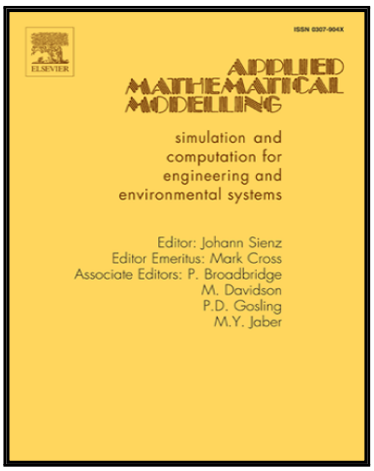


A Feature-Based Mesh Adaptation for the Unsteady High Speed Compressible Flows in Complex Three-Dimensional Domainss

Hoang-Huy Nguyen, Vinh-Tan Nguyen, Matthew Price, Oubay Hassan

PII: S0307-904X(15)00486-2
DOI: 10.1016/j.apm.2015.08.006
Reference: APM 10684

To appear in: *Applied Mathematical Modelling*

Received date: 7 December 2012
Revised date: 24 July 2015
Accepted date: 26 August 2015

Please cite this article as: Hoang-Huy Nguyen, Vinh-Tan Nguyen, Matthew Price, Oubay Hassan, A Feature-Based Mesh Adaptation for the Unsteady High Speed Compressible Flows in Complex Three-Dimensional Domainss, *Applied Mathematical Modelling* (2015), doi: 10.1016/j.apm.2015.08.006

Highlight

- We proposed a featured based mesh adaptation for simulations of compressible ows.
- Adaptivity is done locally by identifying holes and remesh those local regions.
- Local adaptivity is robust and efficient for complex domains with curved boundaries.
- Mesh adaptation improves solution resolutions, especially for strong shocks.
- The adaptation process takes small percentage of run times for unsteady simulations.

A Feature-Based Mesh Adaptation for the Unsteady High Speed Compressible Flows in Complex Three-Dimensional Domains

Hoang-Huy Nguyen[a], Vinh-Tan Nguyen[a,*], Matthew Price[a], Oubay Hassan[b]

[a]*Institute of High Performance Computing, 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632*

[b]*College of Engineering, University of Wales, Swansea SA2 8PP, UK*

Abstract

We propose an unstructured mesh adaptation approach for unsteady high speed compressible Navier-Stokes applications involving blasts and explosions with the presence of strong shock waves propagating in three dimensional complex domains. The idea is to identify the locations of critical physics locally and then re-mesh these regions based on solution derived metrics. The approach ensures both geometry fidelity and mesh validity, especially for areas near complex geometries, a task that is always a challenge in mesh adaptation. The proposed adaptivity is applied for simulations of blast wave propagations and compared with available data in literature. The results show that the proposed method is fully robust and efficient for computational fluid dynamics (CFD) problems in complex three-dimensional domains.

Keywords: mesh adaptation, feature-based, re-meshing, unstructured grids, boundary-conforming, unsteady flows.

*Corresponding author. Address: 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632. Tel: (65) 6419 1591, Fax: (65) 6467 4350.

Email address: nguyenvt@ihpc.a-star.edu.sg (Vinh-Tan Nguyen)

1. Introduction

Modern CFD has the ability to explore problems that are more complex than ever before, partly because of more powerful computing resources. However, the recent evidence suggests that there is still significant unreliability in the numerical predictions made by current CFD codes for the same problem. A strong relation between solution quality and mesh topology has been shown, further indicating that current mesh design practices are not sufficient [6]. Due to the fundamental impact of mesh on the approximation of functions and PDE solutions, mesh adaptation was the focus of many researchers during the last two decades. There are four general approaches of mesh refinement methods. The first approach is p-adaptation, where the interpolation order is locally modified and does not require a new mesh to be generated.While p-adaptation can achieve excellent error convergence for smooth flows, difficulties arise near singularities or discontinuities. This contrasts with the other popular adaptation method, h-adaptation, where the local element size is modified from the current mesh. When combined with unstructured and anisotropic mesh generation capabilities, h-adaptation can improve mesh efficiency in boundary layers, wakes, shocks, etc. However, the disadvantage of h-adaptation is that it could experience large jumps in mesh size and require mesh regeneration. This potentially reduces the effectiveness and robustness of the approach. A related method, r-adaptation, is a simpler variation of h-adaptation. Instead of generating a new mesh, r-adaptation moves node locations without changing the mesh topology to improve the solution accuracy. The final approach is hp-adaptation, where

adjustments in mesh size and interpolation order are combined. In this setting, h-adaptation is employed for non-smooth flow regions in the vicinity of singularities, and p-adaptation is used in smooth flow regions. Sometimes the choice of adaptation strategy in a particular element (h and/or p) is unclear and criteria must be developed to aid that decision. It should be noted that for mesh refinement methods, it is difficult to guarantee the curvature of complex geometries. Refining and coarsening of regions near the domain boundaries usually generate problems [6].

In an effort to improve the robustness and automation of mesh adaptation, this paper proposes an adaptive *re-meshing* method that can be used for variety of problems in CFD and removes the meshing bottleneck which is common to boundary-conforming methods. It should be noted that this method is feature driven adaptivity and is different to mesh adaptation in transient flows involving moving boundaries as were proposed by Löhner [10]; Peraire et al. [15]; Löhner and Baum [11]; Morgan et al. [13]; Hassan et al. [8], etc. The present mesh adaptation is fundamentally different from goal-oriented approach used for unsteady flow simulations [2] in which optimal meshes are generated from a given output functional. Here an isotropic mesh adaptation based on solution features is proposed for unsteady simulations. Adopting isotropic mesh adaptation certainly makes the approach more robust than anisotropic ones in which boundary recovery phase may fail in the process of adaptation. Another difference of the proposed approach is that it does not use strategies such as edge split, edge collapse, edge swap, face swap, point move, etc as were employed in mesh modifications. Instead, it defines the holes locally based on an feature-based indicator and re-meshes these re-

gions independently using surface and volume mesh generators. The surface meshes are performed using advancing front algorithm and volume meshes are based on Delaunay triangulation. The proposed re-meshing process is fully automatic without users intervention in run-time manner provided that initial surface and volume mesh can be generated from input geometries. In addition, it can perform mesh adaptation robustly and effectively in simulation run-time 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, resulting in more accurate solutions.

2. Problem Statement

Considering unsteady inviscid compressible flows governed by the time-dependent, Euler equations on a three–dimensional Cartesian domain $\Omega \subset \mathbb{R}^3$, with surface $\partial\Omega$, it can be expressed in integral form as

$$\int_{\Omega} \frac{\partial \boldsymbol{U}}{\partial t} \mathrm{d}\boldsymbol{x} + \int_{\partial\Omega} \boldsymbol{F}_j n_j \mathrm{d}\boldsymbol{x} = \boldsymbol{0}, \tag{1}$$

where the conventional summation is employed and n_j is the outward unit normal vector to $\partial\Omega$. The unknown vector of the conservative variables, inviscid and viscous flux tensors are given by

$$\boldsymbol{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ \rho \epsilon \end{pmatrix}, \quad \boldsymbol{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_1 u_j + p\delta_{1j} \\ \rho u_2 u_j + p\delta_{2j} \\ \rho u_3 u_j + p\delta_{3j} \\ u_j \left(\rho\epsilon + p\right) \end{pmatrix}. \tag{2}$$

Here ρ denotes the fluid density, u_i the i'th component of the velocity vector and ϵ the specific total energy. Fluid is considered as perfect gas with ideal equation of state $p = \rho RT$ and $\epsilon = c_v T + \frac{1}{2} u_k u_k$ where R is the real gas constant and $c_v = c_p - R$ is the specific heat at constant volume. In this expression, c_p is the specific heat at constant pressure. In this work, the ratio of the specific heats, $\gamma = \frac{c_p}{c_v}$ is set to $\gamma = 1.4$ for air at standard conditions.

