Local Solution-Based Mesh Adaptation for Unstructured Grids in Complex 3D Domains

Hoang-Huy Nguyen[1], Vinh-Tan Nguyen[1] and Oubay Hassan[2]

[1] Institute of High Performance Computing, 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632 (nguyenhh@ihpc.a-star.edu.sg).
[2] College of Engineering, University of Wales, Swansea SA2 8PP, UK (o.hassan@swansea.ac.uk)

1 Introduction

In an effort to improve the robustness and efficiency of mesh adaptation, this work proposes an adaptive *re-meshing* method that can be used for variety of problems in CFD. It should be noted that this method is feature driven adaptivity and that it does not use strategies such as edge split, edge collapse, edge swap, etc as were employed in mesh modifications [2]. Instead, it defines the "holes" locally based on an output-based indicator and re-meshes these regions independently. The surface re-meshing is performed using advancing front algorithm and volume re-meshing is based on Delaunay triangulation. The proposed re-meshing method is completely automatic and no user intervention is required. In addition, it can perform mesh adaptation robustly and effectively for complex configurations, especially for domains with curved boundaries. The numerical examples show that the use of the proposed adaptation strategy in three dimensions offers a great potential in having low cost CFD simulations with high quality mesh.

2 Local Mesh Adaptation

The adaptive re-meshing strategy presented in this paper is to re-mesh the regions where the current mesh resolution is not sufficient to capture steep gradients in localized portions of the flow. In this approach, we will re-mesh edges, surfaces, volumes locally based on spacings derived from solutions. At the same time, it is also important to preserve the global geometrical definition of the domain boundaries, especially with curved surfaces.

2.1 Solution-Based Adaptivity

In the present method, a minimum spacing δ_{min} and a maximum spacing δ_{max} are specified to control the distribution of the new mesh. To define a metric for the mesh, we employed Gaussian distribution based on δ_{min}, δ_{max} and the solution gradient $|\nabla u|_P$ (density, pressure, etc). The size function at node P is defined as follows:

$$\delta_P = \delta_{min} + (\delta_{max} - \delta_{min}) e^{-\frac{(|\nabla u|_P - |\nabla u|_{min})^2}{2c^2}} \tag{1}$$

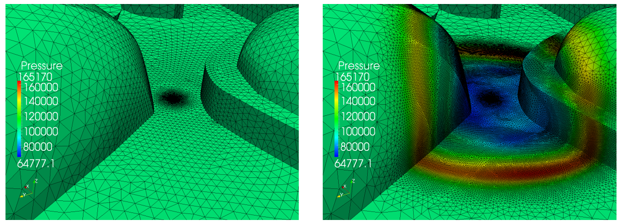
The parameter c, controlling the width of the Gaussian distribution, defines the regions surrounding the high solution gradient where fine meshes are needed (refining). On the other hand, areas where the solution gradient is small will be re-meshed with δ_{max} (coarsening). The mesh will be adapted in the areas where the percentage change between the current mesh spacing and the new one is greater than a certain threshold (ϵ_d). In addition, it is straightforward to define new mesh size using a multifield criteria (for example adapt simultaneously on velocity and pressure). We can use Eq. 1 to obtain new mesh size for each field. Then at each node, the minimum value between the new mesh sizes is used for mesh adaptation.

2.2 Re-meshing algorithm

Elements which are in the adapted regions are marked for deletion. The deleted elements are grouped into 'holes'. Subsequently, these holes are re-meshed with the new nodal spacings which involves the following processes such as: local edge discretization, local surface discretization and local volume regeneration.

Each hole is bordered by a collection of faces in three dimensions which defined the outer hull of the hole to be re-meshed. The collection of these faces will define local geometrical surfaces which are needed to re-meshed as part of the adaptation process. For the local boundary surface re-meshing, it is important to preserve the curvature of the curves and surfaces of the boundaries. Therefore, during the re-meshing processes, it is necessary to refer to the global boundary definitions to ensure that the new generated nodes are placed on the true geometrical curve and surface definition. Fig. 1(a) and Fig. 1(b) show an example of local curves and surfaces re-mesh for the explosion in urban areas. The initial mesh used is depicted in Fig. 1(a). The adapted mesh at a certain time of the simulation is shown in Fig. 1(b). As we can see in the figure, local curves and surfaces are re-meshed with the new mesh size accordingly but still preserve the curvature of the domain boundaries.

The collection of faces (triangles) in three dimensions including new generated faces obtained from local surface re-mesh and surrounding each hole is now regarded as an initial front for the Delaunay triangulation algorithm. The volume mesh generator fills the holes by constructing new elements according to the required distribution of mesh parameters provided by the local background mesh.



(a) Initial mesh (b) Re-mesh local edges, surfaces

Fig. 1. Example of explosion in urban area. The initial mesh (a) shows mesh clustering at the charge area, while the adapted mesh (b) follows the wave propagation.

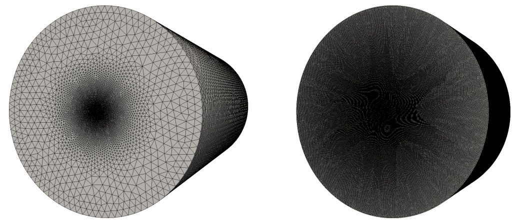
When a new point is generated, a searching process [1] is carried out to identify the element in the old mesh that contains the new point. The solutions are then linearly interpolated to the new point from the nodal values of that element. Although higher order interpolation which guarantee conservation can be used, linear interpolation is commonly employed. It was shown that when local remeshing is used, the conservation properties was not effected by the use of linear interpolation [4].

