European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2006
P. Wesseling, E. Oñate and J. Périaux (Eds)
© TU Delft, The Netherlands, 2006

TIME DEPENDENT ADAPTATION FOR COMPRESSIBLE FLOWS (ECCOMAS CFD 2006)

Jerzy Majewski

Warsaw University of Technology,
Institute of Aeronautics and Applied Mechanics,
Nowowiejska 24, 00-665 Warsaw, Poland
e-mail: jmajewsk@meil.pw.edu.pl

Key words: Computational Fluid Dynamics, Adaptation, Compressible Flows

Abstract. *The paper deals with the algorithm for unsteady anisotropic adaptation. The adaptation is using remeshing approach to create a new refined grid. The grid is generated in computational space defined by metric field based on the tensor error indicator. The verification of the method is done first for 2D supersonic flow in a channel with forward facing step and subsequently for fully 3D flow past the sphere.*

1 ANISOTROPIC ADAPTATION

The adaptation is a very powerful tool for optimizing the calculations. Unfortunately the typical isotropic adaptation used for 3D flows results in excessive number of elements. The reason for this is that during refinement of the grid around, e.g. the shock wave the grid cells becomes smaller uniformly in all directions. It means that after splitting N cells in direction normal to the shock wave the number of the cells will be increased to $N\,K^3$ (K describes how many new edges are created after splitting an old one - if an edge of a cell is split into two edges then $K = 2$). When the anisotropic adaptation is used the grid is refined only in the direction normal to the shock wave and the number of the cells would be increased to $N\,K$.

The adaptation algorithm presented in this paper relies on remeshing of the grid. The remeshing is based on generation of a new grid in the Riemann space in which uniform spacing of the cells is assumed. The final spacing of the new grid is a function of metric field used for definition of the Riemann space. Detailed description of the anisotropic grid generation algorithms can be found in [4] and [2].

The metric used for definition of the Riemann space is obtained from the solution field by means of the presented error estimator. The estimator provides data in a form of metric tensor which describes the directionality of flow features. Few possibilities exist with regard to the choice of the error estimator. In the present case the flow solver is using linear reconstruction within each cell. Therefore it is convenient to use estimation of interpolation error as the error indicator ([3], [4]).

1.1 Calculation of the metric

Assume that E denotes a grid cell inside which a function u is being interpolated and $\mathbf{x}_c$ is the center of E. Then after dropping terms of higher order, the interpolation error for E can be estimated [4] as:

$$\varepsilon_E \leq \max_{\mathbf{x}\in E} \left(\mathbf{x}-\mathbf{x}_c\right)^T |\mathcal{H}| (\mathbf{x}-\mathbf{x}_c\big) \tag{1}$$

where $\mathcal{H}$ is a Hessian of u:

$$\mathcal{H} = \begin{bmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x\,\partial y} & \frac{\partial^2 u}{\partial x\,\partial z} \\ \frac{\partial^2 u}{\partial x\,\partial y} & \frac{\partial^2 u}{\partial y^2} & \frac{\partial^2 u}{\partial y\,\partial z} \\ \frac{\partial^2 u}{\partial x\,\partial z} & \frac{\partial^2 u}{\partial y\,\partial z} & \frac{\partial^2 u}{\partial z^2} \end{bmatrix} = R\cdot\Lambda\cdot R^{-1} \tag{2}$$

The Hessian is symmetric, therefore it has real eigenvalues λ_1, λ_2 and λ_3. The matrix R consist of columns created from right eigenvectors of $\mathcal{H}$ and Λ is a diagonal matrix consisting of corresponding eigenvalues λ_i. Consequently $|\mathcal{H}|$ can be defined as:

$$|\mathcal{H}| = R\cdot \begin{bmatrix} |\lambda_1| & 0 & 0 \\ 0 & |\lambda_2| & 0 \\ 0 & 0 & |\lambda_3| \end{bmatrix} \cdot R^{-1} \tag{3}$$