Flow unsteady conditions are solved by first discretization of the domain into a computational unstructured grid as a set of non overlapping tetrahedral elements. The governing equations are then solved on the discrete domain using second order cell based vertex centered finite volume approach with explicit time stepping scheme. For better capturing of flow features, solution based adaptivity is developed and employed to adjust computational grids. In subsequent sections, these techniques will be discussed in details.

3. Unstructured Mesh Generation

The computational domain Ω is subdivided into a set of non-overlapping tetrahedral elements using a unstructured mesh generation process. In this section, the methods for generating an unstructured grid are briefly summarized. In an unstructured mesh, the number of points and elements which are neighbours to an interior point is not kept constant throughout the domain. The mesh algorithm can handle arbitrary geometries in a fully automatic manner and provide control over the spatial mesh spacing throughout the domain. Therefore, the input data can be reduced to a geometric representation of the domain based on computer-aided design (CAD) defined geometries. The geometrical definition (or domain boundaries) contains curve and

surface components. The curve components are the curvature continuous composite cubic splines. Surface components are represented by means of a rectangular network of points.

3.1. Background mesh and source distribution

Control over the mesh characteristics is obtained by the specification of a spatial distribution of mesh parameters. A background mesh as well as point, line and planar sources can be used to define the control function that specifies the distribution of the mesh spacings [17].

The background mesh employed must cover the region to be discretized. In the generation of an initial mesh, the background mesh usually consist of a small number of elements. For instance, a background mesh consisting of a single element can be used to impose a linearly varying or a constant spacing through the computational domain.

For complex geometries, the use of a distribution of sources ensures the desired mesh size at specific regions in the computational domain. In this approach, an isotropic spatial distribution of element size is specified as a function of the distance from the point of interest to a 'source'. The source may take the form of a point, a line or a triangle. The form adopted for the function is

$$\delta(x) = \begin{cases} \delta_1 & \text{if } x \leq x_c \\ \delta_1 e^{\left|\frac{x-x_c}{D-x_c}\right| \log 2} & \text{if } x \geq x_c \end{cases} \tag{3}$$

The quantities δ_1, D, and x_c denote user-specified values which can be customized to control the form of $\delta(x)$. The final spacing at a point in the domain is computed as the minimum of the background mesh and sources.

3.2. Surface triangulation

The surface mesh generator utilises the concept of a generation front [12]; [16]. At the start of the process the initial front consists of the sequence of straight line segments which connect consecutive boundary nodes. A side is selected from the front and a triangular element is generated. The front is a dynamic data structure which changes continuously. During the generation process, any straight line segment which is available to form an element side is termed active, whereas any segment which is no longer active is removed from the front. After a triangle has been generated, the front is updated and the generation proceeds until the front is empty.

3.3. Delaunay mesh generator

In this work, we employed the efficient Delaunay triangulation algorithm developed by Weatherill and Hassan [21] which is based on the in-circle criterion [4]. This method, given a set of points and connectivity information for the boundary points, performs a triangulation of the points, automatically creates points in the interior of the domain and ensures that the bounding surface of the domain is contained in the triangulation.

4. Unstructured Grid Compressible Flow Solver

In this paper, we employed an explicit flow solver for compressible flows on unstructured grids [14] to discretize the governing equations (Eq. 1) on unstructured grids. The unstructured grid flow solver is constructed from an edge-based cell-centre finite volume approach with compact Harten-Lax-van Leer (HLLC) flux scheme. The HLLC scheme is appended by second order reconstruction of Riemann states thus obtaining second order accuracy.

As the flows develop strong shock waves, various slope limiters were implemented to stabilize solutions and overcome instabilities due to the high order approximation. Summary of the discretization scheme is summarized in the following sections.

4.1. Edge-based Vertex-centre Finite Volume Spatial Discretization

The discretisation of Eq. (1) is accomplished using a cell vertex finite volume procedure. The computational domain Ω is subdivided into a set of non-overlapping tetrahedral elements using a Delaunay mesh generation process with automatic point creation described in the previous section. To enable the implementation of a cell vertex finite volume solution approach, a median dual mesh is constructed by connecting edge midpoints, element centroids and face centroids such that only one node is present in each control volume. The edge coefficients for internal and boundary edges are calculated for every edge using the dual mesh segment associated with the edge connecting point I and J as follows,

$$C_j^{IJ} \equiv n_j^{IJ} = \sum_{K \in \Gamma_{IJ}} A_{\Gamma_I^K} n_j^{\Gamma_I^K} \tag{4}$$

$$D_j^{IJ} = \sum_{K \in \Gamma_{IJ}^B} A_{\Gamma_I^K} n_j^{\Gamma_I^K}. \tag{5}$$

In the expression (4), $A_{\Gamma_I^K}$ is the area of facet Γ_I^K and $n_j^{\Gamma_I^K}$ is the outward unit normal vector of the facet from the viewpoint of node I. And Γ_{IJ}^B is the set of dual mesh facets on the computational boundary touching the edge between nodes I and J.

The contribution of the inviscid flux over the control volume surface for

node I is then computed as

$$\int_{\partial\Omega_I} F_{ij} n_j \mathrm{d}\boldsymbol{x} \approx \sum_{J\in\Lambda_I} \frac{C_j^{IJ}}{2} \mathcal{F}^{IJ} + \sum_{J\in\Lambda_I^B} D_j^{IJ} \mathcal{F}^I \tag{6}$$

where Λ_I denotes the set of nodes connected to node I by an edge and Λ_I^B denotes the set of nodes connected to node I by an edge on the computational boundary. $\mathcal{F}^{IJ}$ is the inviscid numerical flux in IJ direction obtained from solving local Riemann problems at the facets of node I's control volume. There are a number of ways for computation of numerical fluxes, in this work, for high speed compressible flow the compact Harten-Lax-van Leer (HLLC) flux scheme [14] is employed. The HLLC scheme is appended by second order reconstruction of Riemann states thus obtaining second order accuracy. As the flows develop strong shock waves, various slope limiters were implemented to stabilize solutions and overcome instabilities due to the high order approximation.