3 Numerical Examples

3.1 Explosion in cylinder

In this section, simulations of an explosion in a cylinder is carried out with a charge is placed at the centre of one end. Two set of meshes were used including a coarse mesh of 780K elements (with mesh size of 0.03 m at the charge location, 0.35 m for the rest) as seen in Fig. 2(a) and a reasonably uniform finer mesh of 16 millions elements with 0.03 m as mesh size (Fig. 2(b)). First, simulations without mesh adaptation are performed. Flows were simulated using an explicit second order solver with compact Harten-Lax-van Leer (HLLC2) flux scheme [5]. The numerical results for different meshes are shown in Fig. 4. In this figure, the numerical results of pressure probed at the corner of the cylinder are compared with the available empirical data [3]. The peak pressure from empirical data is 4.1×10^6 Pa. It can be observed that the finer mesh apparently provides better solution on both pressure profile and peak pressure. The peak pressure obtained from coarse mesh and fine mesh are 1.45×10^6 Pa and 3.6×10^6 Pa, respectively.

By using mesh adaptation, we start with the coarser mesh (780K elements). The adaptation parameters have been set to: $\epsilon_d = 0.3$, $\delta_{min} = 0.03m$, $\delta_{max} = 0.35m$. Fig. 3 shows the resulting meshes together with the pressure solutions at various times in the simulation. It can be seen that the mesh adaptation can capture the local high-gradient features in the solution and



(a) Coarse mesh (b) Fine mesh

Fig. 2. Initial meshes used for explosion in cylinder: (a) a coarse mesh of 780K elements and (b) fine mesh of 16 millions tetrahedral elements.

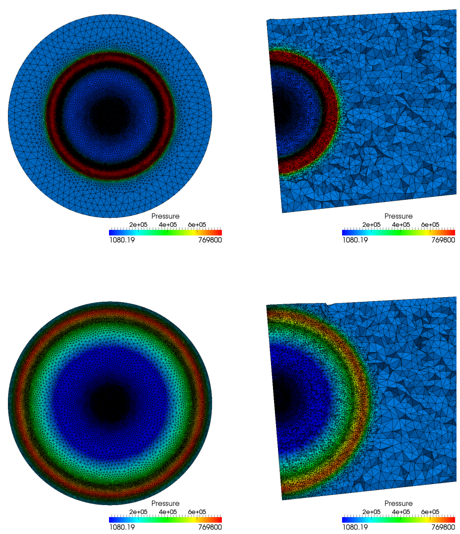


Fig. 3. Local mesh adaptation at different simulation times.

regenerate the mesh locally. The numerical results of mesh adaptation is also presented in Fig. 4. By using mesh adaptation, the peak pressure in mesh adaptation starting from a coarse mesh increase significantly to 4.1×10^6 Pa. In addition, with mesh adaptation, not only does the value of the peak get higher and closer to the empirical, but the arrival time, the time pressure pulse reaches its maximum also shifts toward to the empirical peak. In this simulation, the adaptation period is $1 \times 10^{-4}(s)$. To make the two peaks converge, more adaptation is needed as well as using smaller δ_{min}. The results show good agreement achieved using the mesh adaptation procedure. In addition, the efficiency of the method can be illustrated by comparing the CPU runtime for 10 milliseconds. The time used to run the fine mesh is 97.5 hours with a

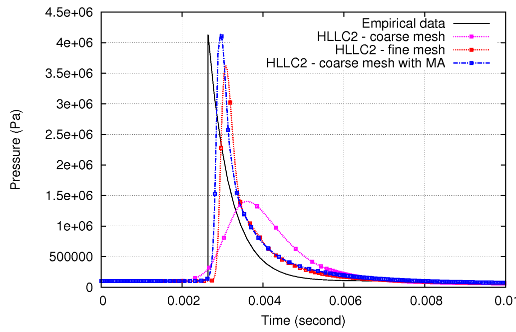


Fig. 4. Pressure probed at the corner of the cylinder on the coarse and fine mesh. Local mesh adaptation (MA) significantly improve the solution in peak pressure and pressure profile.

single CPU (Intel Xeon 2.67 GHz). Using mesh adaptation, it only took 39 hours resulting in about 60% reduction in CPU time.

4 Conclusions

This work has presented a mesh adaptation approach to track the critical features in unsteady compressible Navier-Stokes flows. The approach shows a robust and efficient algorithm. Since, mesh adaptation is only performed at small local areas covering the critical solutions features, the time to do the re-meshing becomes reasonably small compared to the numerical simulation time.

References

1. J. Bonet and J. Peraire, An alternating digital tree (ADT) algorithm for geometric searching and intersection problems, *Int. J. Num. Meth. Eng.*, 31, 1, 1990.
2. P. J. Frey and F. Alauzet, Anisotropic mesh adaptation for transient flows simulations, *Proceedings, 12th International Meshing Roundtable, Sandia National Laboratories*, 335-348, 2003.
3. C. N. Kingery and G. Bulmash, Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Bursts, *ARBRL-TR-02555*, April 1984.
4. K. Morgan, E.J. Probert, O. Hassan and J. Peraire, An adaptive finite element method for transient compressible flows, *Int. J. Num. Meth. Eng.*, 32:11451159, 1991
5. V-T Nguyen, H-H Nguyen, Matthew Price, Jiak Kwang Tan, Strong Shock Capturing Strategies for Compressible Flows on Unstructured Grids in Three Dimensions, *submitted*, 2012.