The interpolation error in a given direction defined by unit vector $\mathbf{w}$, is proportional to a constant C calculated as follows:

$$C = h^2\ \mathbf{w}^T |\mathcal{H}| \mathbf{w} \tag{4}$$

where h denotes a length of a cell in the direction of unit vector $\mathbf{w}$. In the present approach $\mathbf{w}$ becomes a direction of a given edge and h becomes an edge length. Construction of an optimal grid is based on equidistribution of an interpolation error. This is equivalent to assuming that for every edge e_i, the constant C_i is equal to a global C. We can introduce now a scaled metric $\mathcal{M}$:

$$\mathcal{M} = C^{-1}\ |\mathcal{H}| \tag{5}$$

In the last step the eigenvalues are limited to avoid unrealistic values:

$$\hat{\lambda_i} = \frac{1}{\max(h_{min}, \min(h_{max}, h_i))} \tag{6}$$

where values of h_{min} and h_{max} are provided by the user.
The final metric is taken as:

$$\mathcal{M} = R \cdot \begin{bmatrix} \hat{\lambda}_1 & 0 & 0 \\ 0 & \hat{\lambda}_2 & 0 \\ 0 & 0 & \hat{\lambda}_3 \end{bmatrix} \cdot R^{-1}$$

The metric $\mathcal{M}$ is subsequently calculated for every node of the old grid and by means of interpolation is extended to form a continuous metric field. The new grid is generated in the Riemann space defined by the metric field in such a way that all edges have approximately unit length.

1.2 Reconstruction of the Hessian

The key problem in evaluation of the metric tensor lays in reconstruction of the Hessian ([5]). The approach presented in this paper is based on a very robust and efficient Green formula.

Function u used for adaptation is given in a discrete manner and is known only at the grid nodes. Then by assuming that solution inside cells is linear the Green formula can be used:

$$\widetilde{\nabla u} = \frac{1}{\Omega} \oint_{\partial\Omega} u \, \mathbf{n} \, dS \tag{7}$$

which allows to find an averaged gradient $\widetilde{\nabla u}$ by simple contour integration. The Ω denotes the control volume which for the purpose of Hessian reconstruction consist of cells which are neighbours of a given node. The calculations are performed then in two steps:

1. Calculation of ∇u at grid nodes with the Green formula
2. Calculation of $\mathcal{H} = \nabla(\nabla u)$ at grid nodes with the Green formula

1.3 Metric intersection

The metric field described above can be obtained for any scalar field. When the solution is defined by more then one variable (e.g. density, velocity vector, energy) the scalar field used for adaptation should be carefully chosen because some flow features may not be visible in a selected field (e.g. slip lines are not visible in the pressure field).

The present approach is based on mixing few metric fields calculated for different flow parameters in such a way that the result will minimize the grid spacing. It can be done at every grid node by finding an intersection of metrics corresponding to different flow parameters. The metric can be interpreted as a ellipse in 2D or ellipsoid in 3D. Therefore the problem of minimization of the grid spacing is equivalent to finding a new metric

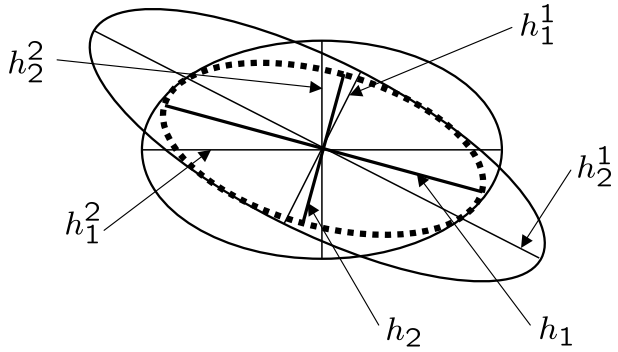


Figure 1: Graphical representation of two metrics (solid line) and the resulting intersection metric (dotted line)

which fits the space created by intersection (see Fig. 1). Since the exact algorithm is very expensive an approximate one is used ([4]).

Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be the metrics for which the minimum metric $\mathcal{M}$ is sought. The e_i^1 and λ_i^1 are the i-th unit eigenvector and eigenvalue of metric $\mathcal{M}_1$ respectively. Then a new metric $\hat{\mathcal{M}}_1$ can be created by finding its eigenvalues from:

$$\hat{\lambda}_i^1 = \max\left(\lambda_i^1,\ (e_i^1)^T \cdot \mathcal{M}_2 \cdot e_i^1\right) \tag{8}$$

and assembling the metric according to:

$$\hat{\mathcal{M}}_1 = R_1 \cdot \hat{\Lambda}_1 \cdot R_1^{-1} \tag{9}$$

where for 3D:

$$\hat{\Lambda}_1 = diag(\hat{\lambda}_1^1,\ \hat{\lambda}_2^1,\ \hat{\lambda}_3^1) \tag{10}$$

In the same way the metric $\hat{\mathcal{M}}_2$ can be created. Then the final approximation can be done by a simple averaging procedure:

$$\mathcal{M} = \frac{1}{2}\ \left(\hat{\mathcal{M}}_1 + \hat{\mathcal{M}}_2\right) \tag{11}$$

This procedure can be also extended to the case when more then two metric fields has to combined.

2 UNSTEADY FLOW ADAPTATION ALGORITHM

The main problem arising in adaptation of unsteady flows is the fact that it is next to impossible to predict in advance the motion of flow features (e.g. shockwaves). Thus the adapted grid may not be refined accordingly to the future position of the flow feature. In particular, the shock wave leaving the refined area may drastically deteriorate quality of the solution or even crash the code. The only reasonable procedure is to track flow features

on the grid which is not yet adapted. Then new grid should be created to accommodate all future positions of the flow features along the adaptation time step [1].

For adaptation of the unsteady flow the time step $\Delta t = [t_0, t_{max}]$ is divided into smaller steps $\Delta\tau$. The length of $\Delta\tau$ is chosen according to the flow problem, trying to keep balance between small values (too many interpolations which can deteriorate conservation property of the scheme) and large ones (cell size becomes to large). For each time step $\Delta\tau$ quasi-steady adaptation loop is performed. In this case however the definition of metric field used for generation of the new grid is different. Namely, in adaptation loop, the metric is calculated as an intersection of many metric fields obtained for snapshots of the solution taken during $\Delta\tau$ step.

In the description of the algorithm following symobols are used:

$\Delta\tau$ - defines a length of the adaptation time step

G_i^k - defines a grid after k levels of adaptation for an adaptation time step i

$S_i^k(\tau)$ - defines a solution for i-th adaptation time step on the grid G_i^k for time τ

I_i - defines a solution field used as a definition of initial solution for adaptation time step i. It does not depend on the adaptation level.

l - is a number of solution snapshots used for calculation of the metric.

$M_{i,j}^k$ - defines a metric field calculated for the solution field $S_i^k(\tau_0 + j\Delta\tau/(l-1))$

M_i^k - defines a metric field used for generation of the grid G_i^{k+1}.

The algorithm:

I. Generate uniform grid G^0

II. Set $i \leftarrow 0$ and $\tau_0 \leftarrow t_0$

III. Define initial solution I_0 for t_0

IV. Set $k \leftarrow 0$

V. Start adaptation loop for time period $[\tau_0, \tau_0 + \Delta\tau]$ using G^0 as the grid G_i^0

1. Interpolate initial solution I_i on a new grid G_i^k to obtain a solution field $S_i^k(\tau_0)$
2. Solve flow problem starting from $S_i^k(\tau_0)$ and store solutions $S_i^k(\tau_0 + j\Delta\tau/(l-1))$ where $j = 0, ..., l-1$
3. If k is greater or equal k^{max} stop adaptation loop and jump to point VI

4. Using error estimator find metric field $M^k_{i,j}$ for every stored solution $S^k_i(\tau_0 + j\Delta\tau/(l-1))$
5. Find a metric field M^k_i by intersecting metric fields $M^k_{i,j}$ $j = 0, ..., l-1$
6. Generate a new grid G^{k+1}_i using metric field M^k_i
7. Return to point 1 setting $k \leftarrow k+1$.