4.2. Time Discretization

When the terms in Eq. (1) is approximated by the above finite volume discretization scheme, the final form of discrete equations is written as

$$\frac{\mathrm{d}\boldsymbol{U}_h(t)}{\mathrm{d}t} = \boldsymbol{R}_h(\boldsymbol{U}_h, t), \tag{7}$$

where $\boldsymbol{U}_h$ is the discrete solution vector and $\boldsymbol{R}_h$ is the residual vector representing FV discretization of the inviscid fluxes,

$$\boldsymbol{R}_h(\boldsymbol{U}_h, t) = \sum_{J\in\Lambda_I} C^{IJ} \mathcal{F}^{IJ} + \sum_{J\in\Gamma_I^B} D^{IJ} \mathcal{F}^I \tag{8}$$

The time derivative term is discretized using finite difference at time step n and the residual term is evaluated from solutions at time step n and

possibly previous time steps. In this work, the following explicit K^{th} order Runge-Kutta time discretization can be used for solving Eq. (7)

$$\boldsymbol{W}_h^{n(i)} = \sum_{l=0}^{i-1} \alpha_{il} \boldsymbol{W}_h^{n(l)} + \beta_{il} \Delta t^n \boldsymbol{R}_h(\boldsymbol{U}_h^{n(l)}), \qquad i = 1, ..., K \tag{9}$$

where $\boldsymbol{W}_h^{n(0)} = \boldsymbol{W}_h^n$ and $\boldsymbol{W}_h^{n+1} = \boldsymbol{W}_h^{n(K)}$. The coefficients are required to satisfy some conditions such that the scheme is satisfied to be total variation diminishing (TVD). A class of TVD Runge-Kutta schemes is presented by [19] and proven to be suitable for solving the hyperbolic conservation laws with stable spatial discretization.

5. Local Mesh Adaptation

In solving (unsteady) problems involving high-speed compressible flows, the location of critical physics is often not known ahead of time. These localized regions will generally move through the computational domain and may sweep across large areas. The mesh will thus need to be adapted to allow for accurate tracking of high-gradient features in the solution. However, it is not recommended to re-mesh the whole computational domain. This is not only computationally expensive but may also result in reduced accuracy, due to the inherent numerical diffusion that occurs when interpolating data from old grid to new grid. Frey and Alauzet [7] used mesh modification to generate an adapted mesh to capture the transient phenomena. In their approach, the ingredients to achieve this goal typically include mesh enrichment, mesh coarsening and local mesh optimization procedures. The local mesh modifications operators are: edge flipping, edge collapsing, edge splitting and node removal, node repositioning and degree relaxation.

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 using surface and volume mesh generators. The localized regions are defined as small as possible to cover the area of fast changing solution but still ensure the smoothness between the adapted and the remaining regions. At the same time, it is also important to preserve the global geometrical definition of the domain boundaries, especially with curved surfaces. The mesh will be refined or coarsened dependent on the fast changing or uniform flow features detected.

5.1. Feature-Based Adaptivity

In pioneering work on error-based mesh adaptivity [10, 17], the mesh size spatial distribution is computed from second derivatives of solutions under the principle of equidistribiution of errors. Those Hessian-based approaches aim at controlling errors across the whole domain ignoring time discretization error in unsteady simulations. In this work, a new mesh size distribution function is defined based on the solution gradient and compared with the current mesh size thus creating local regions for remeshing. The proposed mesh size distribution also takes into account the mesh transition in time. The numerical solution u such as density, pressure, etc at the current time will be used to predict the desired element size for the new mesh.

5.1.1. Solution-based size function

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

$$\delta_P = \delta_{min} + (\delta_{max} - \delta_{min}) e^{-\frac{(|\nabla u|_P - |\nabla u|_{min})^2}{2c^2}} \tag{10}$$

The new mesh size of the local regions to be re-meshed will fall into the range of $[\delta_{min}, \delta_{max}]$. 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). At each node, the minimum value between the new mesh size and the global mesh size defined by the background mesh and sources is used for mesh adaptation.

5.1.2. Re-meshing criteria

As the optimal nodal spacing is determined from given solutions, the mesh will be adapted in the areas where there is a significant difference between the desired spacing and current mesh spacing. An indicator is defined to identify the regions where the mesh needs to be changed. From the new mesh spacing calculated at each node of the current mesh based on the above size function (Eq. 10), the percentage change between the current mesh spacing and the new one is defined at each node as

$$\epsilon_P = \frac{|\delta_P - \bar{\delta}_P|}{\bar{\delta}_P} \tag{11}$$

where δ_P and $\bar{\delta}_P$ are the new and current mesh spacing at node P, respectively. The computed mesh spacing can be smaller or larger than the current mesh spacing; thus requiring either refining or coarsening of the local regions. Nodes in regions where percentage change is greater than a certain threshold ($\epsilon_P \geq \epsilon_d$) will be deleted. Those deleted nodes and its surrounding neighbours collectively form local holes to be re-meshed. The current mesh is then modified with the objective of meeting the new distribution of mesh characteristics as closely as possible.

Fig. 1 shows an example of mesh adaptation procedure for a problem of explosion in a box. A charge is initialized as a region of high pressure inside the box. New mesh spacing can be derived from solution gradient Fig. 1(a) using the size function Eq. (10). The new and current mesh spacing are displayed in Fig. 1(b) and 1(c). This resulted in the percentage change between the spacings that was used to determine the local holes. It can be seen that the local area to be deleted (red regions in Fig. 1(d)) is well within the area of high gradient solutions.

5.2. Re-meshing algorithm

5.2.1. Hole identification and deletion

This process starts from the current unstructured mesh. Nodes which are identified by the re-meshing criteria are marked for deletion. The surrounding nodes that connected to marked nodes are also marked for deletion to create a smooth transient growth in mesh spacing between the re-meshed regions

and the remaining parts of the domain. Finding these surrounding nodes can be done in several rounds to ensure smooth transition of mesh distribution. From our experience, 2 to 3 rounds of identifying deleted nodes results in a good transition of the mesh spacing. Provided the list of deleted nodes, all elements connected to those nodes also need to be marked for deletion. The marked elements are then removed from the original mesh and stored to act as reference background grids for later interpolation of solutions from old mesh to new adapted mesh. The deleted elements are grouped into 'holes'. As seen in Fig. 1(d), the red areas define the holes with $\epsilon_d = 30\%$ as the percentage change threshold. 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.

5.2.2. Local curve and surface re-meshing

Each hole is bordered by a collection of faces in three dimensions which defined the outer hull of the hole to be re-meshed. In some cases, the faces are not located on the domain boundaries Fig. 2(a) and only require a local volume adaptation for these holes. On the other hand, there are holes with faces residing on the geometrical surface components Fig. 2(b). The collection of these faces will define local geometrical surfaces which are needed to re-meshed as part of the adaptation process. New generated faces of the local surfaces and the remaining faces of the hole will be used as initial fronts for the local volume re-meshing.

The local regions to be triangulated on surface components are defined by the closed loops of oriented curves Fig. 3. In addition, these loops may contain local curves which are on the global curve components and also need

local re-meshing. Due to the randomness in forming the holes, the configuration of local surfaces can be complex. Therefore, special treatments are needed to enhance the local boundary definition of the surfaces to be discretized so that these local areas are more manageable by the surface mesh generator.

