ADVANCED IMMERSED BOUNDARY CARTESIAN MESHING TECHNOLOGY IN FloEFD™

WHITEPAPER

Mentor Graphics®

MECHANICAL ANALYSIS

www.mentor.com

CONTENTS

ABSTRACT

For the numerical simulation of Navier-Stokes equations, the choice of the mesh type plays a significant role. Comparative calculations on different mesh types illustrates that the best simulation precision, characterized by minimum Local Truncation Error (LTE), is obtained on Cartesian meshes. For the boundary representation the Immersed Boundary (IB) approach, which does not require a boundary-conforming mesh, is used. Use of Cartesian meshes together with Immersed Boundary approach makes it possible to efficiently: minimize approximation errors; build operators with good spectral properties, so that robustness of method is guaranteed; speed up the process of grid generation; and make grid generation robust and flexible. Many other CFD methods require a mesh that fits the boundaries of the computational domain and often complex internal geometries. The body-fitted grid generation used is time-consuming, often requiring manual intervention to modify and cleaning-up the CAD geometry as a pre-requisite.

To implement the IB approach efficiently in FloEFD, a number of issues needed to be resolved: approximation of the governing equations in cut-cells that contain the solid-fluid interface; capture of boundary layers effects irrespective of boundary layer thickness using a Two-Scale Wall Functions (2SWF) approach (see Mentor Graphics Corp., 2011); automatic mesh generation with automatic detection of initial mesh settings (octree-based mesh structure); and Solution Adaptive Refinement (SAR).

Test cases given in this paper represent a small selection of our validation examples that illustrate the IB approach precision and flexibility of FloEFD meshing technology in the wide range of industrial examples of geometry and physical formulations.

1 INTRODUCTION

FloEFD's meshing technology has been in use for about 20 years, since the end of the 1980s. The task of developing a numerical flow simulation solution brought us to a decision point. The Navier-Stokes equations need to be discretized before they can be solved. However, before diving into the software development it was necessary to answer some important questions: Firstly, what mesh is the best for simulation precision? And secondly, what mesh is the best to use for practical applications?

When the conservation laws for fluid flow are discretized on a spatial finite-volume mesh, the *error* in approximating these governing integral equations, together with the *error* in approximating the wall boundary conditions and flow boundaries, and therefore the simulation accuracy, is defined to a great degree by the choice of mesh *geometry*.

The detailed reviews of basic types of mesh geometries are presented in several publications (see Weatherill & Hassan 1994, Filipiak 1996, and Parry & Tatchell 2008). These works show the wide use of body-fitted meshes for solving today's industrial problems. As a rule, for more complicated geometries *unstructured* meshes are generated (Fig. 1). In some cases the geometry allows the use of meshes with a regular structure. Such meshes are called *structured* meshes (Fig. 2). In addition mixed mesh topologies: structured in some sub regions of computational domain and unstructured in other ones (Fig. 3) are also used. Such meshes may be called *partially structured or partially unstructured*.

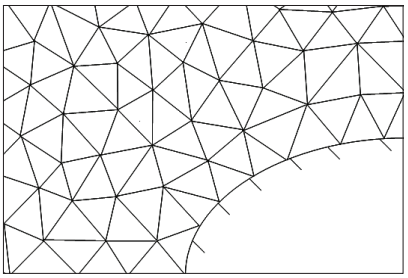


Fig. 1: Unstructured body-fitted mesh

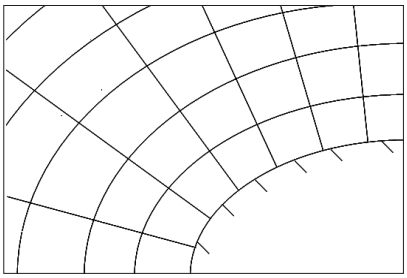


Fig. 2: Structured body-fitted rectangular mesh

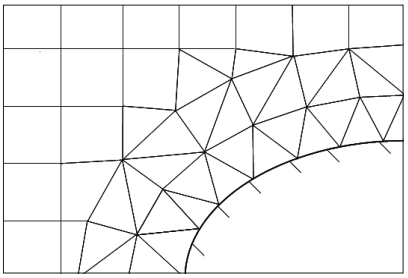


Fig. 3: Combination of structured Cartesian mesh and hybrid unstructured body-fitted mesh near the wall

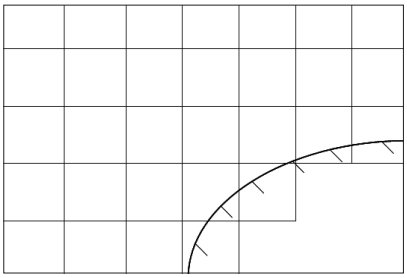


Fig. 4: Structured Cartesian immersed-body mesh

The approach that uses the *immersed-body* mesh (Fig. 4) is the alternative to the *body-fitted* approaches shown in Figure 1 to 3. Deliberately *not* conforming the mesh to the solid surface makes it possible to use a Cartesian mesh, which in general can be not *body-fitted*. Such mesh can be defined as the set of *cuboidal* rectilinear cells, which are adjacent to each other and external boundary of the computational domain and orientated along the Cartesian coordinates. Cuboidal cells intersected by the surface ("*cut-cells*") are treated according to the boundary conditions defined on the surface.

The advantages of body-fitted meshes are as follows:

- It is possible to place the mesh nodes optimally to resolve the geometrical features of the problem. This feature of body-fitted approach exhibits most clearly in the case of unstructured meshes, which are the most "flexible" in terms of arbitrary placement of nodes during mesh generation.
- It is possible to increase the mesh density towards the wall to ensure satisfactory resolution of the near-wall viscous layer.

At the same time the body-fitted meshes are characterized by serious disadvantages caused by:

- Approximation errors (Local Truncation Errors) caused by the spread of inter-nodal distances (mesh irregularity) that is greatest for the unstructured meshes (see Table 1).
- Loss of computational robustness, caused by the secondary fluxes at skewed faces, which increase the degree of schemes explicitness and, as a result, a decrease in the diagonal dominance of the difference operators.
- Serious difficulties in automating the process of mesh generation caused by the geometric complexity of the industrial problems; surface defects within the CAD-representation; and the need to control the cell size and topology in the near-wall mesh to capture boundary layer development.

The examples of two structured quadrilateral and of one unstructured triangular meshes used to simulate the inviscid, isentropic, supersonic flow of ideal gas in a plane circular channel are presented below (see Aftosmis & Berger, 2002). Fig. 5 displays the most typical variants of these meshes. Fig. 5 (a) displays the "qualitative" unstructured near equilateral triangular mesh, with angles that are not less than 29 degrees. Fig. 5 (c) displays the structured quadrilateral curvilinear mesh, which is generated by the intersection of concentric circles with radial rays. Fig. 5 (b) displays the structured mesh obtained from the curvilinear mesh Fig. 5 (c) by the division of each quadrilateral cell into two triangular ones.

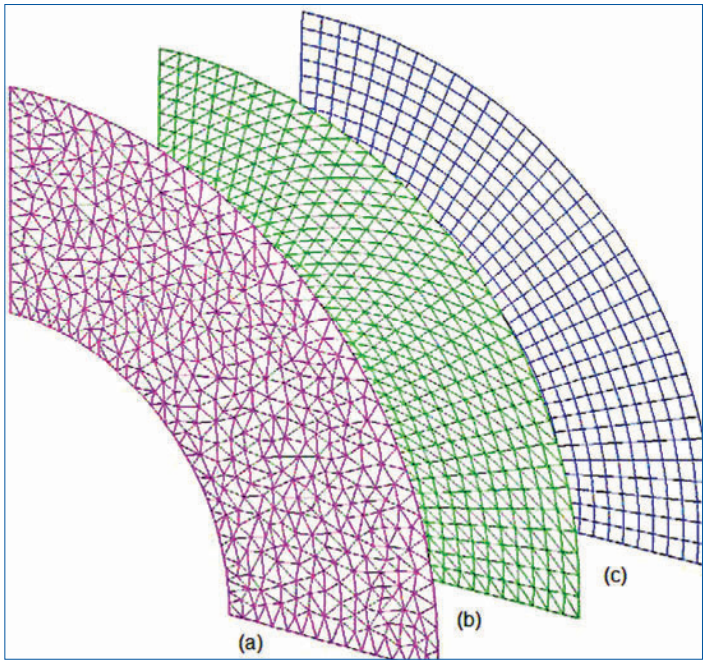
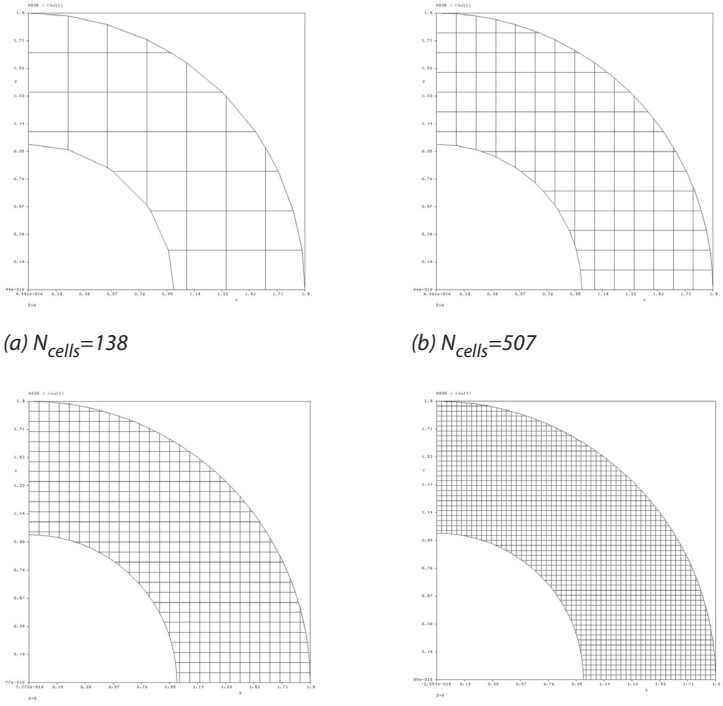


Fig. 5: Body-fitted meshes: (a) Unstructured Triangular Mesh; (b) Structured Triangular Mesh; (c) Structured Curvilinear Quadrilateral Mesh

Fig. 6 displays the sequence of successively refined IB Cartesian meshes.



(a) N_{cells}=138 *(b) N_{cells}=507*

(c) N_{cells}=1928 *(d) N_{cells}=7549*

Fig. 6 Sequence of Cartesian meshes with increased mesh density

For all mesh sequences presented above, Table 1 displays the values of Local Truncation Error (LTE) defined in the integral norm $||LTE||_{L_1}$, which are obtained by applying the finite-difference operator that approximates the mass conservation law in the supersonic flow using the scheme of the formally second order of accuracy with respect to the accurate solution (Aftosmis & Berger, 2002). Here *Cells* is the number of mesh cells; *n* is the coefficient of mesh convergence.

TABLE 1

Mesh	Unstructured triangular		Structured triangular		Structured curvilinear		Cartesian [Aftosmis]		Cartesian [FloEFD]	
	Fig. 5 (a)		Fig. 5 (b)		Fig. 5 (c)		Fig. 6		Fig. 6	
	Cells	$\|\|LTE\|\|_{L_1}$	Cells	$\|\|LTE\|\|_{L_1}$	Cells	$\|\|LTE\|\|_{L_1}$	Cells	$\|\|LTE\|\|_{L_1}$	Cells	$\|\|LTE\|\|_{L_1}$
Results	128	0.52552	144	0.37926	144	0.30998	138	0.03065	140	0.03014
	505	0.22529	525	0.07571	525	0.09223	507	0.00930	516	0.00916
	1918	0.11936	2001	0.01565	2001	0.02422	1928	0.00246	1944	0.00235
	7490	0.05940	7809	0.00347	7809	0.00629	7549	0.00059	7526	0.00058
n	1.02		2.28		1.94		2.11		2.06	

The presented results show clearly that the quality of approximation of the mass conservation law is significantly worse on non-Cartesian meshes:

- The level of $||LTE||_{L_1}$ is approximately 10 times higher on structured curvilinear meshes than on a Cartesian mesh, and higher again on structured triangular meshes. The rate at which the errors reduce with increasing mesh count, characterized by the mesh convergence coefficient, n, is of the order 2 for Cartesian and structured curvilinear and structured triangular meshes.
- Compared against the Cartesian mesh, we see the level of $||LTE||_{L_1}$ for the unstructured triangular mesh is approximately 17 times higher on a coarse mesh, rising to 100 times higher on a fine mesh. The difference in the rate at which $||LTE||_{L_1}$ reduces with reducing cell size is due to the difference in the mesh convergence coefficient, n, between structured and unstructured meshes. As n is higher for the Cartesian mesh, $||LTE||_{L_1}$ reduces faster as the Cartesian mesh is refined (also see Fig. 7).