VI. Save solution $S^{k^{max}}_i(\tau_0 + \Delta\tau)$ as initial solution I_{i+1}

VII. Set $i \leftarrow i+1$ and $\tau_0 \leftarrow \tau_0 + \Delta\tau$

VIII. If τ_0 is less than t_{max} jump to point IV.

3 NUMERICAL RESULTS

In this section the results of numerical verification of the adaptation algorithm are presented. First example of the steady problem shows the advantages of the anisotropic approach to adaptation. The other two testcases, first 2D and second 3D provide results obtained by using the adaptation algorithm applied to unsteady compressible flow.

3.1 Adaptation of the MHD steady flow in a sinusoidal 3D channel

The first testcase is the steady MHD flow through a 3D channel with sinusoidal bump, with supersonic inlet flow. The wall is fully conducting, hence vanishing normal B-field is imposed. Inside the channel the pattern of oblique shock waves should be resolved. The initial grid and the grid after the 3^{rd} adaptation are shown on the Figure 2. The corresponding field of magnitude of magnetic induction vector is shown on Figure 3. It is worth noting that significant improvement of the solution was obtained while the number of points was increased by a factor less than 1.6.

3.2 Adaptation of the unsteady Euler flow for a 2D channel with forward facing step

The next testcase is the unsteady Euler flow past a forward facing step in a channel (Mach at inlet equal 3.0). The solution of this problem has been presented in [7]. It represents a flow with complex unsteady shock wave interacting with the solid walls of the channel.

The initial grid with uniform spacing $h = 0.02$ is presented on Figure 4a. The adaptation time step is equal to $\Delta\tau = 0.02$, which means that 150 adaptations were performed before reaching global time $t = 3.0$. For each time step $\Delta\tau$, three levels of adaptations were done gradually increasing the number of nodes. The grid for the final level of adaptation was created with setting a minimum cell size to 0.0015. The number of nodes for

the finest grid for time $t = 3.0$ consisted of 42193 nodes. A uniform grid with equivalent resolution would roughly contain 1300000 of nodes.

The grids used for calculations for time $t = 1.0$ are presented on Figure 4 and the corresponding density fields can be seen on Figure 5. The final grid after third adaptation and the density field for time $t = 3.0$ can be seen on Figure 6. The improvement of the solution obtained by means of adaptation is clearly visible - the shock waves are thin, the slip lines starting from triple points have been fully recovered.

The solution of the Euler equations has been obtained using 2nd order Cell Centered FVM with WENO reconstruction.

3.3 Adaptation of the unsteady 3D Euler flow past sphere

This testcase presents results obtained using the unsteady adaptation for 3D Euler flow past a sphere with Mach number of 3.0. For this condition a bow shock appears in front of the sphere and reaches the steady position as time tends to infinity.

The calculations were performed using COOLFluiD FVM code implementing first order Lax-Friedrichs scheme. The adaptation time step was chosen $\Delta\tau = 0.001$, and for each step two adaptations were done. The sequence of grids for time $t = 0.003$ and $t = 0.019$ are presented on Figure 7. The corresponding density fields are shown on Figure 8.

Once again applying adaptation improved the quality of the results without significantly increasing number of nodes.

4 CONCLUSIONS AND FUTURE WORK

Presented anisotropic adaptation algorithm has been successfully applied to 2D and 3D time dependent inviscid flows. The adaptation has improved the quality of the solution and recovered details of flow which were missing in the solution based on uniform grid. The presented algorithm is capable to locally predict the grid areas that should be adapted.

The present error estimator relies on assumption that a solver used for calculations is second order accurate. This assumption is almost never true since most solvers switch to the first order in the vicinity of the shocks. Therefore more accurate approach to define metric field should be used. Similarly higher order solvers require error estimators based on different principle.

Error estimator should also be improved to produce better metric fields for strong shocks. The approach based on Hessian reconstruction presented in this paper has problems with resolving strongly anisotropic fields (e.g. around strong shock waves). Small variation of the solution along the shock wave (caused by numerical representation of the shock) is strong enough to trigger almost isotropic refinement.