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 definition to ensure that the new generated nodes are placed on the true geometrical curve and surface definition. Fig. 4(a) and Fig. 4(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. 4(a). At the beginning, a fine mesh was used at the source of explosion. As the explosion sweeps through the nearby building structures, the mesh will be adapted to the change in solutions. The adapted mesh at a certain time of the simulation is shown in Fig. 4(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.

5.2.3. Local volume re-meshing

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. A local background mesh is constructed based on the deleted tetrahedral elements of each hole. The volume mesh generator, using the above Delaunay triangulation scheme 3.3 fills the holes by constructing new elements accord-

ing to the required distribution of mesh parameters provided by the local background mesh.

5.2.4. Solution interpolation

There exist various techniques for solution interpolation between different meshes from a classical linear interpolation to conservative interpolation using cut-cell approach [1] or supermesh [5]. The conservative interpolation ensures mass conservation but introduces extra computational cost to the procedure. Moreover it was shown in [1] that benefits of conservative approach is small compared to cost for simulations of blasts and explosions. In this work the classical linear interpolation approach was adopted. When a new point P from the adapted mesh is introduced to the previous background mesh, a searching process is carried out to identify the element κ_i in the old mesh that contains the new point. The alternative digital tree [3] is used to accelerate this searching process. The solution $u(P)$ is then linearly interpolated to the new point P from the nodal values of that element.

$$u(P) = \sum_{j=1}^{n_k} b_j(P) u(P_j), \tag{12}$$

where n_k is the number of vertices P_j of element κ_i and b_j is the barycentric coordinate of P with respect to the element κ_i.

5.2.5. Local mesh adaptation procedure

In summary, the local mesh adaptation process will include the following steps:

1. Derive solution-based mesh size

- New nodal spacing is derived from current solution with respect to a 'key' variable (pressure, density, velocity, etc) (Eq. 10).
- Compare with the global spacing to get the final desired mesh size.

2. Identify holes
   - Compute percentage change between new and current nodal spacings (Eq. 11).
   - Use re-meshing criteria to mark regions where the mesh should be changed.
   - Group deleted elements into holes and build a list of remaining nodes.
3. For each hole, do re-meshing locally
   - Identify the boundary faces of the hole.
   - From the boundary faces, if there are no local edges located as part of the global geometrical definition curves, go to step 3b to do local surface triangulation.

   (a) Re-mesh curves on domain boundaries
   - Construct local line sources by using the original curve segments with new spacings.
   - Discretize the local curves.
   - Update the list of nodes and proceed to surface re-meshing.

   (b) Local surface re-meshing
   - The local surface components of the hole boundaries are extracted from the list of deleted elements.

- If there are no faces located on the domain boundaries, go to step 3c to do the volume adaptation.
- Define the surface regions to be re-meshed based on the closed loops of oriented curves. Each local surface may contain several regions.
- Start advancing front triangulation for each region and perform mesh enhancements.
- Update the list of nodes and export the new surface mesh for volume generation.

(c) Generate local volume mesh for each hole

- Construct a local background mesh by using the deleted elements of the hole with the new spacings.
- Using the new generated faces as boundaries (if any), the remainder of the hole is re-meshed using the isotropic Delaunay triangulation with the local background mesh defined above.
- Add the local new meshes to the global mesh.

4. Interpolate values of the unknowns for all the newly generated nodes.

5.2.6. Efficiency

For mesh adaptation, re-meshing the whole computational domain is not favourable. It is necessary to obtain better accuracy but at a lower computational expense. In this approach, adaptation will be performed only at regions surrounding the critical physics areas. The re-meshed regions should be as small as possible but still ensure the smooth transition in mesh size between the remaining region and the holes. The parameter c in Eq. (10) con-

trolling the width of the Gaussian distribution function will define the local regions covering the high gradient features where fine meshes are needed. In addition, imposing the current spacing at the boundary of the holes instead of using the new spacing will help to create the smooth transition between the holes and the remaining parts.

6. Numerical Examples

6.1. Explosion in cylinder

In this section, simulations of an explosion in a cylinder were carried out to demonstrate the effectiveness of the proposed local mesh adaptation scheme. The problem involves simulations of the flow field when a charge is placed at the centre of one end of the circular cylinder of radius of 5 meters and 20 meters in length. The equivalence of 100kg TNT charge was initialized using a spherical burst model. As the charge is activated, a high-pressure field will move toward the cylinder walls generating reflected waves and interacting with each other to propagate downstream to the other end of the cylinder. The air inside the cylinder is modelled as compressible and initially set at atmospheric condition ($101325Pa$). Here, we employed the explicit flow solver as described in section 4 coupled with the proposed mesh adaptation to simulate the explosion in the cylinder. As the flows develop strong shock waves, the local extremum diminishing (LED) limiter was used to stabilize solutions and overcome instabilities due to the high order approximation. Using the proposed approach, it is possible to observe the solution of transient flows using adaptive re-mesh methods.

First, simulations without mesh adaptation were performed to set as

benchmarks for comparisons. The mesh were clustered at the centre of one end where the charge is placed to better representing the explosion. Two set of meshes were generated for simulations including a coarser mesh of 780K elements (with minimum mesh size of 0.03 m at the charge location, 0.35 m for the rest of the cylinder) as seen in Fig. 5(a) and a reasonably finer mesh of 16 million tetrahedral elements with 0.03 m as mesh size Fig. 5(b). Flows were simulated using first order (HLLC1) and second order (HLLC2) scheme. In mesh adaptation process, we started with the coarser mesh (780K elements) clustered at the charge location to capture the explosive initialization. The adaptation parameters have been set to: $\epsilon_d = 0.3$, $\delta_{min} = 0.03m$, $\delta_{max} = 0.35m$. The explosive wave is advanced toward the wall of the cylinder. At a certain adaptation frequency, the mesh is adapted based on the pressure solution to better follow the wave propagation in this case. Fig. 6 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 regenerate the mesh locally. The elements whose size and shape do not meet the requirement of the new solution distribution are identified and the mesh is adapted according to the new solution-derived mesh size. In addition, local mesh adaptation shows advantages in re-meshing efficiency. As can be seen in Fig. 6, adaptation is only performed at one end of the cylinder where large gradients are observed. In these figures, the local holes to be remeshed follow the critical flow as they move from the centre toward the cylinder wall. The mesh are reasonably large at regions with uniform flow and sufficient small at the critical areas.