The difference between Cartesian and unstructured meshes will be even more pronounced in 3D, where tetrahedral cells are used for the unstructured mesh. **To put this into context, based on this illustrative example, a typical industrial application in FloEFD using of the order of 1M cells, would require 100+M cells in a tool that uses an unstructured tetrahedral mesh to achieve the same level of accuracy in representing the underlying conservation equations.**

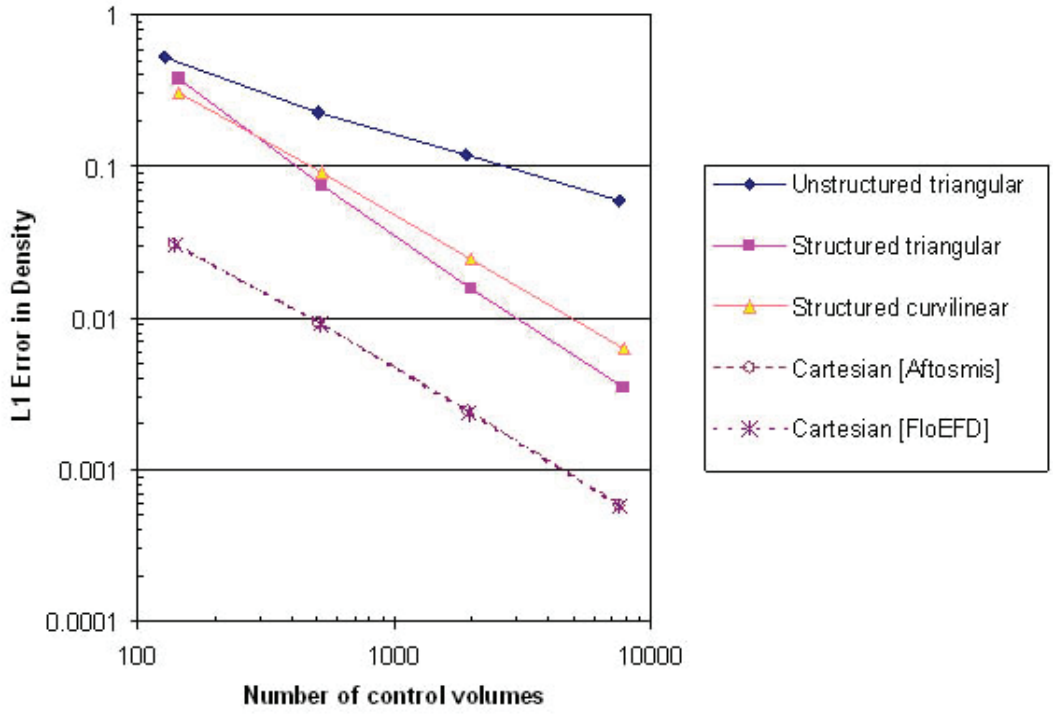


Fig. 7: L1 norm of local truncation error in density for sequence of refined meshes (see Table 1, Aftosmis & Berger, 2002)

If the mesh contains skewed cells, in which the cell face is not normal to the line between the cell centers on each side of the cell face, then $\|LTE\|_{L_1}$ is higher than for a Cartesian mesh. This includes both structured and unstructured meshes. If the mesh topology is unstructured, with an irregular distribution of mesh nodes, then the mesh convergence coefficient, *n*, is reduced from around 2 to around 1, so that the error reduces more slowly as the mesh size is increased. A Cartesian mesh is free from both of these disadvantages and can be regarded as the ideal mesh, producing the minimum $\|LTE\|_{L_1}$.

In conclusion, Cartesian meshes are characterized by their high approximation accuracy of the conservation laws. Let us examine the advantages and disadvantages of Cartesian meshes in general.

2 ADVANTAGES OF THE CARTESIAN MESH

2.1 LTE MINIMIZATION

Integral minimum of LTE follows from the fact that the structure of Cartesian mesh is maximally regular in the main area of the flow.

Irregularity in the mesh is introduced at solid-fluid interfaces, which cause the regular cells to be split into a number of control volumes. The proportion of these cells reduces as the base mesh is refined. Another source of irregularity in the mesh are cells adjacent to a change in refinement level in the mesh. However, the proportion of these cells also reduces as the base mesh is refined.

If N is the number of cells in a particular coordinate direction, then the model will typically contain of the order of N^3 cells. The number of cut cells is proportional to the surface area within the model, which is typically proportional to N^2. The proportion of cut cells is therefore proportional to $N^2/N^3 = 1/N$. A similar argument applies to the number of cells that are adjacent to a change in the mesh refinement level. Overall, irregularity is minimized the closer the mesh is to being fully Cartesian. It is then generally better from an accuracy perspective to use a finer base mesh with fewer refinement levels than vice versa, however this will lead to a finer mesh overall.

2.2 ROBUSTNESS OF DIFFERENCING SCHEME

Robustness of the differencing scheme is ensured by the absence of secondary fluxes at skewed faces, since the Cartesian based system ensures there are no skewed faces in the model. This in turn ensures maximum implicitness and diagonal dominance of difference operators in the matrices being solved.

2.3 SIMPLICITY, SPEED AND ROBUSTNESS OF MESH GENERATION ALGORITHM

Relative simplicity and a speed of automatic generation of Cartesian mesh are made possible by the application of the immersed-boundary mesh technique.

CAD systems were originally developed for design process, not for numerical simulation needs. Consequently defects in CAD geometry (so called “Dirty CAD”) are the common problem when using industrial CAD models for simulation (see Fig. 8).

Many CAD systems use Boundary Representation (BREP) geometries to define 3D objects. A BREP surface is composed of a number of patches which can be quite complex like in the case of a Non-Uniform Rational B-Spline (NURBS). This complexity is a source of non-watertight joints resulting from imperfect edge-edge contact between adjacent surface patches. This may not be a problem when producing drawings or even rapid prototypes, but having a watertight (i.e. manifold) geometry is essential when creating a mesh for a flow simulation, especially in the case of body-fitted meshes which first require a surface mesh. Due to this, special efforts are necessary to repair or clean this kind of defect (Wang & Srinivasan, 2002 and Thompson et. al, 1985) before meshing.

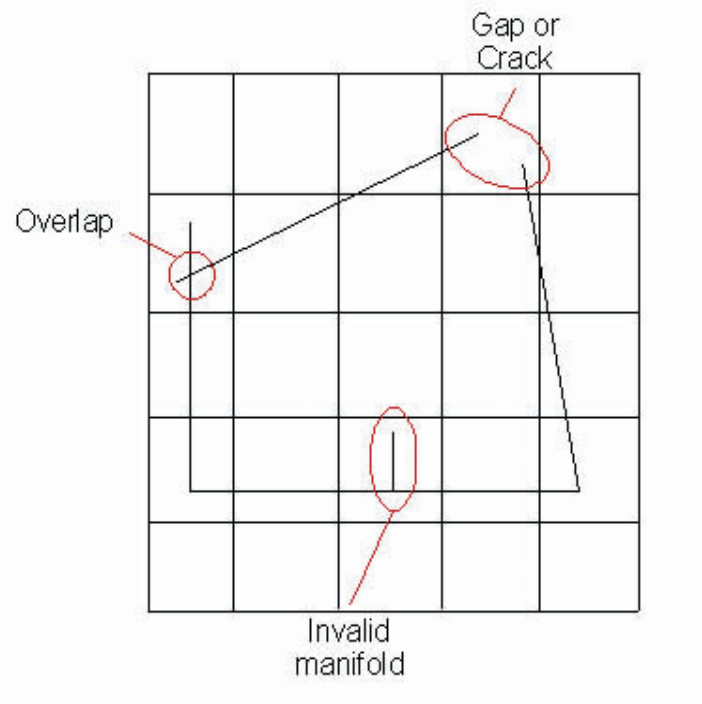


Fig. 8: Typical defects view on CAD geometry

Body-fitted meshes are characterized by their high sensitivity to the quality of CAD geometry. Commonly, defects in the surface representation require user intervention to resolve the ambiguities and heal the defects of CAD geometry. Once this is done, a surface mesh is generated, usually beginning with nodes created on the solid surface. Then the surface triangulation is generated, most often by using a Delaunay algorithm, with point insertion as required. In addition, in some situations, over-refinement of the surface can result in an excessive number of small triangles. This often happens in areas that are not important for flow simulation (for example, geometry features having a small radius, small spikes, joints and material interfaces, etc.). After that, based on the surface triangulation, the volume mesh is generated. Often it is a mesh with tetrahedral elements that meet Delaunay criterion (see Delaunay, 1934, Lawson, 1977, Watson 1981, Baker 1989 and Weatherill & Hassan 1994).

In the octree approach, when the initial mesh is generated, the geometry is always captured beginning with the large cells used for the background Cartesian mesh, and then with increasing precision as these are refined in accordance with criteria set. This approach makes it possible to stop the mesh refinement process once the mesh is locally fine enough to capture the physics of the problem.

Unresolved geometry within cut-cell can be completed by geometry interpolation inside the cell. Since the smallest cells needed to capture the physical detail present in the CAD model are relatively large compared to the size of any defects in the geometry, the sensitivity to CAD-geometry defects is much less. Cracks, overlaps, small components (for example, small spikes, etc.), and small defects in geometry definition can be ignored by using sufficiently large cells (see Fig. 9).

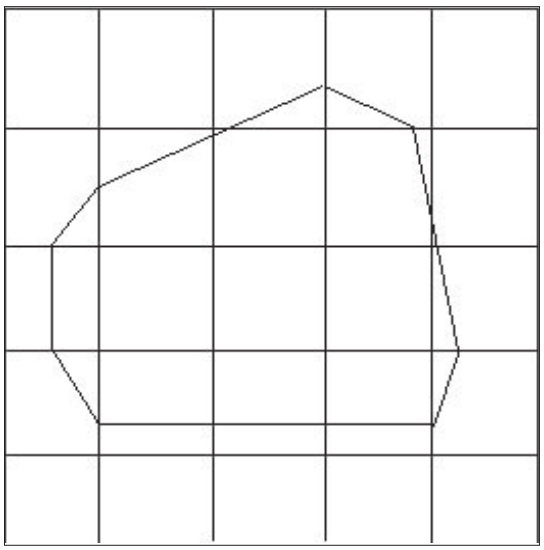


Fig. 9: Defects are 'healed' by geometry interpolation inside the cell

If there are narrow channels within the geometry that are important to the flow simulations, the required cell size can be achieved by mesh refinement up to the level corresponding to the prescribed characteristic width of the channel. Even in such cases it is typically the case that the dimensions of any CAD geometry defects are significantly less than the characteristic width of the channel.

It is necessary to point out that the immersed body mesh approach could be implemented with tetrahedral and other types of elements (see Löhner et al, 2004), but in terms of approximation accuracy and ease of implementation, use of a Cartesian mesh is much better.

2.4 SIMPLICITY OF NAVIGATION ON MESH

In discussing the advantages of a Cartesian mesh we should also point out that the structured nature of the mesh results in much faster calculation of mesh-based information required by the solver, as well as speeding up the search for data associated with neighbor cells. Examples of such computations include the construction of cell geometry from the coordinates of its vertices, determining the solid volume and surface intersection with cell elements and faces, computation of spatial steps, and the location of cell centers. Searching for neighbor cell data is facilitated by the structured nature of the base mesh and the relationship between refined cells and their parents. Together these serve to speed up the computational efficiency.

