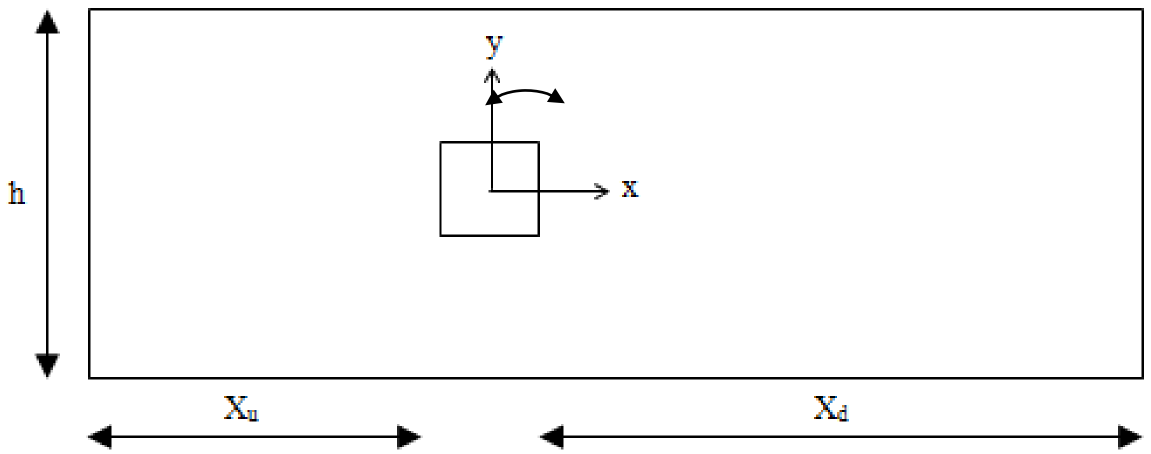

Fig. 1A schematic of the oscillating cylinder in the computational domain

IV. VARIOUS TYPES OF DEFORMING MESH AVAILABLE IN ANSYS FLUENT

The different ways of implementing a deforming mesh in ANSYS Fluent is shown in Fig 2. In Fig 2 (a) and (b), the region near the oscillating cylinder is modelled using quadrilateral cells.This is because more quadrilateral elements can be generated within a given thickness normal to the cylinder wall, as compared to the triangular elements; we require larger number of elements in the region closest to the cylinder to accurately calculate the higher gradients near the walls. However, as shown in Fig 2 (a) and (b), the region away from the walls can be modelled using either quadrilateral cells or triangular cells. In all the three images shown in Fig 2, there is a circular boundary around the vicinity of the cylinder outside of which the quadrilateral elements are body fitted, that is, the elements are aligned along the boundaries of the domain. The region within the circular boundary will be used to define the deforming zone, whereas, the small layer of quadrilateral cells near the cylinder walls along with the cylinder will be used to define the oscillating zone as will be described in the next section. In Fig 2 (c), the elements inside the circular boundary are aligned in the radial and circumferential direction and the sizes of the elements increase as they radiate out from the cylinder surface. They have been formed using a technique called O-grid to minimise the skewness of the elements. The concept of an O-grid is shown schematically in Fig 3. As shown in Fig 3, the top and the bottom edges are divided into certain number of divisions – either equally or unequally using a mathematical function for its variation. These elements are projected on to the arc formed by the interception of the line on the circle. In a similar way, the elements from the other three outer edges are projected on to the remaining three arcs. The elements on the inner edges are projected on the edges of the square cylinder.

In Fig 2(c), the elements inside the circular region oscillate as a whole and do not deform; the entire region slides on the outer stationary region and hence the mesh is called a sliding mesh. In Fig 2(b), the quadrilateral elements inside the circular region deform with time. In Fig 2(c), the triangular elements not only deform but also regenerate and collapse dynamically.