It is also necessary to improve the grid generator in order to properly handle anisotropy near the boundaries. At this moment 3D boundary recovery step is not functioning properly. To temporarily overcome this problem the surface grid has been adapted using isotropic algorithm (the volume grid is modified using fully anisotropic approach). It

introduces localy more new nodes than it could for anisotropic boundary. However, since most points are inside the grid, the growth factor is roughly equal to 1.3 (it strongly depends on geometry and flow structure).

Further testing for the fully 3D unsteady flows is still necessary.

ACKNOWLEDGEMENTS

The work was done with cooperation with Von Karman Institute and was supported by Belgian Federal Science Policy Office.

REFERENCES

[1] Alauzet, F., George, P.L.,Mohammadi, B., Frey, P., Borouchaki, H., *Transient fixed point-based unstructured mesh adaptation*, Int. J. Numer. Meth. Fluids, 43, 729-745, 2003.

[2] Athanasiadis, A.N., *Three-Dimensional Hybrid Grid Generation with Application to High Reynolds Number Viscous Flows*, Ph.D. thesis, Von Karman Institute, Universite Libre de Bruxelles, 2004.

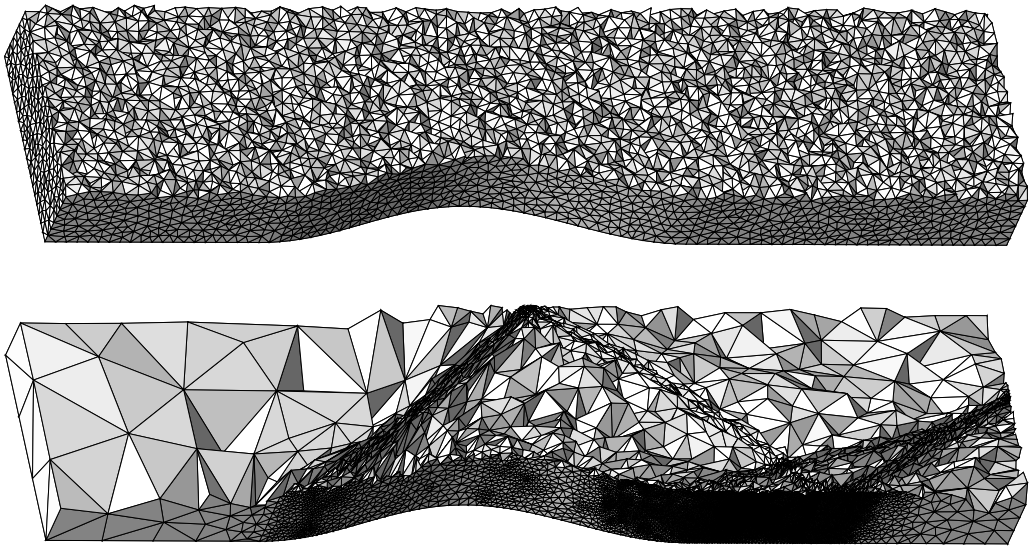
[3] Castro-Daz, M.J., Hecht, F., Mohammadi, B., Pironneau, O., *Anisotropic Unstructured Mesh Adaptation for Flow Simulation*, International Journal for Numerical Methods in Fluids, 25, 4, pp. 475-491, 1997.

[4] George, P.L., Borouchaki,H., *Delaunay Triangulation and Meshing - Application to Finite Element*, Hermes, ISBN 2-86601-692-0, Paris,1998.

[5] Formaggia, L., Perotto, S., *Anisotropic Error Estimation for Finite Element Methods*, 31st Computational Fluid Dynamics, VKI Lecture Series 2000-05, 2000.

[6] Majewski, J. , *An anisotropic adaptation for simulation of compressible flows*, Mathematical Modelling and Analysis, Volume 7, p. 127-134, 2002.