2.5 ISOTROPIC VERSUS ANISOTROPIC REFINEMENT

Mesh refinement can be implemented in a number of ways. Usually a tree structure is used, in which the larger cells are refined into smaller daughter cells. The most common approaches are binary tree refinement (Fig. 10), where the parent cell is refined to two daughter cells, or octree refinement (Fig. 11), where the parent cell is refined to eight equal daughter cells. There are other variants. For example, in FloTHERM (Parry & Tatchell, 2008) embedded localized fine grid regions are used, where each parent cell can have more than 2 children in each coordinate direction.

On the one hand the binary tree refinement is more universal and flexible, making possible to generate the meshes better adapted to the solid geometry by using fewer cells.

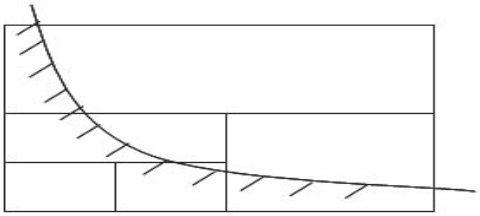


Fig. 10: Binary-tree refinement

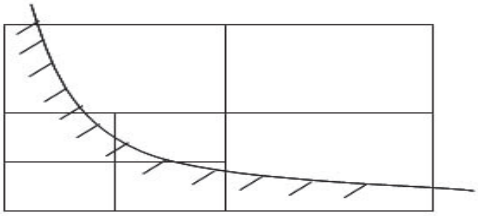


Fig. 11: Octree refinement

But practical attempts to implement such approach (see Berger & Aftosmis, 1998) show that the anisotropic refinement causes significantly irregular meshes, often with aspect ratios that decrease the solver accuracy.

For 2D problems there is a benefit in terms of cells number. But usually the typical number of cells for 2D problems is not too high; therefore, even if a 3D definition is used for solving a 2D problem, the computation time is not too high. In the case of arbitrary 3D geometry, such meshes are not sufficient to guarantee the proper resolution of the boundary layer due to the irregularity in the cell size. Additional issues with binary refinement include how to choose the refinement criterion and how to interface between regions with cells that have been refined in different orientations.

Usually, the refinement criterion is calculated as the variation of the vector normal to the surface within the cell. If the variation in a specific direction is sufficiently high, the criterion is true and refinement takes place, splitting the parent cell into two new ones orthogonal to this direction. The refinement across the normal to the surface can cause extremely high step irregularity. Due to this a more cautious refinement strategy would be needed, both initially and for Solution Adaptive Refinement.

In terms of mesh topology there are some additional problems at the boundaries of 'badly joined' refined cells. For example, if we refine two neighbor cells (see Fig. 12), in the way that one is split in the y-direction and the other in the z-direction, at the boundary we have face pairs which are not conformal.

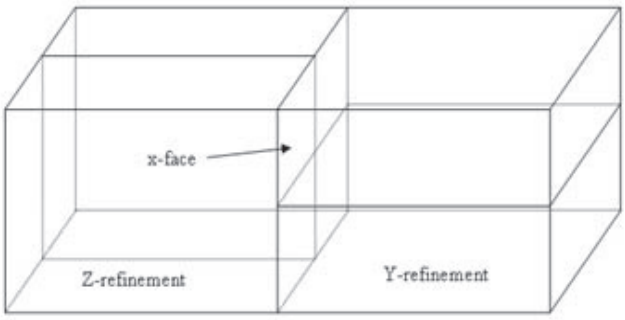


Fig. 12: Faces from the left and from the right side on x-face are different

This topological situation requires additional efforts for interfacing, either by further refinement (thereby reducing the benefit of binary refinement on cell count), or by accepting a more complicated mesh structure that requires additional software logic (increasing the computation time per cell). If additional refinement is chosen, the mesh refinement process converges sufficiently quickly, but the enforced refinement of the neighbor cells causes the cell count to increase significantly, greatly reducing the benefit of this approach.

3 CARTESIAN MESH GENERATION IN FLOEFD

The technique used for mesh generation itself is strongly dependent on the properties and possibilities of the available solver and vice versa. The key challenges that arise if the Cartesian meshes are used, are presented here along with their solution in FloEFD:

1. *Treatment of sub grid-scale viscous near-wall layers, i.e. where the thickness of the boundary layer is less than or approximately equal to the characteristic size of mesh cell normal to the wall.* This is solved through the use of a novel Two-Scale Wall Functions (2SWF) technique, where the sub-scale boundary layer is modeled using a special integral procedure (Mentor Graphics Corp., 2011). Otherwise, to solve the Navier-Stokes equations with two-equation k-ε turbulence would require a very fine computational mesh. Fig. 13 displays the flow that includes the 'thin', 'thick' and 'intermediate' layers, which are resolved universally by the 2SWF.

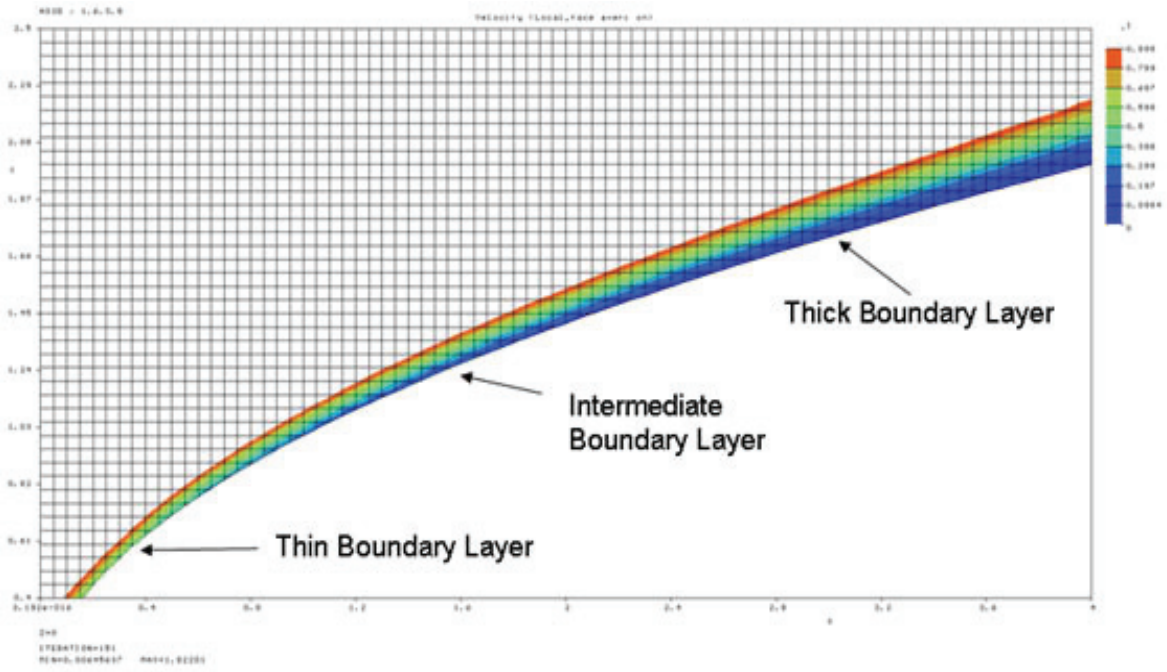


Fig. 13: Velocity flow field with "Thin", "Intermediate" and "Thick" Viscous Boundary Layer

2. *Resolution of the wall, if the wall thickness is less than the characteristic size of mesh cell (i.e. 'thin' walls)* is achieved through the implementation of an original technique, in which the mesh cell can contain any number of "fluid" or "solid" control volumes, breaking the normal paradigm of a 1:1 correspondence between cells in the mesh and solver control volumes.
3. Fig. 14 displays the velocity field around a solid, which includes segments of both "thick" and "thin" walls.

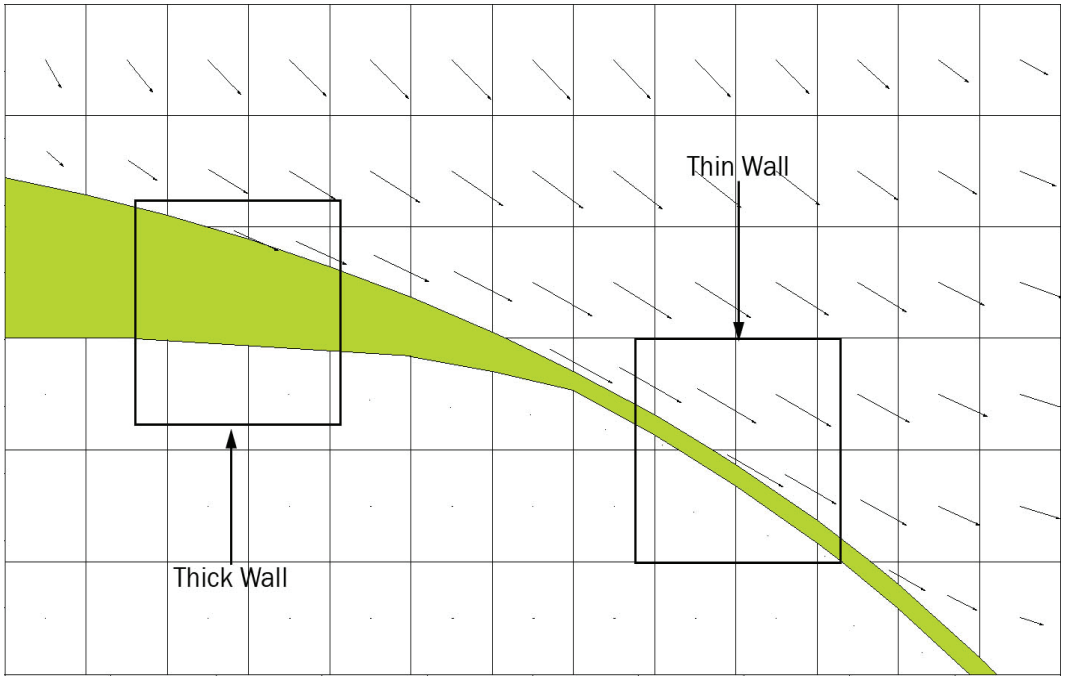


Fig. 14: Velocity fields near the solid, if there is a transition from "thick" to "thin" wall

In addition, there are several factors that result in irregular mesh step (and therefore a drop in the accuracy of the approximation):

- Limited computational resources makes it necessary to find the right balance between the total number of cells, the maximum permitted level of cell refinement (that allows to resolve the geometrical features) and the regularity of the mesh. In the case of arbitrary geometry the best choice is not evident.
- An irregular base mesh. For maximum accuracy, the cell-cell contraction or expansion of the base mesh should not be very sharp. Usually the cell-cell size ratio should not exceed 1.2:1.
- A large difference in refinement level between adjacent cells. To avoid this issue, FloEFD restricts the difference in refinement level between a cell and its neighbor to be no more than 1, forcing the refinement of neighbor cells where necessary.

There are also some specific situations that arise which will influence the process of mesh generation, for example, in the case of narrow channels, sharp edges, solid-solid interfaces, etc. We will discuss the most important aspects of this technique in the next section.

3.1 INITIAL MESH GENERATION

As it was mentioned above, the Cartesian mesh based on octree technology, is used in FloEFD.

The mesh generation process begins with the definition of the rectangular computational domain. Three sets of planes orthogonal to the Cartesian coordinate system are then defined inside it. The intersection of these planes defines the set of rectangular cells (cuboids) that form the base mesh (Fig. 15). The gaps between basic mesh planes can be varied. In addition, it is possible to specify the set of key planes, which are used to prescribe the position of some planes in the base mesh. Since it is necessary not to use a significantly irregular mesh, the restriction on the change in the spacing of the planes, h, is set to $h_i/h_{i-1}<1.15$ in FloEFD.