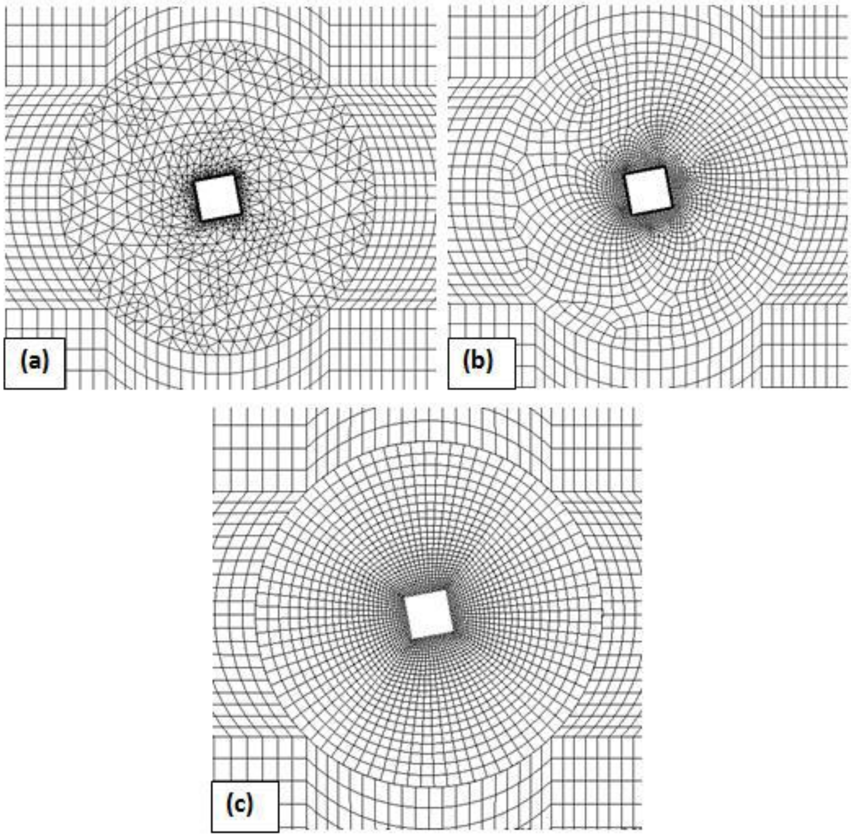

Fig. 2 Types of dynamic mesh techniques

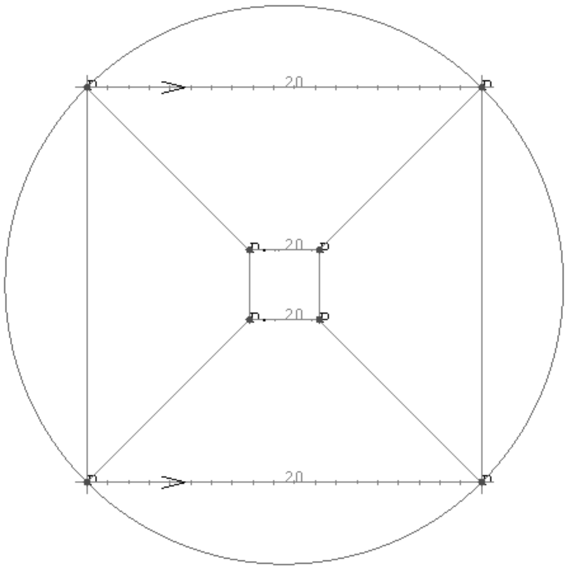

Fig. 3 The O-grid technique

V. DEFINING THE DYNAMIC MESH ZONES

The different regions of a dynamic mesh – oscillating, deforming / remeshing, stationary can be defined using the '*dynamic mesh zones'* dialog box. The *dynamic mesh zones dialog box* and its various options are shown in Fig. 4.