The results using different simulation strategies including mesh adaptation approach are shown in Figure 7. In this figure, the numerical results of pressure probed at the corner of the cylinder are compared with the available empirical data [9]. The empirical data is derived from a hemispherical TNT surface burst and the pressure data is presented for a case of reflected shock on a flat surface at 5 meters away from the charge. The peak pressure from empirical data is 4.1×10^6 Pa. While the finer mesh apparently provides better resolutions to solution, the second order scheme, HLLC2, improves the solution further. It can be observed from Figure 7(a) that HLLC2 performs better on both pressure profile and peak pressure. The numerical results of mesh adaptation starting from a coarse grid are also shown in Figure 7(a). By using mesh adaptation, HLLC1 scheme can give better results at 2.1×10^6 Pa peak which is the same as the fine mesh result using HLLC1 solver. The peak in mesh adaptation with HLLC2 scheme increase significantly to 4.1×10^6 Pa and almost close to the empirical data. 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. The efficiency of the proposed mesh adaptation approach can be illustrated by comparing the CPU time of various simulations as shown in Figure 7(b) for the same simulation. Run times of coarse mesh (780K elements) with adaptation are compared with run times used to run fine mesh (16M elements) without mesh adaptation. The times used to run the fine mesh with HLLC1 and HLLC2 solvers are 18 hours and 97.5 hours, respectively, with a single CPU (Intel Xeon 2.67 GHz). By employing mesh adaptation, run times with HLLC1 and HLLC2 solvers are 11

hours and 39 hours, respectively, resulting in about 40% and 60% reduction in CPU time. Therefore, mesh adaptation does help to reduce the time in producing accurate numerical results. It also noted that the adaptation time is just a small percentage of the total run time. In case of HLLC1 scheme, the adaptation time is about 30% of the total run time. However, in case of HLLC2 scheme, it can be seen that the solver time is much more than the adaptation time due to the fact that HLLC2 solver is more expensive while the adaptation time remains the same.

For a comparison with experimental data, a similar simulation setup was also carried out for a blast in a cylindrical tube [18]. The test consists of a 24 m long steel tube with a 1.5 m diameter. A spherical charge of 500 g Swedish plastic explosive (Sprangdeg m46) was detonated at 1.0m from the open end. Pressures are measured at gauges 11m and 20m from the charge along the length of the tube. Numerical simulations were carried out to test the accuracy and efficiency of mesh adaptation process. A comparison of the test data and simulation is shown in Figure 8 between experimental data, simulation results on uniform mesh and adaptive mesh based on pressure. Experimental data has been operated on with a low-pass filter. The uniform mesh had an element size of 3cm near the charge location (for initialization) and 5cm elsewhere for a total of 2.12 million tetrahedral elements. The initial mesh for the adaptive runs had an element size of 3cm near the charge and 15cm elsewhere with 0.64 million tetrahedral elements. The minimum and maximum element size controls for adaptivity were 3 cm and 15 cm respectively. The simulations were run on 4 CPUs of a workstation with 32 Gb of RAM. It can be seen that numerical simulations agree well with experimental data

for both shock peak pressure and arrival time. However, the simulations do not predict the negative pressure dip after the shock which is observed in the experiments [18]. Mesh adaptive simulations based on pressure and density provided nearly identical pressure history as shown in Figure 8. As shown in Figure 9(b) density-based adaptivity requires more elements to capture the density gradients at the contact interface behind the shock.. It can be seen that compared to uniform mesh, mesh adaptivity reduces number of elements in the domain while maintaining sufficient mesh resolution in critical flow regions for accurate shock capturing. Figure 9(a) shows a comparison of CPU time for different simulations. It is clearly seen that the mesh adaptive simulations result in large reduction of run-time compared to running on a uniform mesh while rendering the same accuracy. Density based adaptation required longer run time compared to pressure based counter part due to larger mesh generated using density as the refinement feature. For applications of blast and explosion, it is found that pressure based adaptation is the most efficient strategy for both accuracy and efficiency.

6.2. Explosion in urban area

Another example will demonstrate the capability of the proposed mesh adaptation on a large scale area with complicated geometrical features. The problem concerns an explosion of an equivalence of 2000kg TNT charge in an urban area Figure 10 as was described in [20]. The metric is defined based on the pressure gradient. The adaptation parameters have been set to: $\epsilon_d = 0.3$, $\delta_{min} = 0.5$ m, $\delta_{max} = 2.0$ m for a computational domain size of $160\text{m} \times 130\text{m} \times 50\text{m}$. By varying the parameter c in Eq. (10), one can control the refining areas around the regions with critical physics. In this example,

we set $c = 0.04$ to define the width of the Gaussian distribution. Adapted meshes are presented in Figure 11. As shock waves propagate through the domain, they are focused more in the street channels between the buildings; thus requiring mesh adaptation on geometrical surface definition. Figure 12 shows comparison of pressure history measured at two different locations in a similar scaled city set-up as shown in [18]. It can be seen that the transient phenomena are well captured by the proposed mesh adaptation procedure as compared to experimental data. Simulations with mesh adaptation provide similar results with uniform mesh while it takes much less time (about 10 times less) to run the simulations. It highlights the benefits and predictive capability of the local mesh adaptation process in simulating explosion in areas with complex geometries.

7. Conclusions

This paper has presented a mesh adaptation approach to track the critical features in unsteady compressible Navier-Stokes flows. It is capable of generating suitable meshes for problems in which the development of the solution is not known before hand. The approach shows a robust, reliable and efficient algorithm to cater for local mesh adaptation based on solution output at any time. The adaptation procedures combine local curve, surface and volume re-meshing allowing the quality of the solution to be enhanced. In addition, the boundary geometries of the computational domain are always preserved in the adaptation process. 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. The approach provides a powerful tool for mesh adaptation for domains of complex geometry. Future works will focus on the scalable parallelization, local anisotropic mesh adaptation and mesh adaptation in fluid structure interactions.

References

References

[1] Alauzet F, Mehrenberger M, P1-conservative solution interpolation on unstructured triangular meshes, *Int. J. Numer. Meth. Engng.*, Volume 84, Issue 13, 24 December 2010

[2] Belme A, Dervieux A, and Alauzet F, Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows, *J. Comp. Phys.*, Vol. 231, pp. 6323-6348, 2012

[3] 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.

[4] A. Bowyer, Computing Dirichlet tessellations, *The Comput. J.*, **24**:162-166, 1981.

[5] Farrell PE, Piggott MD, Pain CC, Gorman GJ, Wilson CR, Conservative interpolation between unstructured meshes via supermesh construction, *Comp. Meth. Appl. Mech. Eng.*, 198, 2632-2642, 2009

[6] K. J. Fidkowski and D.L. Darmofal, Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics, *AIAA Journal*, Vol. 49, No. 4, April 2011.

[7] P. J. Frey and F. Alauzet, Anisotropic mesh adaptation for transient flows simulations, *Proceedings, 12th International Meshing Roundtable, Sandia National Laboratories*, 335-348, 2003.

[8] O. Hassan, E. J. Probert, K. Morgan and N. P. Weatherill, Unsteady flow simulation using unstructured meshes, *Computer Methods for Applied Mechanical Engineering*, **189**:1247-1275, 2000.

[9] C. N. Kingery and G. Bulmash, Airblast Parameters from TNT Spherical Air Burst and Hemispherical Surface Bursts, *ARBRL-TR-02555*, April 1984.

[10] R. Löhner, An adaptive finite element scheme for transient problems in CFD, *Computer Methods in Applied Mechanics and Engineering*, **61**:323-338, 1987.

[11] R. Löhner and J.D. Baum, Numerical simulations of shock interaction with complex geoometry three dimensional structures using a new adaptive h-refinement scheme on unstructured grids, *AIAA Paper*, 90-0700, 1990.

[12] R. Löhner and P. Parikh, Three dimensional grid generation by the advancing front method, *Int. J. Num. Meth. Fluids*, **8**:1135-1149, 1988.