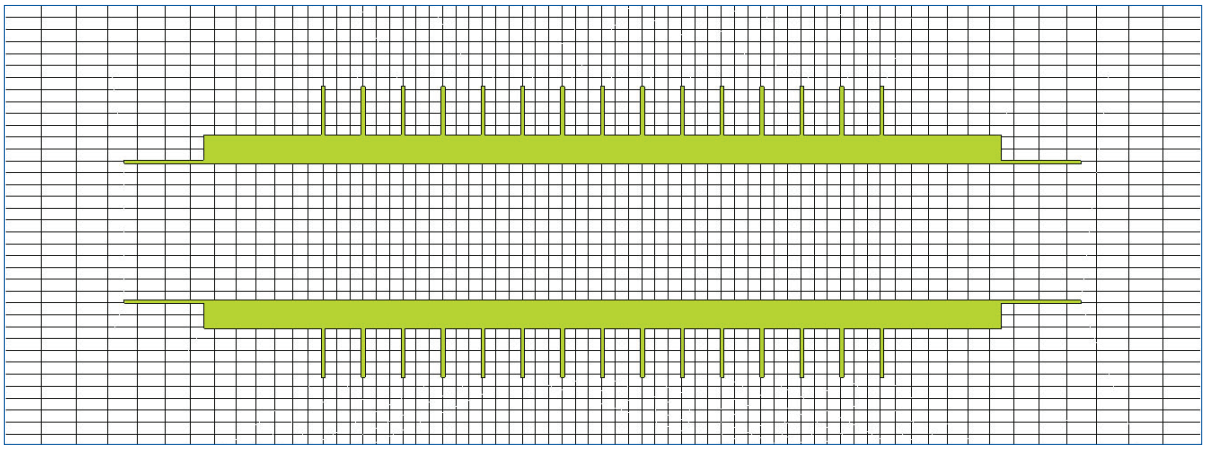


Fig. 15: Base mesh

At the next stage of the process, the geometry of the objects inside the computational domain are captured. The program traverses the mesh cells sequentially and for each cell it parses the geometrical configuration inside the cell. In addition, several refinement criteria are checked. If one of the respective criteria is true, the cell is tagged for further refinement.

The difference in refinement levels of neighbor cells should not be higher than 1 as this is necessary to minimize the approximation error. Accordingly, when the refinement criteria have been checked, additional iterations are carried out to tag cells that should be refined to ensure this condition is met.

The refinement level of the cell is the number of times it has been refined relative to the initial cell in the base mesh. In FloEFD the maximum number of refinement levels is limited to 7, so that the smallest cell size is 1/128 of the original cell in the base mesh (level 0), although most refinements will stop far short of this.

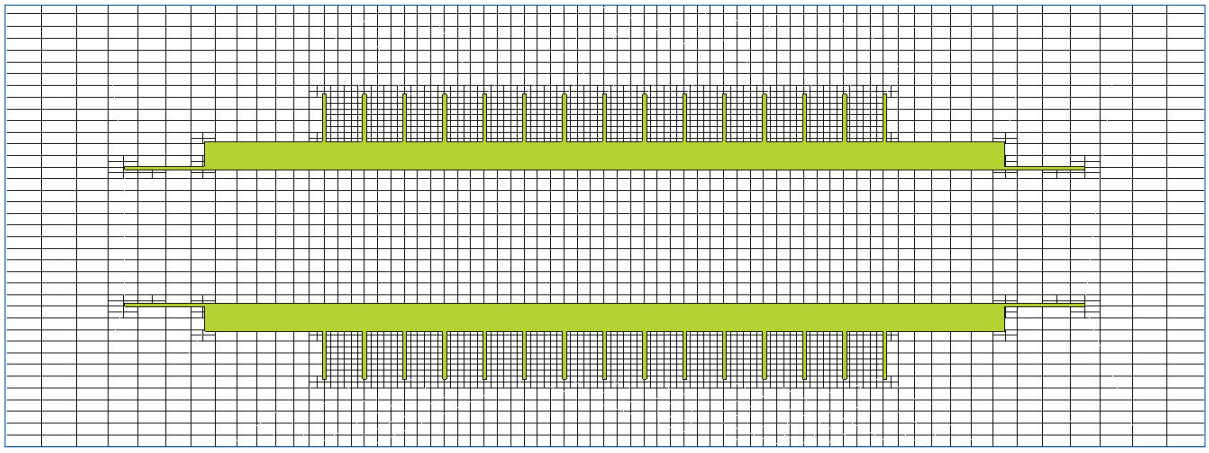


Fig. 16: Initial mesh showing refinement around CAD geometry

The mesh, generated as the output of mesh generator, is called an *initial mesh* (Fig. 16) and it is the set of rectangular cells with computational cells each containing one or more *control volumes* (Fig. 17)).

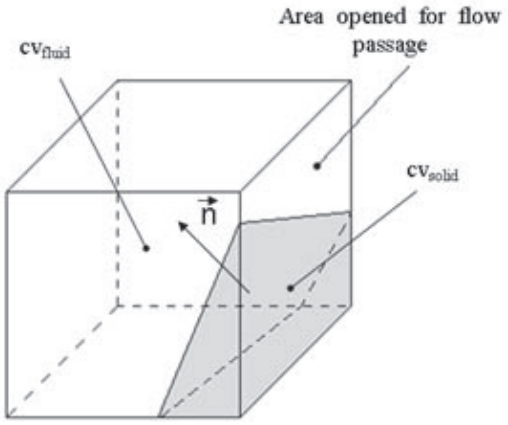


Fig. 17: Cell and control volumes inside

If a solid boundary intersects the cell, the software defines the points of intersection of cell edges with the surface. On the basis of this information, the cell is refined into the set of control volumes. A single cell can contain an arbitrary number of control volumes, with cases where this number can significantly exceed 2. Solid and Fluid control volumes can be alternated several times inside the cell (Fig. 18).

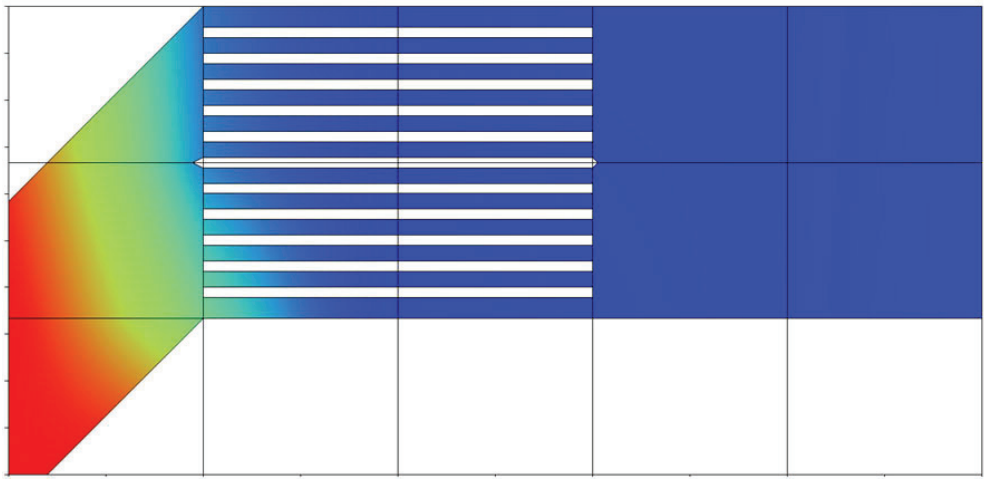


Fig. 18: Multiple control volumes may reside within one cell

For each of these control volumes all necessary geometrical parameters such as volume and the coordinates of cell center are calculated. The areas and normal vector direction are calculated for the cell faces that bound the control volume. The treatment aims to achieve the correct overall pressure drop and heat exchange rather than resolve the detail of what's happening within the individual flow channels. It can be viewed as a set of interconnected 1D CFD solutions implicitly coupled with the surrounding 3D CFD.

During the calculation the initial mesh can be subjected to additional Solution Adaptive Refinement (see Section 3.4), and the octree technique is used to further refine cells or merge refined cells.

3.2 MESH REFINEMENT CRITERIA

The initial mesh generation process anticipates where it will be necessary to have a refined mesh during computation in order to correctly capture the flow physics. Usually it is assumed that regions in the vicinity of sharp edges, narrow channels and any other areas related to specific features of the physical problem definition (such as heat sources, etc.), have to be refined. However, it should be noted that these assumptions can not anticipate, and so do not take into account, other regions that also need a fine mesh to resolve the flow physics such as cavitation, flow separation, and shock capture.

Geometric detail is captured by means of cell refinement. The process is controlled with the help of several criteria, with limitations set on the number of refinement levels for each criterion.

The refinement criterion is an expression that looks like: $C_{split} > \varepsilon_{split}$. If it is true, the cell has to be subdivided. The value of indicator C_{split} is calculated individually for each cell according to an algorithm that controls that type of refinement. The value of ε_{split} is a constant defined automatically, but available to the user to change if desired. Setting refinement levels individually for each criterion makes it possible to influence the total number of cells that will be generated.

CAD, as a source of geometric information, provides a parametric definition of the solid surfaces. In FloEFD, the parametric definition is used to form the surface representation. Use of the CAD kernel's parametric definition ensures that the approximated surface approaches the real surface as the mesh is refined.

This approach has a further advantage over using a triangulated representation of the surface, being that it is possible to obtain the correct surface area and volume of the solid present in a cell, irrespective of how fine the mesh is. As such, this is also an advantage over body-fitted meshes, which require a fine surface mesh in order to correctly capture surface area and solid volume in regions of high geometric curvature.

Refinement criteria can be categorized into the following groups:

1. **Initial Geometry**. Criteria that are used for initial capture of geometry: *Tolerance, Curvature and Small Solid Features*.
2. **Problem Specific**. Criteria that deal with limitations and requirements caused by problem definition in local sub areas: *Cell Type, Solid Boundary,* selected *Volume, Surface* and *Edge* refinements.
3. **Narrow Channels**. To increase the mesh quality in narrow channels, the special technique *(Narrow Channel Refinement)* has been developed. It ensures that an adequate number of cells are included to resolve the flow in the channel.

Each refinement criterion is applied independently, although in some situations different criteria can result in the same refinement. If one or more refinement criteria are satisfied for a given cell, the cell is refined provided that the maximum number of refinement levels has not been exceeded. Surface triangulation, performed by the CAD system, is used to provide the information used in the refinement process.

3.2.1 CURVATURE REFINEMENT CRITERION

The purpose of Curvature Refinement Criterion is to resolve geometrical details like corners and edges. According to this criterion, cells that contain geometry having a high degree of curvature within the cell are refined (Fig. 19). The cell is tagged for refinement when the maximum variation of the angle between vectors normal to the surface triangles is greater than some critical value, and the limit level of curvature refinement has not been reached.

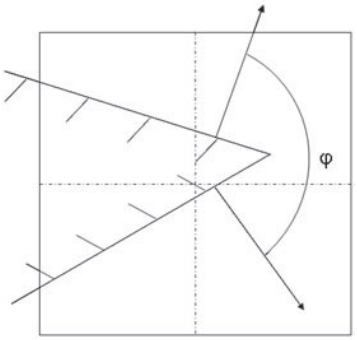


Fig. 19: Curvature Criterion: Refinement is necessary, if normal vector variation within the cell $\varphi > \varphi_{curvature}$ (here: $C_{split} = \varphi$ and $\varepsilon_{split} = \varphi_{curvature}$)

It is often convenient to use curvature refinement in cases where the mesh refinement needs fine-tuning. For example, if it is necessary to generate a more detailed mesh near a blunt angle, which has not been refined sufficiently during the initial capture of geometry.

3.2.2 TOLERANCE REFINEMENT CRITERION

The Tolerance Refinement Criterion aims to improve the resolution of geometrical details like corners and edges within a single cell. This criterion identifies where the interpolated surface, formed from the intersection of the geometry with the faces of the cell, deviates too much from the actual geometry (Fig. 20). This is done by testing whether the maximum distance between the interpolated surface and the most distant vertex of the surface triangles is greater than some critical value. Provided the limit on the number of levels of refinement for this criterion has not been reached, the cell is tagged for refinement.

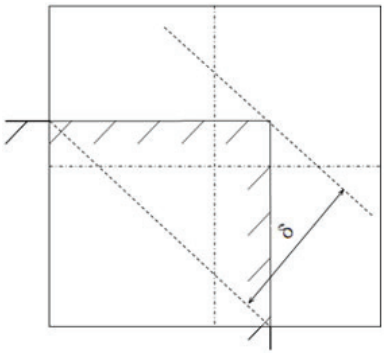


Fig. 20: Tolerance Criterion: Refinement is necessary if $\delta > \delta_{tolerance}$ (here: $C_{split} = \delta$ and $\varepsilon_{split} = \delta_{tolerance}$)

Note there is a difference between the curvature and tolerance refinement criteria. The curvature criterion will force refinement, in regions of high curvature, associated with both blunt and sharp geometry. The tolerance refinement criterion allows the surface resolution to be improved if the interpolated surface does not go "deep enough" into the cell.