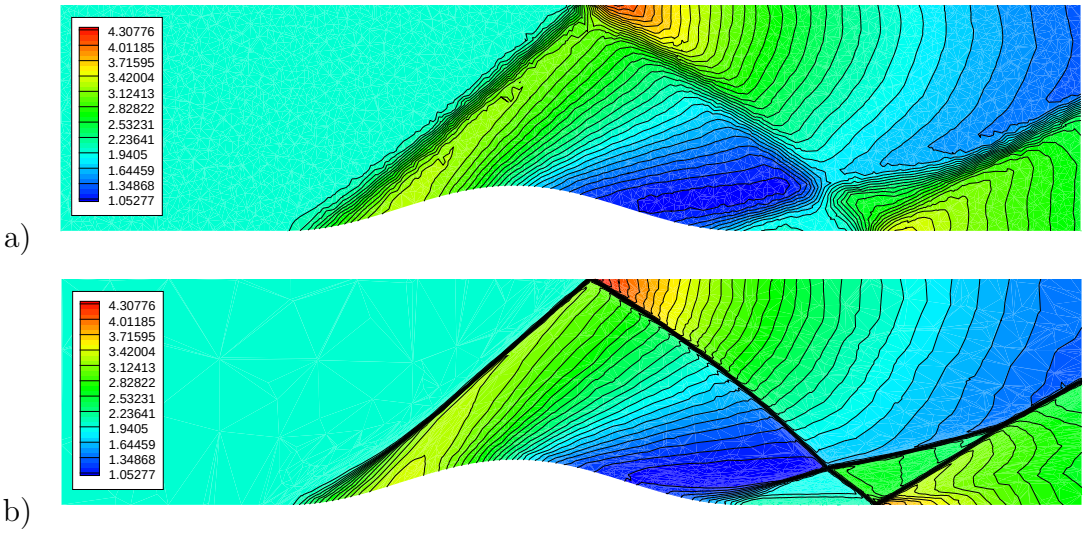
[7] Woodward, P.R., Colella, P., *The numerical simulation of two-dimensional flows with strong shocks*, J. Comput. Phys., 54, 115-173, 1984.



a)

b)

Figure 2: Grids used for calculation of the MHD problem inside a channel with sinusoidal bump:
a) initial grid (23117 nodes)
b) grid after 3rd adaptation loop (36361 nodes)



a)

b)

Figure 3: The field of the magnitude of the magnetic induction vector:
a) calculated on the initial grid
b) calculated on the grid after 3rd adaptation loop