[13] K. Morgan, E.J. Probert, O. Hassan and J. Peraire, An adaptive finite

element method for transient compressible flows with moving boundaries, *Int. J. Num. Meth. Eng.*, **32**:751-765, 1991.

[14] V-T Nguyen, H-H Nguyen, M.A. Price, J.K. Tan, Shock capturing schemes with local mesh adaptation for high speed compressible flows on three dimensional unstructured grids. *Computers & Fluids* 70, 2012. 126135.

[15] J. Peraire L. Formaggia and K. Morgan, Simulation of store separation by the finite element method, *Applied Mathematical Modelling*, **12**:175-181, 1988.

[16] J. Peraire, J. Peiro, L. Formaggia, K. Morgan, and O. C. Zienkiewicz, Finite element Euler computations in three dimensions, *Int. J. Num. Meth. Eng.*, **26**:2135-2159, 1988.

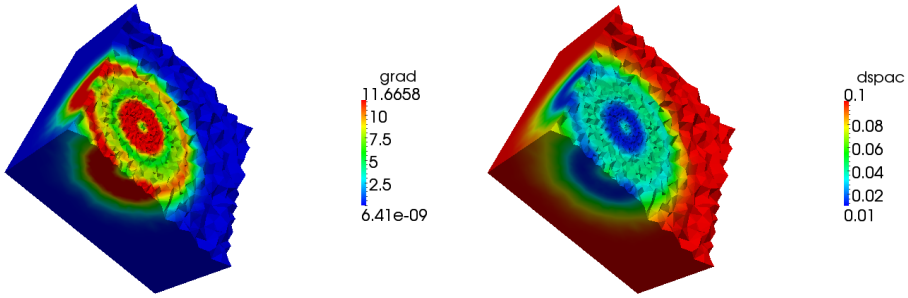
[17] J. Peraire, K. Morgan and J. Peiro, Unstructured finite element mesh generation and adaptive procedures for CFD. Applications of Mesh Generation to Complex 3-D Configurations, *AGARD Conference Proceedings No. 464*, 18.1-18.12,1990.

[18] M.A. Price, V.-T. Nguyen, H.H. Nguyen, J.K. Tan, C.S. Chew, T. Karasek, Computational Framework for Simulation of Air Blast and Structural Interactions. *Proc. 22nd Symposium on Military Aspects of Blast and Shock*. 4-9 Nov., 2012. Bourges, France.

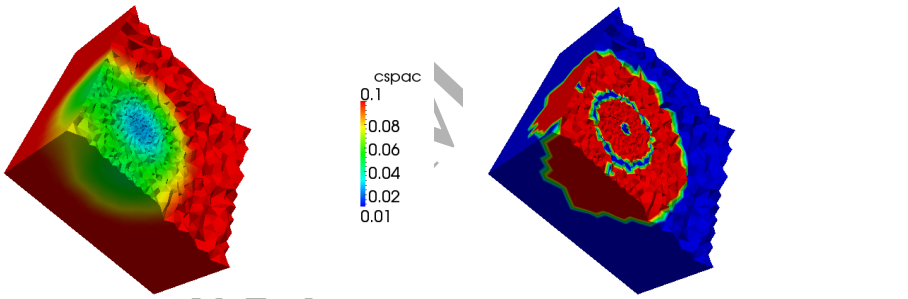
[19] C.W. Shu., Total-variation-diminishing time discretizations, *SIAM. Sci. Stat. Comput.*, Vol. 9(6), 1073-1084, 1988.

[20] P. D. Smith and T. A. Rose, Blast wave propagation in city streets - an overview, *Progress in Structural Engineering and Materials*, Vol. 8, **1**:16-28, 2005.

[21] N. P. Weatherill and O. Hassan, Efficient three dimensional Delaunay triangulation with automatic boundary point creation and imposed boundary constraints, *Int. J. Num. Meth. Eng.*, **37**:2005-2039, 1994.

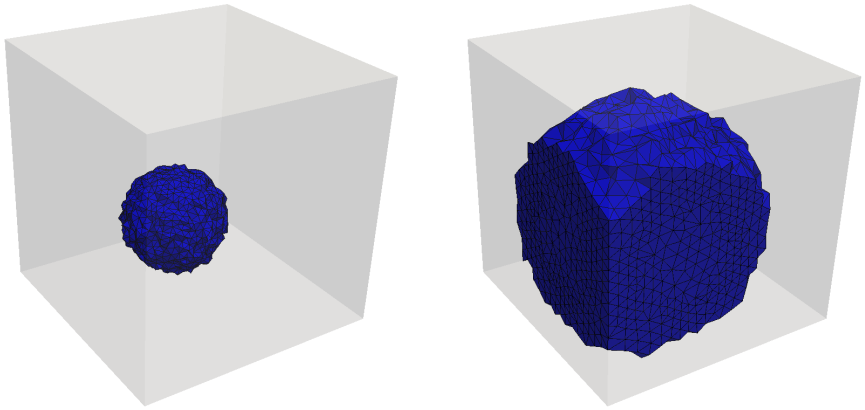


(a) Solution gradient (b) Mesh size derived from solution



(c) Current characteristic mesh size (d) Deleted hole (red area)

Figure 1: Adaptation process for the problem of explosion in box. The new nodal spacing (b) is determined from given solutions (a) and compared with current mesh spacing (c). The difference between new and current mesh spacing is computed to determine element to be deleted. Those deleted elements will then form local deleted regions or holes (d) for re-meshing.



(a) Hole completely inside the domain (b) Hole with faces on boundaries

Figure 2: Local holes to be re-meshed in the cases of hole not intersecting with domain surfaces (a) and hole with faces residing on the geometrical surface component (b)