Usually, the tolerance criterion value δ can be estimated approximately as ¼ min (d_{gap},d_{wall}), where d_{gap} and d_{wall} are some values that characterize the minimal gap size and wall thickness available in the model.

3.2.3 SMALL SOLID FEATURES

There are several geometrical situations when the criteria described above are insufficient to adequately resolve the geometry. One such example is when a relatively small solid, such as a cylinder or a sphere is fully contained within one of the coarse mesh cells in the base mesh. Note that the tolerance criterion does not apply. Since the geometry does not intersect with the mesh, it is impossible to generate the interpolated surface to test against.

Small Solid Features criterion is used for the initial surface capture before other refinement criteria are applied. The small solid features criterion is the same as the curvature criterion, but with threshold value of $\varphi_{curvature}$=120° fixed in advanced. The angle of 120° is chosen due to the fact that after triangulation of arbitrary 3D solid, the maximum angle between the surface normals for the set of triangles always exceeds this value. In the extreme case of a sphere, where the triangulation creates a regular tetrahedron, the condition $\varphi_{curvature}$=120° is true. As a result, the mesh sensitivity to small geometrical details is controlled just by this criterion alone.

To illustrate how this criterion works it is helpful to consider an example. If, for the case presented in Fig. 21, we deactivate all criteria, the small solid shown becomes invisible to the mesh. If we activate the Tolerance criterion, it does not help the situation, since its operation requires an interpolated surface. If the Small Solid Features criterion is activated and we impose the restrictions for refinement level L_{SSF} = 2, the solid will be captured as shown in (Fig. 22). The size of the cell with refinement level Level=2 is of the same order as the solid characteristic dimension d which is enough for FloEFD to capture the geometry.

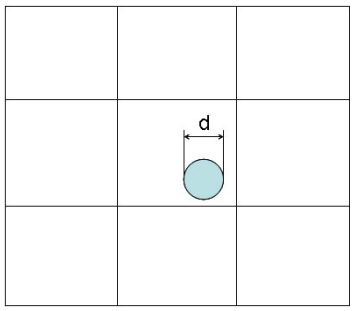


Fig. 21: Body with the typical size d is invisible for the mesh Level=0

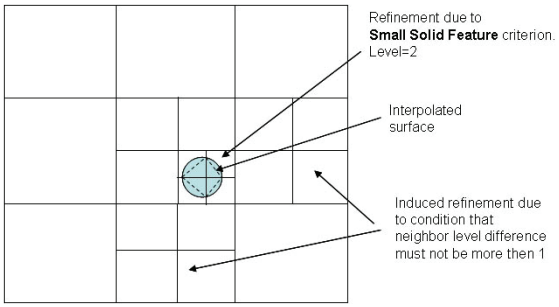


Fig. 22: $Level_{SSF}$=2 cells are the same order of magnitude as the body size, d, small enough to capture the body

In theory the same result could be obtained using the Curvature criterion with the criterion value $\varphi_{curvature}$ set explicitly to 120°. However, using the Small Solid Features criterion makes it possible to distinguish the primary process of coarse mesh capture of small solid features from the finer operations of the curvature and tolerance criteria, which are activated once the small solid geometry has already been captured. This allows these latter criteria to be adapted in accordance with the physical formulation of the specific problem.

The operation of this criterion also helps in other situations (Fig. 23) that are important for proper capture of the important details of the surface.

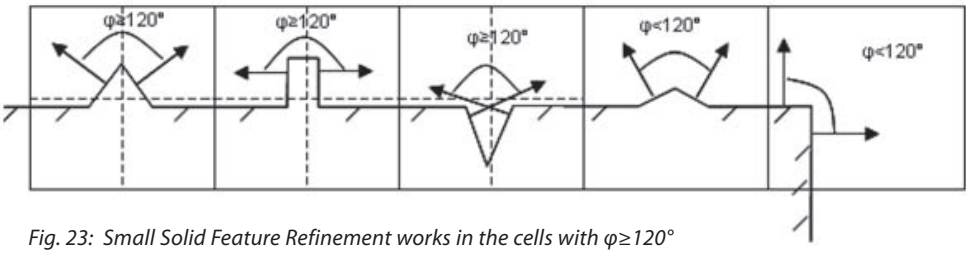


Fig. 23: Small Solid Feature Refinement works in the cells with φ≥120°

The most important property of this criterion is that it generates refined cells only in the case where something exists inside the cell that is otherwise invisible to the mesh, or if there is some topologically important feature present on the surface. By using this criterion it is possible to generate a topologically correct mesh with the minimum number of cells.

3.2.4 REFINEMENT IN LOCAL REGIONS

The local mesh settings option is one more tool that helps to create an optimal mesh. The application of local mesh settings is especially beneficial, if you are interested in resolving a particular region within a complex model.

The local mesh settings can be applied to a component, being an assembly, part, face, edge or vertex. To apply the local mesh settings to some region, it is necessary to specify this region as a 'virtual' solid part or sub-assembly (CAD systems allow the creation of 'disabled' parts for this purpose) or to select an existing part or sub-assembly in the model. The local mesh settings are applied to cells intersected by the selected component, be it an assembly, part, face, edge, or just the single cell enclosing the selected vertex.

3.2.5 NARROW CHANNEL REFINEMENT

The term 'narrow channels' is conventionally used to describe a flow passage where the opposite walls are relatively close to each other (Fig. 24).

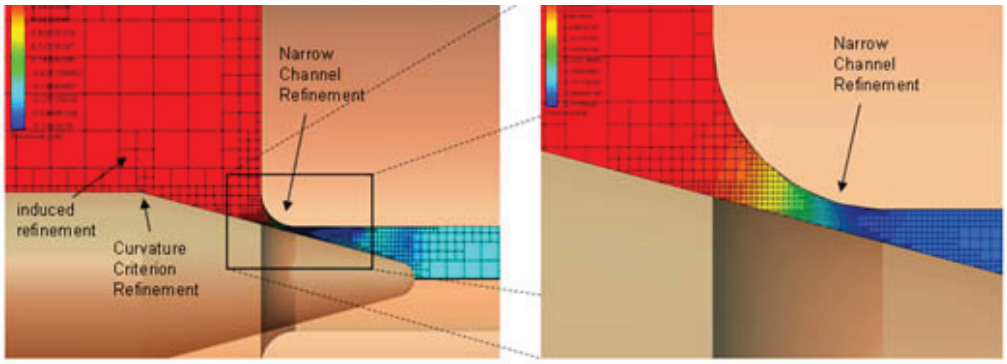


Fig. 24: Mesh refinement in narrow channel

The basic concept of the narrow channel refinement is to resolve the narrow channels with a sufficient number of cells to provide a reasonable level of solution accuracy. It is especially important to resolve the narrow channels in analysis of low Reynolds number flows and analyses involving long channels, where the boundary layer thickness becomes comparable with the size of the cut cells.

The mesh generator attempts to set the effective number of cells across the narrow channel as close as possible to the predefined target value of the approximate number of cells criterion (N_{gap}). Both the approximate number of cells criterion and the narrow channels refinement level (L) influence the mesh in narrow channels. The base mesh in the narrow channels will be split to have N_{gap} cells per channel width, provided that the refinement level of the cells generated in the narrow channel does not exceed L. The limit on the number of refinement levels is necessary to avoid the undesirable mesh splitting in very fine channels that may cause an unreasonably large number of cells.

This refinement criterion stands apart from the other criteria as it is not based on the local properties of cells intersected by solid geometry. Rather it uses a ray tracing method to detect the number of cells between solid surfaces that form a narrow channel and then tags these cells for refinement where necessary.

Other control parameters used with this refinement criterion are the minimum and maximum heights of the narrow channels. These define the range of channels to which this criterion is applied. Outside this range the flow passage will not be considered as narrow and so will not be refined according to this criterion.

3.3 AUTOMATIC MESH SETTINGS

As noted above, FloEFD's octree mesh generator uses a system of refinement criteria, each with their own values and refinement levels. This is relatively straightforward compared to body-fitted mesh generators. However, in order to further simplify the process of defining the mesh a method to automatically define the refinement criteria, values and levels for a case Automatic Parameters Definition (APD) technology has been developed. There have been a number of improvements since the initial deployment of FloEFD's APD technology from 1998-2000, most notably thin wall resolution. Consequently, the evolution of this approach requires a more lengthy description.

The main idea of the APD technology is to define both the base mesh and refinement settings from the physical definition of the model's boundary conditions, etc. together with the most general information about the geometry.

APD uses the following input parameters:

1. Problem type (Internal\External, Compressible\Incompressible, 3D\2D)
2. Minimum gap size (MGS) d_{gap}
3. Minimum wall thickness (MWT) d_{wall}
4. Level of Initial Mesh or Result Resolution Level (RRL)
5. Computational model bounding box
6. Fluid region bounding box
7. Symmetry settings applied to the computational domain boundaries
8. Middle size of the computational model
9. Wind direction for external flows.

APD output:

1. Base mesh
    a. Computational domain box B_{CD}
    b. Key planes sets for x,y,z direction and cells ratios at these planes
    c. Number of cells for x,y,z direction N_x, N_y, N_z.

2. Refinement settings
    a. Small solid feature refinement level L_{SSF}
    b. Tolerance refinement level L_{tol} and tolerance d_{tol}
    c. Narrow channels refinement level L_{NCR} and the number of cells per channel width N_{gap}

The default value of MGS is the smallest size of all flow openings or faces where solution goals have been specified to control convergence of the solver.

If set manually, MGS should be set to the smaller of the size of smallest flow passage to be resolved by the mesh, and the size of smallest geometric object in the flow that needs to be resolved. FloEFD is then able to calculate the refinement level needed to resolve this geometrical feature.

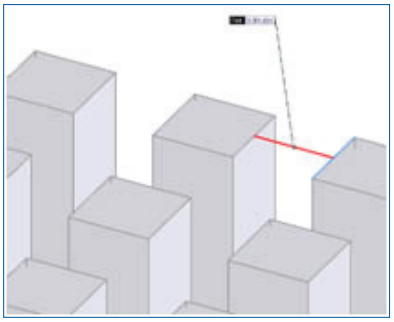


Fig. 25a: Minimum gap size

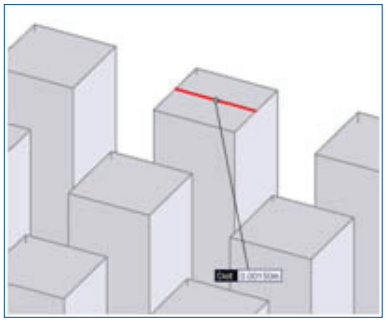


Fig. 25b: Minimum wall thickness

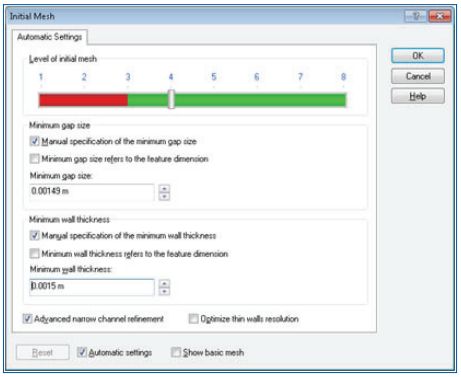


Fig. 25c: Mesh setting window

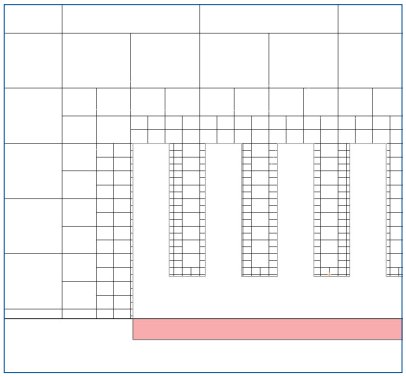


Fig. 25d: Mesh view

Fig. 25: Automatic Initial Mesh Definition by using minimum gap size, minimum wall thickness and level of initial mesh in order to calculate the needed levels of mesh refinement

The APD technology captures experience gained from manual mesh creation to provide values for all the necessary mesh refinement parameters for each Result Resolution Level. In the case of the initial mesh, the Result Resolution Level is the Level of Initial Mesh shown in Fig. 25(c) above. The Result Resolution Level may change during the solution process due to the use of Solution Adaptive Refinement in FloEFD.