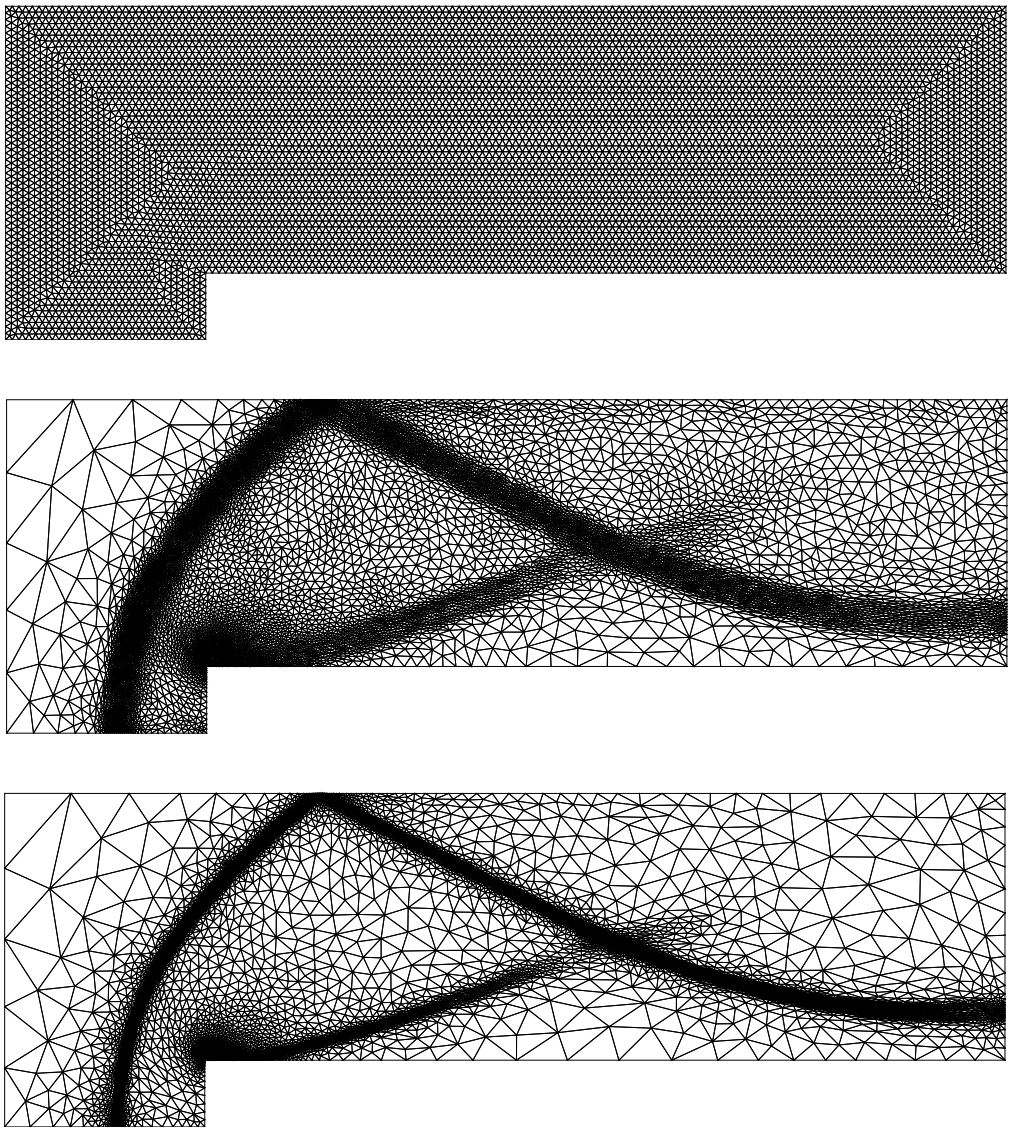
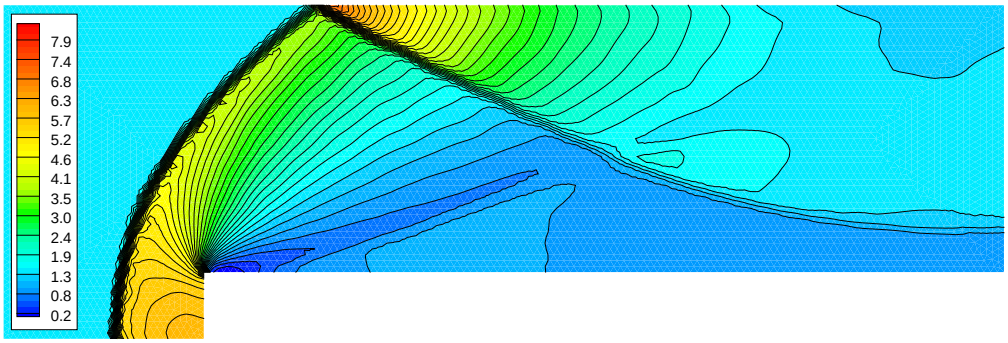
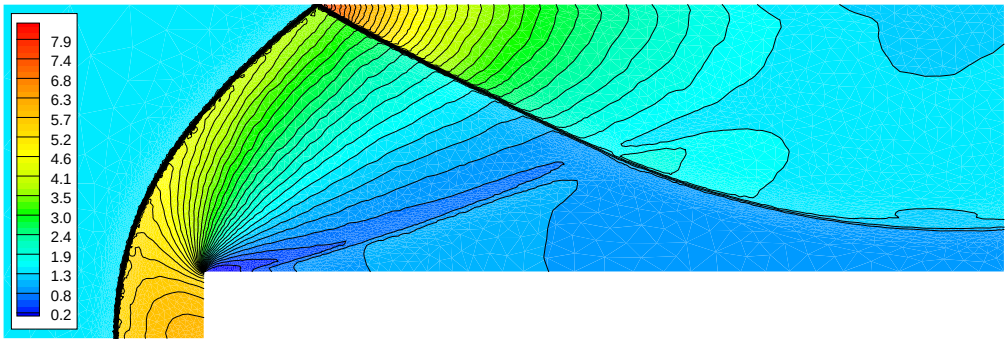