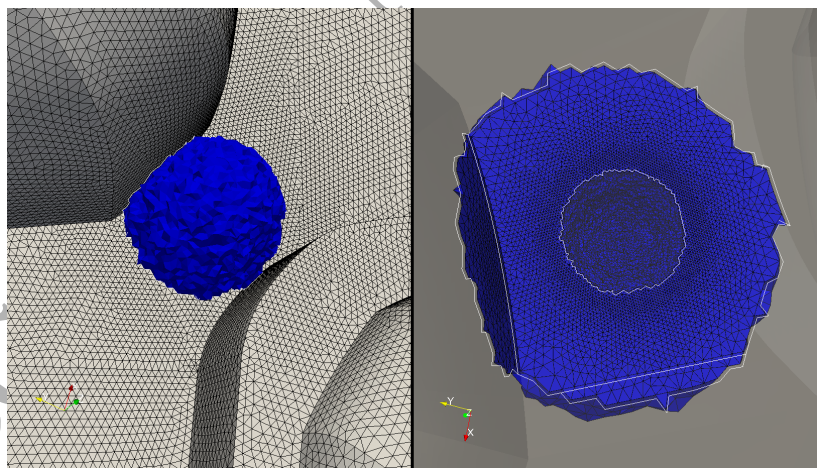
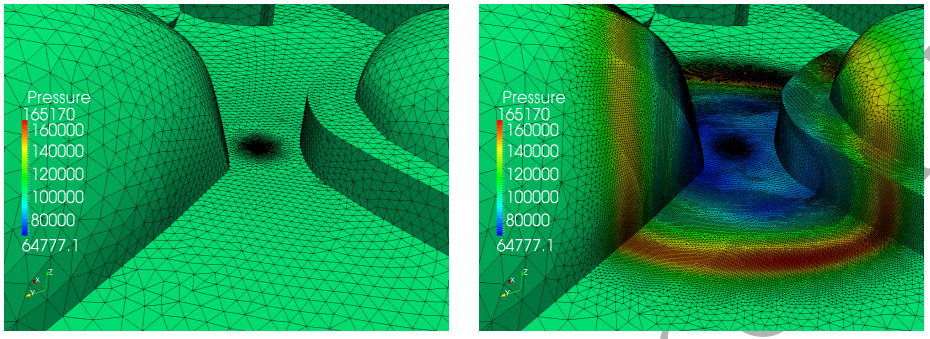
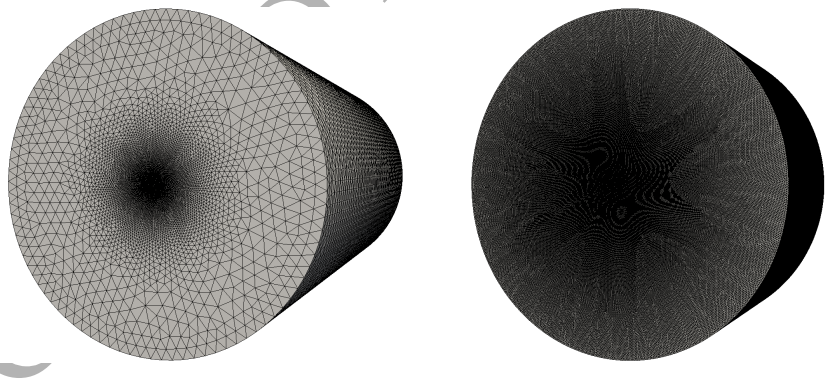


Figure 3: A hole with local curves and surfaces located on boundary components of the domain (explosion in urban area).



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

Figure 4: 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.



(a) Coarse mesh (b) Fine mesh

Figure 5: Initial meshes used for explosion in cylinder: (a) a coarse mesh of 780K elements and (b) fine mesh of 16 millions tetrahedral elements.

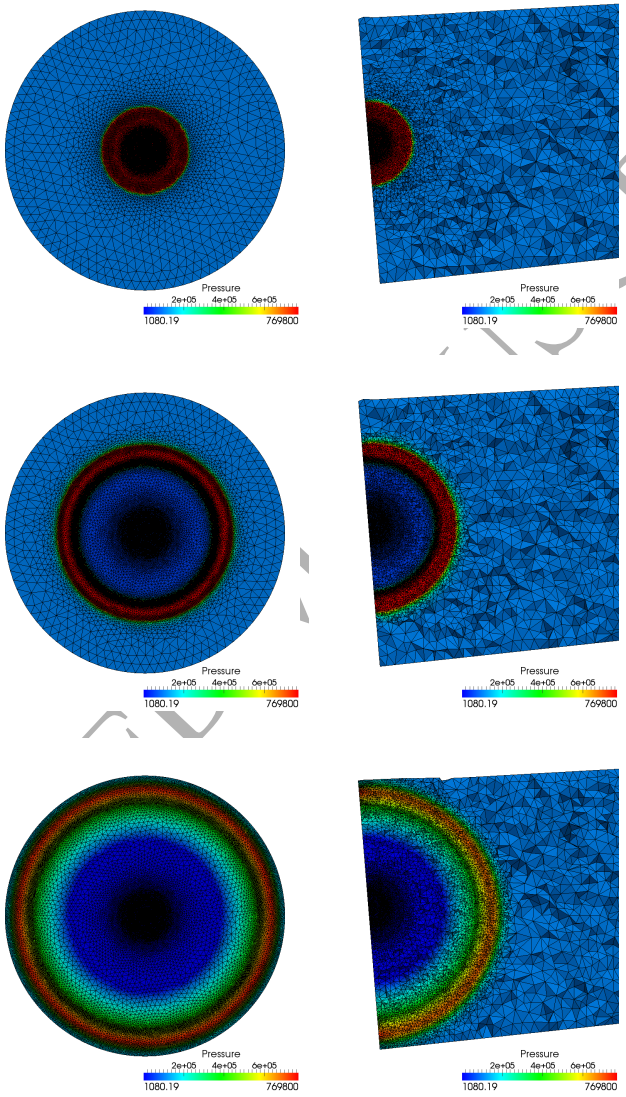
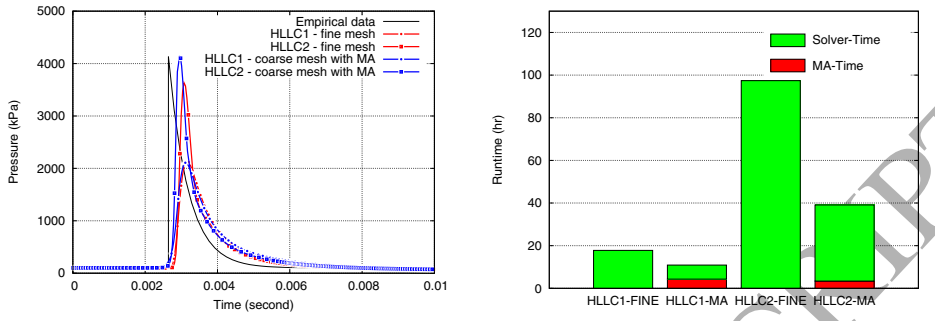
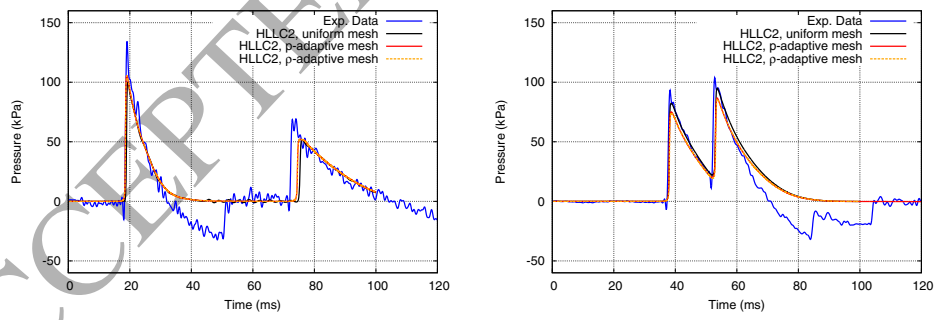


Figure 6: Local mesh adaptation at different simulation times.