3.4 SOLUTION ADAPTIVE REFINEMENT

The fundamental goal of Solution Adaptive Refinement (SAR) is to optimize the mesh distribution in order to minimize the spatial error arising from the discretization of the governing differential equations. In FloEFD this optimization process is implemented as part of the global iteration process.

For a steady-state calculation, SAR cycles are implemented at a certain iteration frequency. Iterations are chosen on the basis of the estimated residual decrease at current iteration in comparison with residuals calculated after the previous SAR cycle. For transient cases, SAR cycles are implemented at certain moments in time. The time approximation of conservation laws and difference scheme conservatism are ensured. In addition, history dependent refinement is applied for such problems: once any cell has been sub-divided, the generated sub-cells cannot be re-merged. This approach makes it possible to refine the whole solution domain, where potentially any significant disturbances of the flow can appear.

At each SAR cycle the refinement level *l* is increased by 1 unless the certain predetermined maximum value is achieved. Cells that have been previously refined can be merged thus, reducing the refinement level by 1. Two parameters, calculated from the local flow conditions, control the action of the SAR at a particular grid cell: a refinement indicator C_{split} and an unrefinement indicator C_{merge}.

The C_{split} and C_{merge} parameters are calculated from the gradients present in the cell and in its immediate neighbours, so that C_{split} takes into account not only the local gradient but also the number of neighbour cells and their size. To damp out numerical oscillations in the variable fields as the solution progresses, data used to calculate C_{split} and C_{merge} is collected over a number of iterations, and the collection does not start until the case has partially converged.

If C_{split} exceeds a certain predetermined value ε_{split} then the LTE is expected to be high for the considered cell, and the cell is refined. For the set of 8 daughter cells the unrefinement indicator C_{merge} is introduced and its value calculated at the next SAR cycle. If C_{merge} is then calculated to be less than a certain predetermined value ε_{merge} the LTE is expected to be low for the octet of cells and so they are merged. This approach was initially developed with 1st order indicators and applied to propulsion systems (see Gavriliouk et al. 1993, 1994a-1994c, Krulle et al. 1994 Barbashov et al. 1995, Hagemann et al. 1996, and Fey 2001), before being and extended to other applications.

Just as with the initial mesh, to exclude the significant mesh irregularity in the regions of refinement the difference between the refinement levels of neighbor cells is limited to 1.

An example that demonstrates the efficiency of SAR in FloEFD is the stationary solution of the problem of supersonic ideal inviscid gas flow around the sphere with radius of 1 (M=2 and y=1.4). Only one quarter of the sphere, cut by two planes of symmetry, is considered. Fig. 26 displays the mesh and pressure distribution in the plane of symmetry. Refinement levels are l=0,1,2,3. It is seen that if l is increased, the resolution of the shock wave and the expansion fan are improved significantly and the lines of constant pressure correctly approximate circles. The information presented in Table 2 quantifies the efficiency of the technique.

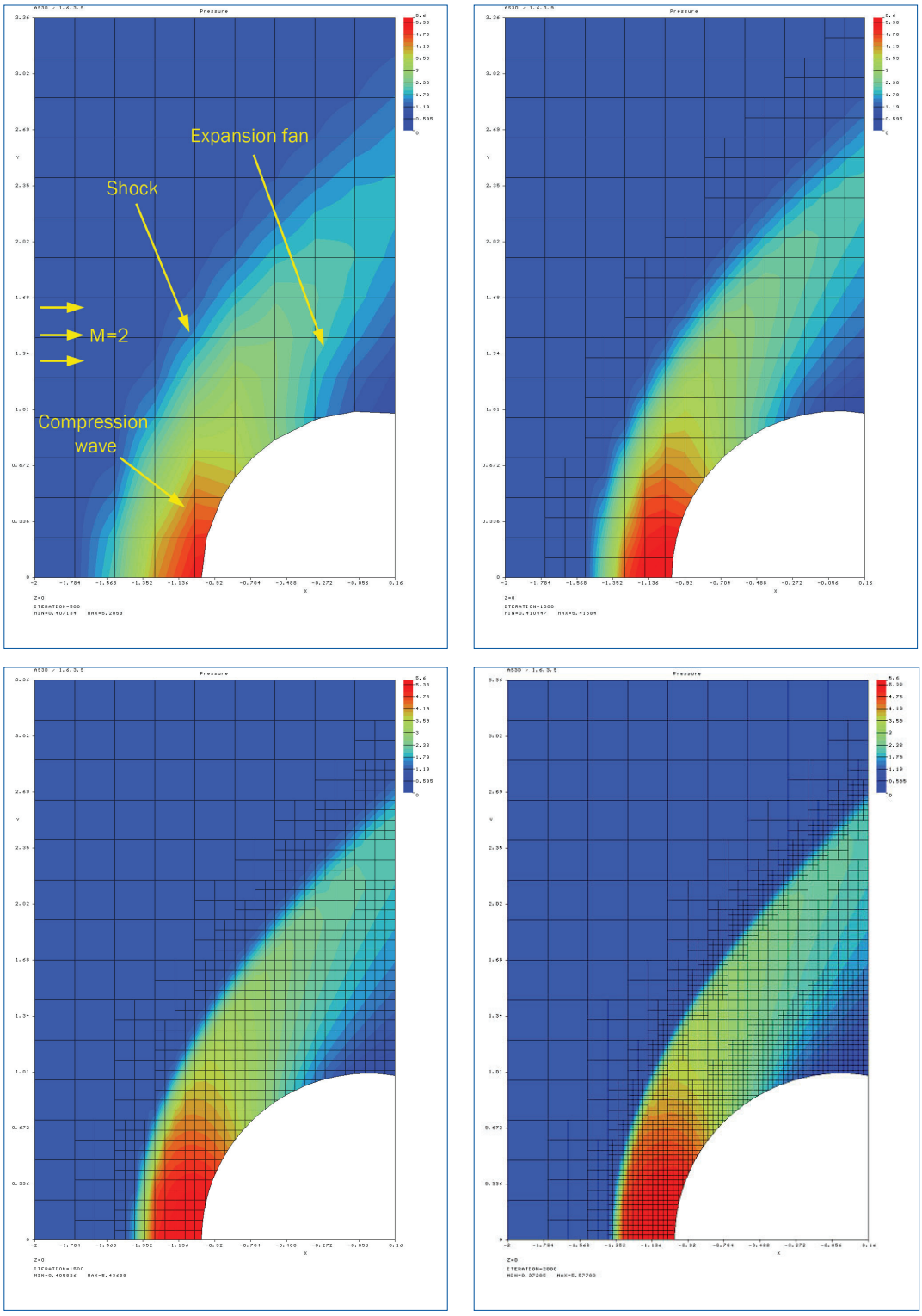


Fig. 26: Mesh and Pressure distribution calculated for supersonic flow of ideal inviscid gas around a quarter of the sphere. A Solution Adaptive Refinement was applied with the refinement levels of 0, 1, 2, 3. The images show the flow at symmetry plane

TABLE 2

Refinement Level, *l*	Total number of cells	Number of cells between shock nd spreak point	C_d	Error %
0	2250	1.5	0.80374	1.86
1	7269	3	0.76207	-3.1
2	33316	6	0.78164	-0.93
3	143069	12	0.78672	-0.28
High accuracy solution result (Lubimov & Rusanov, 1970):			**0.7890**	

In Table 2 for each refinement level *l* the drag coefficient C_d for the front half (up to the angle of 90° relatively longitudinal axis) of the sphere is presented. C_d, calculated by FloEFD, is compared with accurate solution obtained according to Lyubimov and Rusanov's precise method [0]. It is seen that as *l* is increased from 0 to 3, the solution accuracy rises and by refinement level *l*=3 it is below 1%. To obtain the same accuracy without SAR, i.e. using a uniform mesh, it would be necessary to use 1,152,000 cells - **8 times** higher than used above.

These results show that the efficiency of SAR for external problems in FloEFD is very high.

4 INDUSTRIAL EXAMPLES

4.1 PIN-FIN HEAT SINK

Free convection cooling performance of a pin-fin heatsink has been studied experimentally (Yu & Joshi, 2002). The case of 9x9 square pin-fin array presented in this paper has been investigated via FloEFD (Balakin & Churbanov, 2004) and compared with the above measurements. The model exactly reproduces the experimental conditions. It consists of two rectangular Plexiglas enclosures, one inside another. The internal enclosure contains an aluminum pin-fin heatsink mounted on a heated component (see Fig. 27).

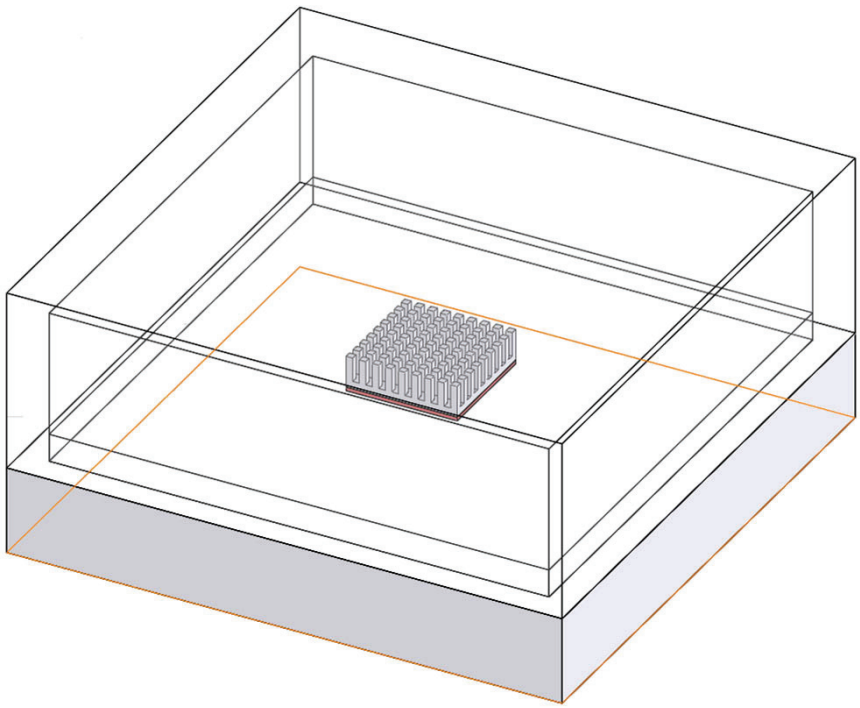


Fig. 27: A close-up of the model: the internal enclosure with 9x9 pin-fin array over the heating component flush mounted on the bottom

Two configurations are simulated. One with the heatsink mounted with the base horizontal, fins upward, and the other with the base vertical. Thermal conduction within the solid regions is modeled together with convection and radiative heat transfer.

Fig. 28 shows a comparison of experimental visualization (image on right) with predicted flow pattern for heat generation rate $Q = 1$ W (central image) on a vertical plane through the center of the heatsink for the model with the base mounted vertically. The mesh, shown in the image on the left, is obtained by SAR, and consists of about 145,000 cells. The images show a high level of agreement between simulation and experiment.