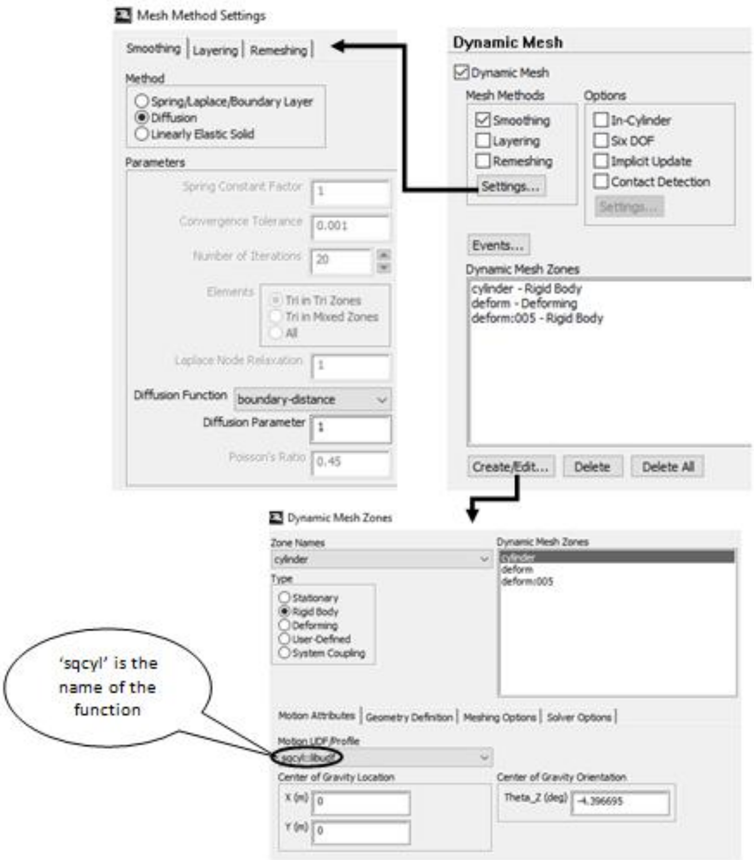

Fig. 4 Defining the dynamic mesh zones

In this case, the cylinder and the layers adjacent to it, which oscillate should be defined as *rigid body* and the oscillations of the zone can be defined using a user-defined function (UDF) written in C language*;* in Fig. 4 the name of the function is shown as 'sqcyl'.The region within the circular boundary, excepting the rigid body region, should be specified as deforming. The properties of the deforming zone can be specified from the *'mesh method settings dialog box'* shown in Fig. 4. This dialog specifies the three important ways of deforming a mesh namely *smoothing, layering* and *remeshing.*Layering technique is useful when translatory motion is involved such as the motion of a piston inside a cylinder, linear oscillation of a cylinder in a direction normal to that of the flow, etc.

In the case of a dynamic mesh with deforming quadrilateral cells, remeshing is not supported in ANSYS Fluent and hence the cells can only be deformed by smoothing technique, where in, the motion of the cylinder is transferred to the region outside the cylinder by the deformation of the elements. In smoothing techniques, the number of nodes remains the same. The smoothing of cells can be done in three ways – spring-based, diffusion-based, and linearly elastic solid-based.

In the case of spring based smoothing method, the line elements in the mesh are modelled as spring elements and their spring stiffness is specified.In the case of dynamic mesh with triangular elements as shown in Fig. 2 (a), the smoothing method should be specified as spring-based because remeshing can be used only with spring-based smoothing technique.

In diffusion-based smoothing, a partial differential equation of the velocity vector of the nodes is solved[3].The coefficient of the velocity vector, diffusion coefficient, in the differential equation should be known for solving the PDE. The diffusion coefficient is a function of diffusion parameter and the boundary distance or cell volume. The diffusion parameter is specified in the Graphic User Interface (GUI)Diffusion-based smoothing technique is more expensive than spring-based technique but is more accurate.

In the linearly elastic solid-based smoothing technique, the entire deforming mesh zone as a whole is considered as a linearly elastic solid and partial differential equations of stress tensor and the strain tensor, where in the stress and strain tensors are expressed in terms of the displacement of the mesh. The user is required to input the Poisson's ratio in the GUI.

VI. IMPORTANCE OF MESH METRICS

Once the dynamic mesh zone has been defined, it is important to preview the mesh motion and the check the quality of the mesh. In ANSYS Fluent the most important parameter that defines the quality of an element is the orthogonal quality. The orthogonal quality of an element refers to how closely the vector from the cell centroid to the centroid of a face coincides with the normal vector of the face. In other words, it refers to the cosine of the angle between the two aforementioned vectors. The steps involved in calculation of orthogonal quality are as follows[2]:

Step-1. For each face in a cell (in 2D, a quadrilateral/triangular element is called as a *cell* and each of its edges is called as a *face*), 2 normalized vector dot products are calculated as follows:

$$\frac{\vec{A_i}.\vec{f_i}}{|\vec{A_i}||\vec{f_i}|},\frac{\vec{A_i}.\vec{c_i}}{|\vec{A_i}||\vec{c_i}|}$$

where,

ijates
ISSN 2348 - 7550

$\overrightarrow{A_i}$ refers to the area vector of a face i

$\overrightarrow{f_i}$ refers to the vector from the centroid of the cell to the centroid of the face i

$\overrightarrow{c_i}$ refers to the vector from the centroid of the cell to the centroid of the neighbouring cell that shares the face i

Step-2. The minimum of the above two quantities, calculated for all the faces of a cell will be assigned as the orthogonal quality of that particular cell.

The orthogonal quality of the dynamic mesh at different time steps is shown in Fig. 5. The orthogonal quality of a cell varies from 0 to 1 where 0 represents the worst quality. In Fig. 5, it can be seen that at the flow time of 12 s, the orthogonal quality is only about 0.32. When the maximum skewness is estimated for this dynamic mesh, it comes to around 0.94 which is a very high value. The skewness of the cells can be reduced by reducing the size of the elements or by increasing the diffusion factor.