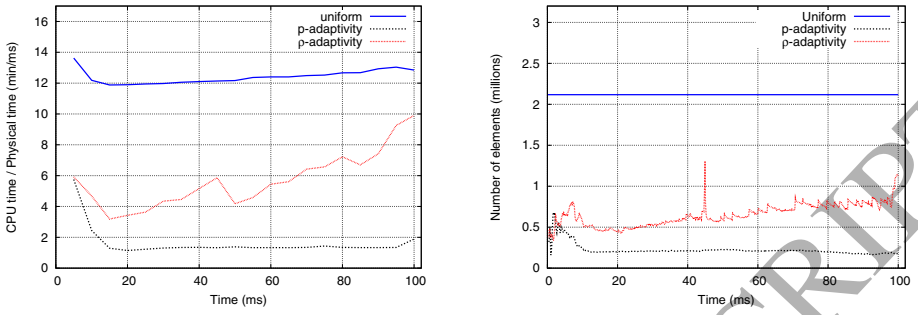
(a) Presure history (b) Running time

Figure 7: (a) Comparison of pressure histoty at the corner of the cylinder between different approaches. It can be observed that second order scheme provides higher peak and better pressure profile than first order scheme. HLLC1 and HLLC2 scheme with local mesh adaptation (MA) significantly improve the solution in peak pressure and pressure profile. (b) Mesh adaptation using first and second order HLLC scheme provide more accurate prediction with much less running time as compared with the same numerical scheme on fine mesh. Mesh adaptation process takes about less than 10% in simulations using HLLC2.



(a) At gauge #1 (b) At gauge #2

Figure 8: Comparison of pressure history at (a) gauge #1 and (b) gauge #2 at 11m and 20m along the length of the tube.



(a) Run time (b) Mesh statistics

Figure 9: Comprison of (a) run time and (b) mesh size between different simulations using uniform mesh, adaptivity based on pressure and density.

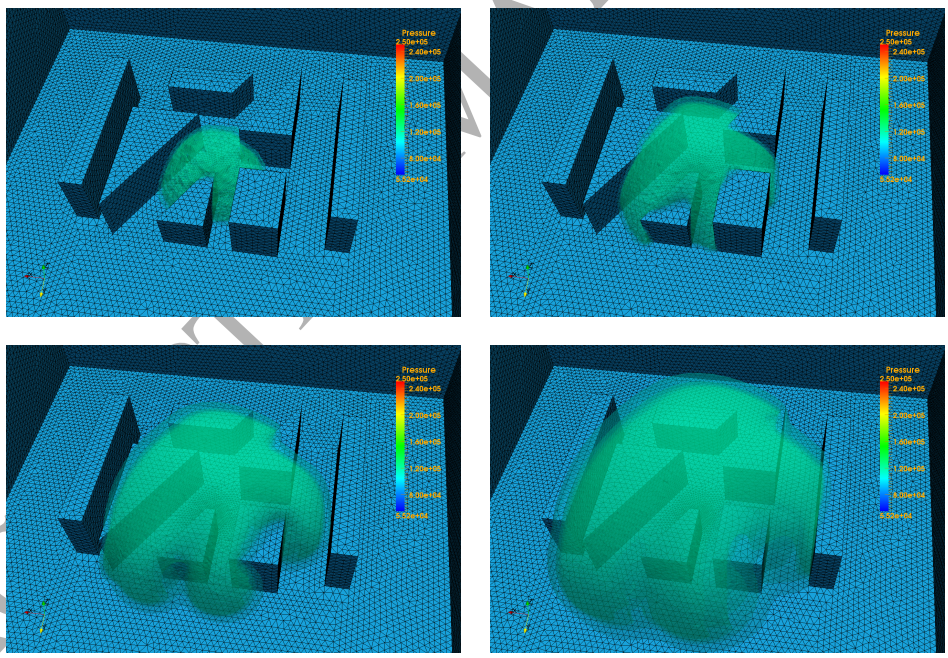


Figure 10: Isopressure surfaces at times t = 8 ms, 21 ms, 40 ms and 60 ms showing the development of the explosion in the city area.

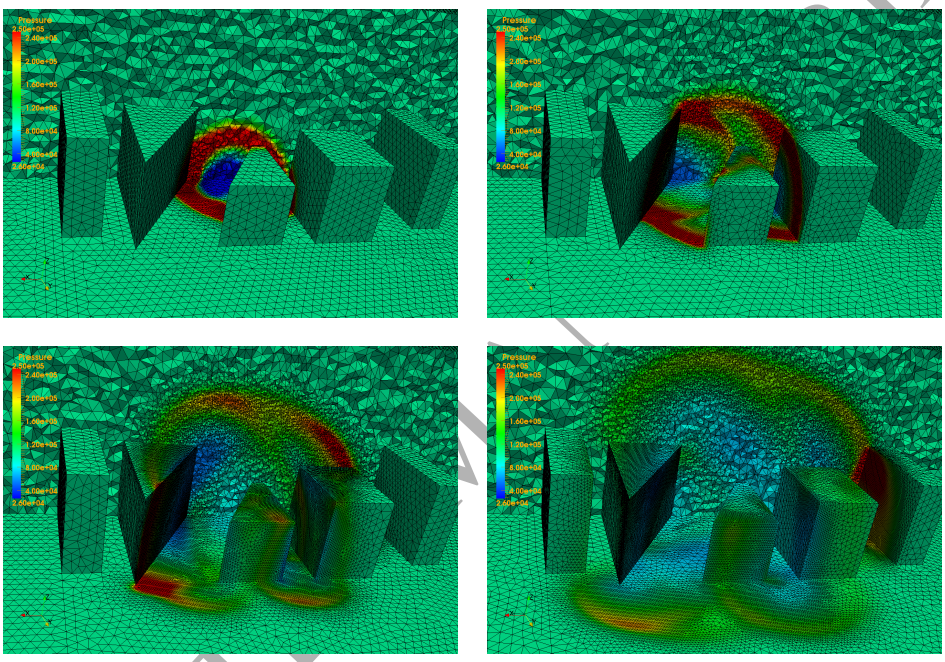
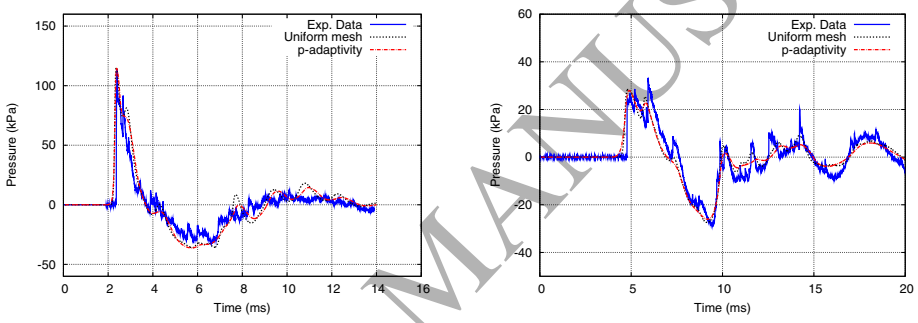


Figure 11: Adapted volume meshes and cutting plane through the domain at times t = 8 ms, 21 ms, 40 ms and 60 ms. The mesh is refined at high gradient pressure and coarsened at regions with small gradient. Refinement is applied to geometrical surfaces where mesh sizes on the building surfaces are adjusted following the wave propagation.



(a) At gauge #1 (b) At gauge #2

Figure 12: Pressure history and comparison between numerical simulations and experiment data for blast in a scaled city [18]. Pressure is probed at two different location along the street channel.