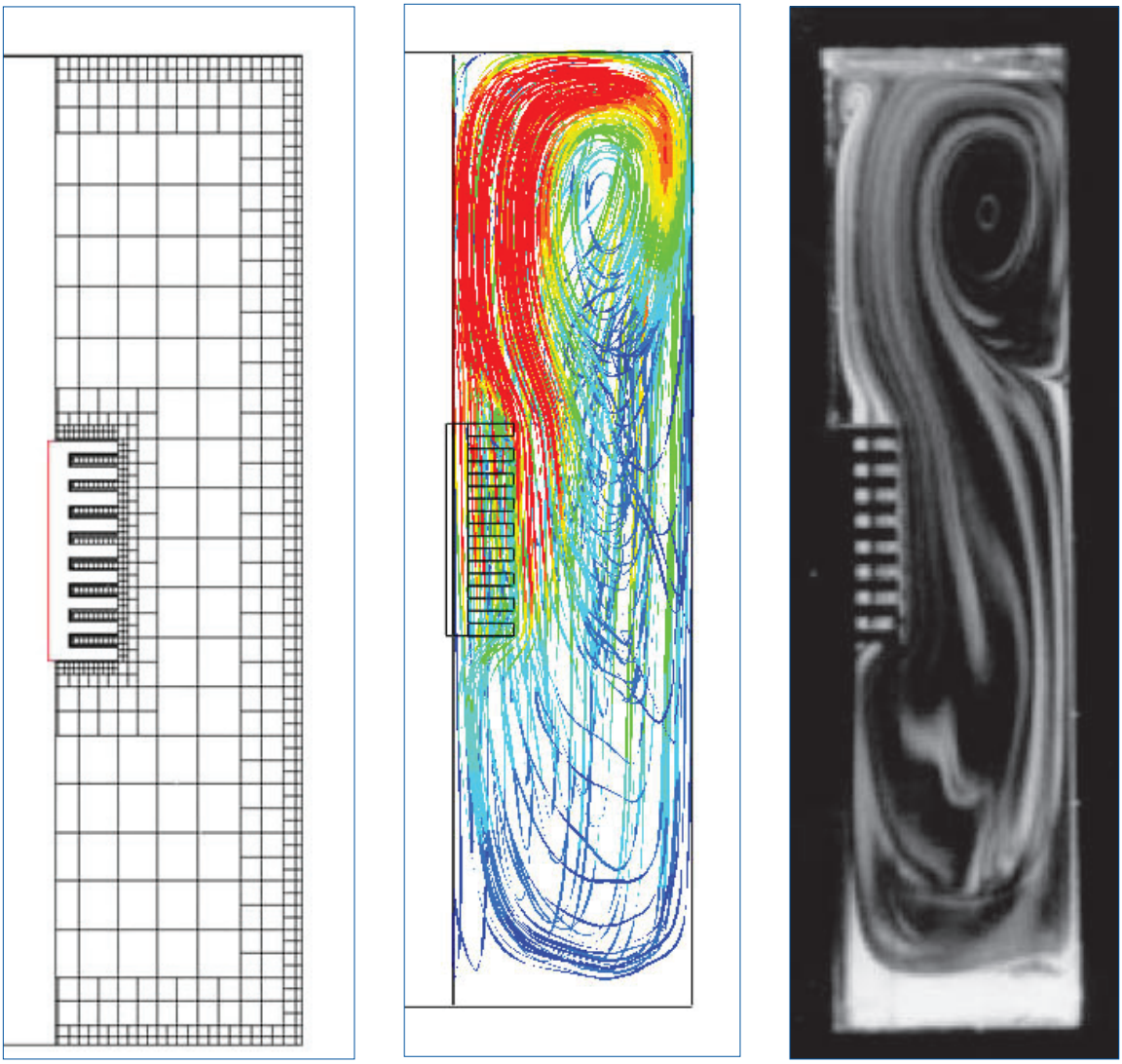


Fig. 28: Mesh view (a), predicted flow trajectories colored by velocity magnitude (b) experimental visualization (c), Q = 1 W, vertical case, z = 0

The next set of images show the comparison of the predicted flow pattern with experiment for the heatsink mounted horizontally. Fig. 29, features a comparison of the experimental visualization (bottom image) with the predicted flow patterns for heat generation rate $Q = 0.5$ W (center image) on a vertical plane through the center of the heatsink. The mesh, obtained using SAR is shown in the top image and consists of about 142,000 cells.

A comparison of predicted and measured thermal resistance for considered cases shows that numerical results are within 5% of experimental data – an excellent correlation between computations and measurements. Note that the experimental results for the horizontal case are slightly non-symmetrical. This is due to the difficulty in maintaining uniform boundary conditions on the outer Plexiglas enclosure.

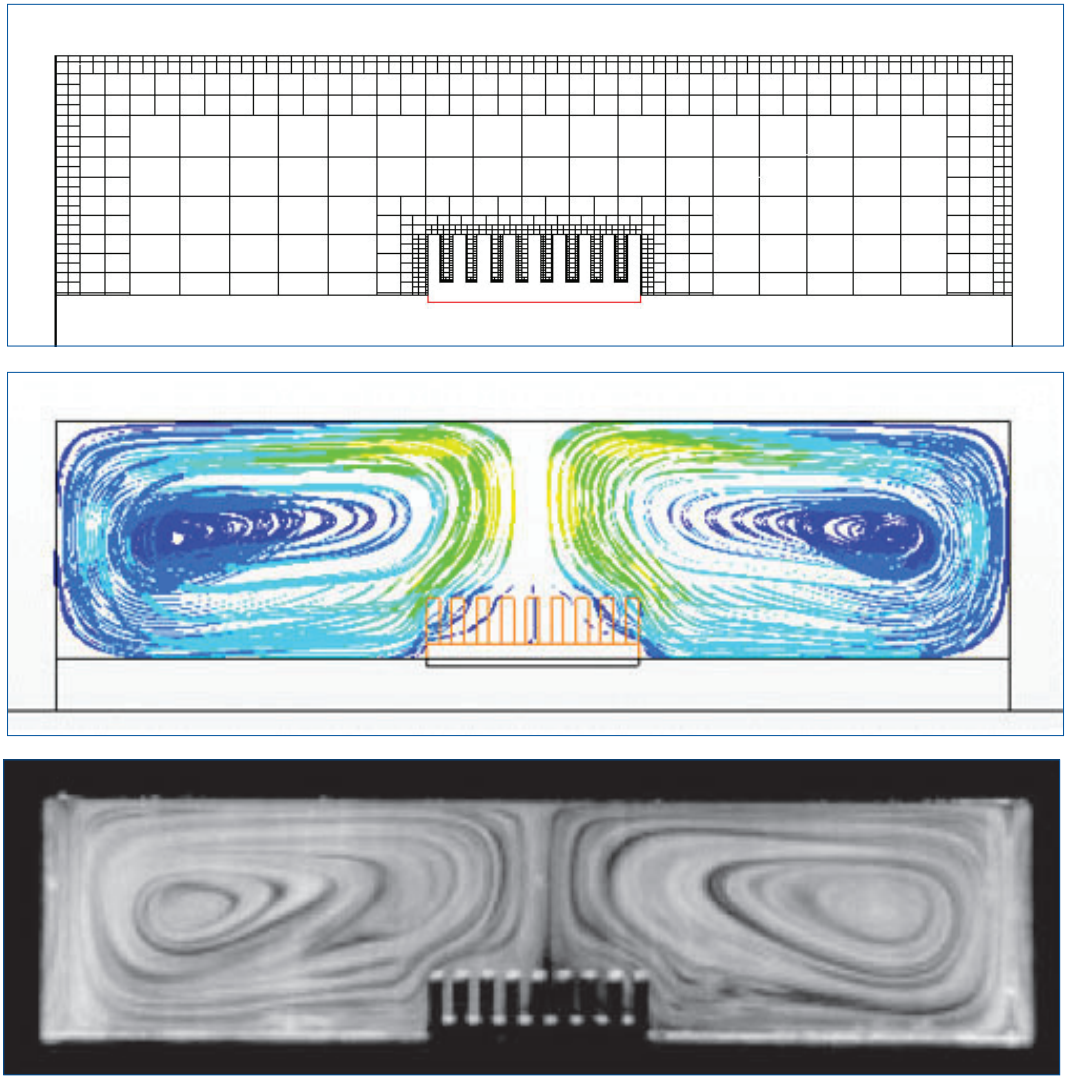


Fig. 29: Visualization from [0], Q = 0.5 W, horizontal case, z = 0 m

4.2 ARIAN VULCAIN ENGINE FLOW SIMULATION

The nozzle jet of Arian rocket Vulcain is considered. The pressure ratio is p_c / p_a=100. Fig. 30 shows the Mach number field (Fey, 2001) computed by FloEFD and an appropriately adapted Cartesian mesh, which consists of about 156,000 cells.

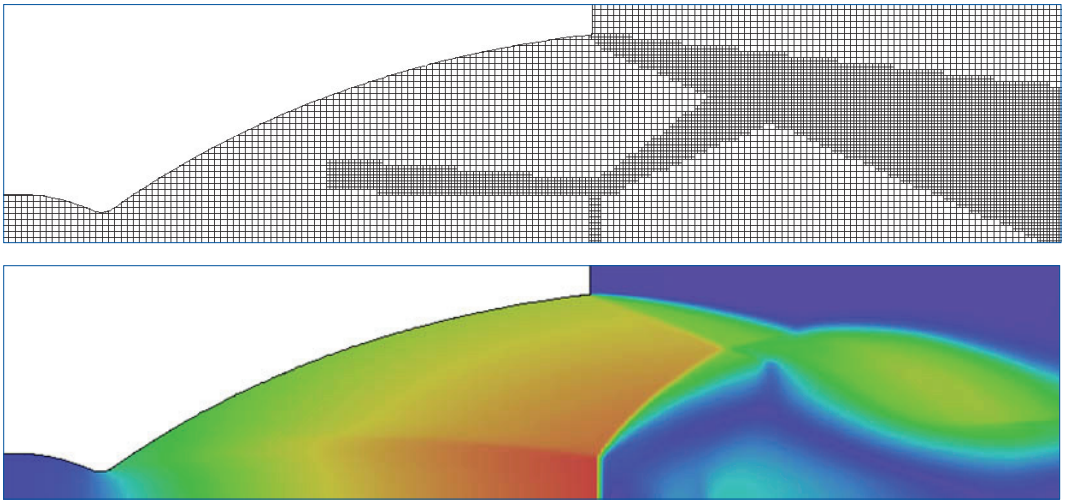


Fig. 30: Mach number field in Vulcain nozzle and appropriate adapted Cartesian mesh computed by FloEFD technique (Fey, 2001). Pressure ratio is p_c / p_a =100

A very high level of accuracy is observed for FloEFD:

- the high resolution of the shock generated by discontinuity of the 2nd derivative of the profile of the nozzle wall on the line of interface of its mouth and extending part; which is a pre-requisite for
- accurate computation of Mach disc size and position, being at the exit plane of the extending part; and
- accurate calculation of integral parameters: specific mass flow and specific impulse with relative error lying within 0.1% .

4.3 FLOW AROUND SPACECRAFT "PROGRESS-M" NOSE

Numerical simulation of the supersonic flow around the nose part of the spacecraft "Progress" refers to the full model of the vehicle geometry, which includes superstructures (Fig. 31), under external pressure of about 1 Pa, is provided by FloEFD (Krylov *et al.*, 2002) and the direct simulation with the Monte Carlo (DSMC) method (Ivanov *et al.*, 1998).

Fig. 32 displays the final FloEFD adapted mesh, which consists of about 1,000,000 cells, and Fig. 33 shows the pressure fields computed by FloEFD (left) and the DSMC method (right).