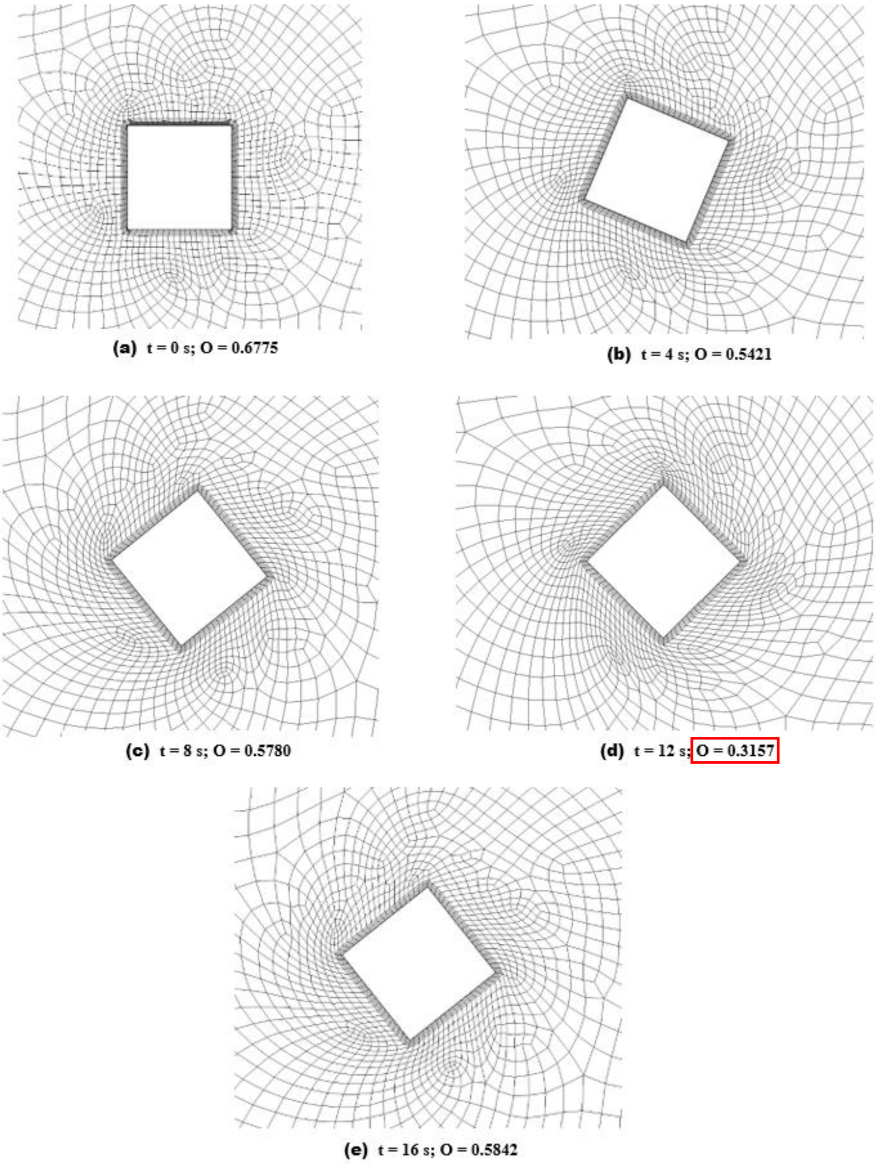

Fig. 5 The quality of the dynamic mesh zone at various instants; the simulations were done at F_R = 1.0 and θ_0 = 45°. t = flow time, O = Minimum Orthogonal quality

VII. CONCLUSION

The important problems in constructing a dynamic mesh that can be inferred from the text are the concerns regarding the quality of the mesh that need to be addressed. The different types of the dynamic mesh should also be checked for accuracy in obtaining the important integral parameters which can be compared with the existing literature on experiments and numerical simulations. It is also important to verify the grid independence and time step independence of the mesh by changing the grid size and time step size performing the simulations.

REFERENCES

[1.] UDF Manual, in ANSYS Fluent 12.0,April 2009, ANSYS Inc.

[2.] User's Guide, in ANSYS Fluent Release 15.0, November 2013, ANSYS, Inc.

[3.] Theory Guide, in ANSYS Fluent Release 15.0, November, 2013, ANSYS, Inc.