Fig. 31: Spacecraft geometry with superstructures. Pressure gauges P1 and P2 are located and seen on the right side of the surface

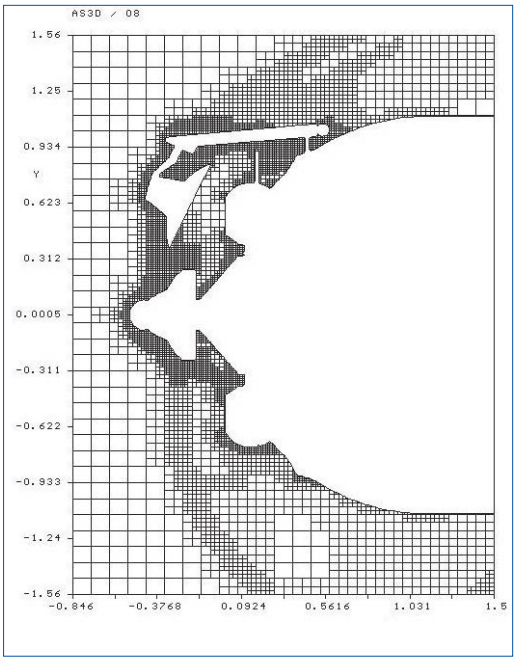
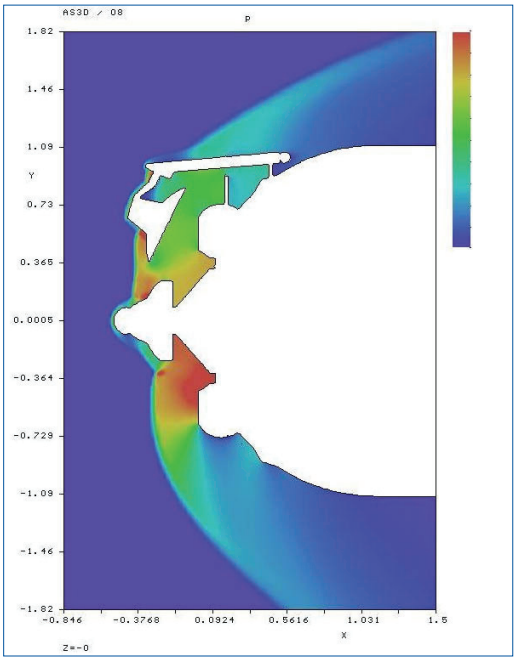


Fig. 32: The final FloEFD adapted mesh



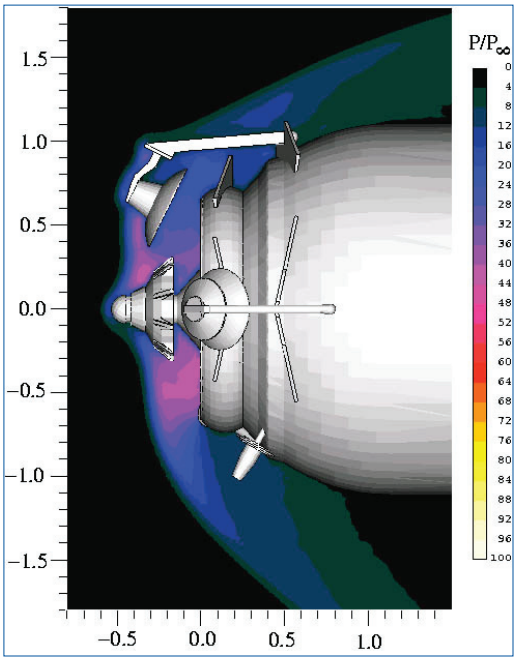


Fig. 33: Pressure fields computed by FloEFD (Krylov et al., 2002 [left]) and by direct simulation with the Monte Carlo (DSMC) method (Ivanov et al. 1998 [right])

Comparison of these fields shown in Fig. 33 shows they correspond very closely.

Table 3 presents the relative pressures in gauges P1 and P2 at the moments: t=166.5s and t=171.5s computed by FloEFD (Krylov *et al.*, 2002) and by direct simulation with the Monte Carlo (DSMC) method (Ivanov *et al.*, 1998).

TABLE 3

Time	P1	P2
t=166.5s	23.5 (DSMC) 23.6 (FloEFD)	42.5 (DSMC) 47.2 (FloEFD
t=171.5s	11.5 (DSMC) 12.7 (FloEFD)	25.5 (DSMC) 28.5 (FloEFD)

It can be seen that the FloEFD data and DSMC data are in very good agreement: their difference is less than 10%.

5 CONCLUSIONS

The general purpose CAD-embedded CFD solver in the FloEFD software package from Mentor Graphics uses an octree based mesh technology, combined with a unique immersed boundary approach for wall friction and heat transfer. Totally free from mesh distortion, this Cartesian-based approach has been shown to deliver the lowest error when the Navier-Stokes equations are discretized onto the mesh. FloEFD's octree refinement allows far more rapid changes in mesh density than can be achieved with tetrahedral meshes, reducing computation time. Octree is particularly suited to Solution Adaptive Refinement for example to capture shocks in high-speed flows. Excellent agreement has been shown for several industrial examples that have included: buoyancy driven flow, conjugate heat transfer (convection, conduction and radiation), and hypersonic flow with shock capture.

6 REFERENCES

Aftosmis, M.J. and Berger, M.J. 2002. Multilevel Error Estimation and Adaptive h-Refinement for Cartesian Meshes with Embedded Boundaries. AIAA 2002-0863. *40th AIAA Aerospace Sciences Meeting and Exhibit*. 14-17 January, Reno NV.

Baker, T.J., 1989. Automatic Mesh Generation for Complex Three-Dimensional Regions Using a Constrained Delaunay Triangulation, *Engineering with Computers*, Vol. 5, pp.161-175

Balakin, V. Churbanov, A. Gavriliouk, V. Makarov, M. and Pavlov, A., 2004. Verification and Validation of EFD.Lab Code for Predicting Heat And Fluid Flow, *Proceedings of CHT-04 ICHMT International Symposium on Advances in Computational Heat Transfer*, April 19-24, Norway, CHT-04-179

Barbashov, E. Gavriliouk, V. Guriev, V. Isaev, V. Losenkov, A. Odintsov, E.V. Sergienko, A.A. Sobachkin, A.A., 1995. Mathematical Modeling of Propulsion Systems, IAF 95-S.1.05, *46th International Astronautical Congress,* October 2-6, Oslo, Norway

Berger, M.J. Aftosmis, M.J. 1998. Aspects (and Aspect Ratios) of Cartesian Mesh Methods, *Proceedings of the 16th International Conference on Numerical Methods in Fluid Dynamics*, To Appear in "Lecture Notes in Physics", Springer–Verlag, Heidelberg Germany, 6–10 July, Arcachon, France

Delaunay, B.N., 1934. Sur la Sphere Vide. *Izvestia Akademia Nauk SSSR, VII Seria, Otdelenie Matematicheskii i Estestvennyka Nauk* Vol. 7 pp.793-800

Filipiak, M., 1996. Mesh Generation, Edinburgh Parallel Computing Centre, The University of Edinburgh, Version 1.0, November 1996

Frey, M. 2001. Behandlung von Strömungsproblemen in Raketendüsen bei Überexpansion, *Institut für Aerodynamik und Gasdynamik* Universität Stuttgart, 2001

Gavriliouk, V.N. Denisov, O.P. Nakonechny, V.P. Odintsov, E.V. Sergienko, A.A. Sobachkin, A.A., 1993. Numerical Simulation of Working Processes in Rocket Engine Combustion Chamber, *44th Congress of the international Astronautical Federation*, IAFF-93-S.2.463, October 16-22, Graz, Austria.

Gavriliouk, V.N. Lipatnikov, A.V. Kozlyaev, A.N. Odintsov, E.V. Sergienko, A.A. Sobachkin, A.A. 1994a. Computation Modeling of the Combustion Problems with the use of "AeroShape-3D". *Numerical Technique-ISTS* 04-d-27

Gavriliouk, V.N. Krulle, G. Schley, C.-A. and others, 1994b. Numerical simulation of combustion process in rocket combustion chambers with coaxial injection, *AIAA Paper 94-3329*.

Gavriliouk, V. Krulle, G. Odintsov, E. Schley, C.-A. Sergienko, A. Sobatchkine, A., 1994c. Numerical Simulation of Combustion Process in Rocket Combustion Chambers with Coaxial Injectors, AIAA 94-3329, *30th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, June 27-29, Indianapolis, IN

Hagemann, G. Schley, C.-A. Odintsov, E. Sobatchkine, A., 1996. Nozzle flowfield analysis with particular regard to 3D-plug cluster configurations, *32nd AIAA, ASME, SAE, and ASEE, Joint Propulsion Conference and Exhibition*, Lake Buena Vista, FL, July 1-3

Ivanov M.S., Markelov G.N., Gimelshein S.F., 1998. Statistical simulation of reactive rarefied flows: numerical approach and applications, *AIAA Paper 98-2669*.

Krulle, G. Gavriliouk, V. Schley, C.-A. Sobachkin, A., 1994. Numerical simulation of Aerodynamic Processes and its Applications in Rocket Engine Problems, *45th Congress of the international Astronautical Federation*, IAFF 94-S2.414, October 9-14, Jerusalem, Israel.

Krylov, A.N. Kotov, V.M. Tokarev, V.A. Shcherbakov, N.A. Khokhlov, A.V. Ivanov, M.S. Vaschenkov, P,V Kashkovsky, A.V. and Markelov G.N. 2002. Numerical Modeling and Experimental Data Analysis of the Flow Near Spacecraft "Progress-M" Nose After the Head Fairings Jettisoning. *ESA SP* Vol. 487 pp 307-314

Lawson, C.L., 1977. Software for C1 Surface Interpolation, *Mathematical Software III*, pp.161-194

Löhner, R. Cebral, J. Castro, M. Baum, J.D. Luo, H. Mestreau, E. and Soto, O., 2004. Adaptive Embedded Unstructured Grid Methods, *Mecánica Computacional*, Vol. XXIII, pp. 29-42, G.Buscaglia, E.Dari, O.Zamonsky (Eds.), Bariloche, Argentina, November.

Lubimov, A.N. Rusanov, V.V. 1970. Techenija gasa okolo tupykh tel. *Gas flows around blunt bodies*, Vol. I. Nauka Press, Moscow (in Russian)

Mentor Graphics Corp., 2011. Enhanced Turbulence Modelling in FloEFD™

Parry, J. and Tatchell, D., 2008. Flomerics' EFD Meshing Technology: A White Paper

Thompson J.F, Warsi Z.U.A, Martin C.W., 1985. *Numerical Grid Generation: Foundations and Applications*. North-Holland: New York, NY.

Watson, D.F., 1981. Computing the Delaunay Tesselation with Application to Voronoi Polytopes. *The Computer Journal*, Vol. 24(2) pp.167-172

Weatherill N.P. and Hassan O., 1994. Efficient Three-dimensional Delaunay Triangulation with Automatic Point Creation and Imposed Boundary Constraints, *International Journal for Numerical Methods in Engineering*, Vol. 37, pp.2005-2039

Wang Z.J. and Srinivasan, K., 2002. An adaptive Cartesian grid generation method for 'Dirty' geometry, *International Journal for Numerical Methods In Fluids*.

Yu, E. and Joshi, Y., 2002. Heat Transfer Enhancement from Enclosed Discrete Components Using Pin-Fin Heat Sinks. *Int. J. Heat Mass Transfer*, Vol. 45, No. 25, pp 4957-4966.

	Nomenclature
BREP	Boundary Representation
C_d	Drag coefficient
C_{merge}	Unrefinement indicator value in Refinement Criterion
C_{split}	Refinement indicator value in Refinement Criterion
d_{gap}	Minimal gap size (MGS)
d_{wall}	Minimal wall thickness (MWT)
h	Mesh step size, i.e. cell size
l	refinement level
L	Mesh refinement level
M	Mach number
LTE	Local Truncation Error
n	Mesh convergence coefficient
N	Number of cells
NURBS	Non-Uniform Rational B-Spline
p	Static pressure
q	Heating power
RRL	Result Resolution Level
SAR	Solution-Adaptive Refinement
t	time
δ	distance between the interpolated surface and the most distant point of the surface
$\delta_{tolerance}$	threshold value of δ in Tolerance Refinement Criterion
ε_{merge}	Unrefinement threshold value in Refinement Criterion
ε_{split}	Refinement threshold value in Refinement Criterion
γ	Ratio of specific heats (Cp/Cv)
φ	Value of an angle
$\varphi_{curvature}$	Threshold value of φ in Curvature Refinement Criterion

	Subscripts
a	ambient (pressure)
c	chamber (pressure)
gap	across the gap
i, i-1	cell indices
SSF	Small Solid Features Refinement Criterion
Tol	Tolerance Refinement Criterion
x, y, z	x, y, and z directions respectively

Corporate Headquarters
Mentor Graphics Corporation
8005 SW Boeckman Road
Wilsonville, OR 97070-7777
Phone: 503.685.7000
Fax: 503.685.1204

Sales and Product Information
Phone: 800.547.3000
sales_info@mentor.com

Visit www.mentor.com/company/office_locations/ for the list of Mechanical Analysis Division Offices

MGC 02-11 TECH9690-w