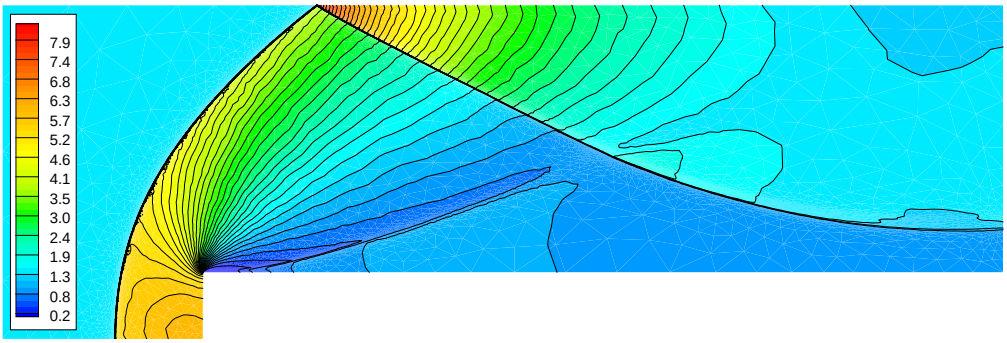
Figure 4: The grids for different levels of adaptations for $t = 1.0$:
a) grid with uniform spacing used for calculations without adaptation (7444 nodes)
b) grid after first adaptation (8553 nodes)
c) grid after second adaptation (22168 nodes)



a)



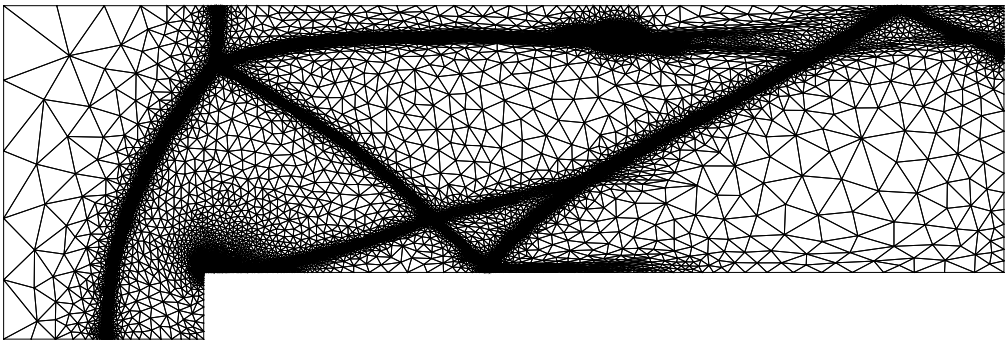
b)



c)

Figure 5: The density field for different levels of adaptations for $t = 1.0$:
a) without adaptation
b) after first adaptation
c) after third adaptation

a)



b)

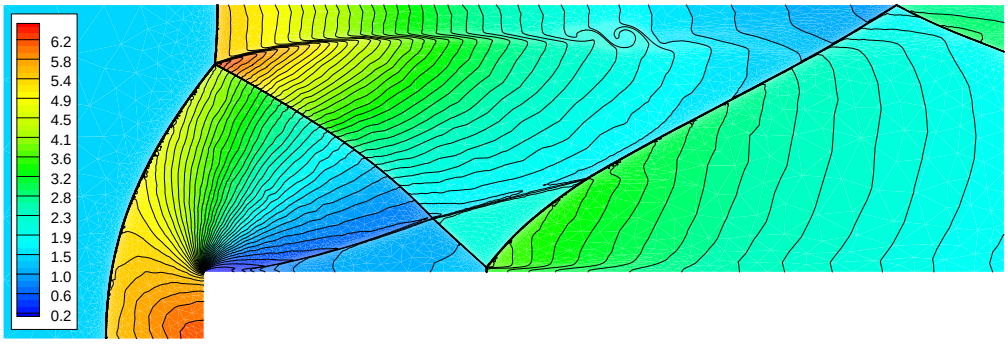


Figure 6: Results after third adaptation for $t = 3.0$: a) grid (42193 nodes) b) density field

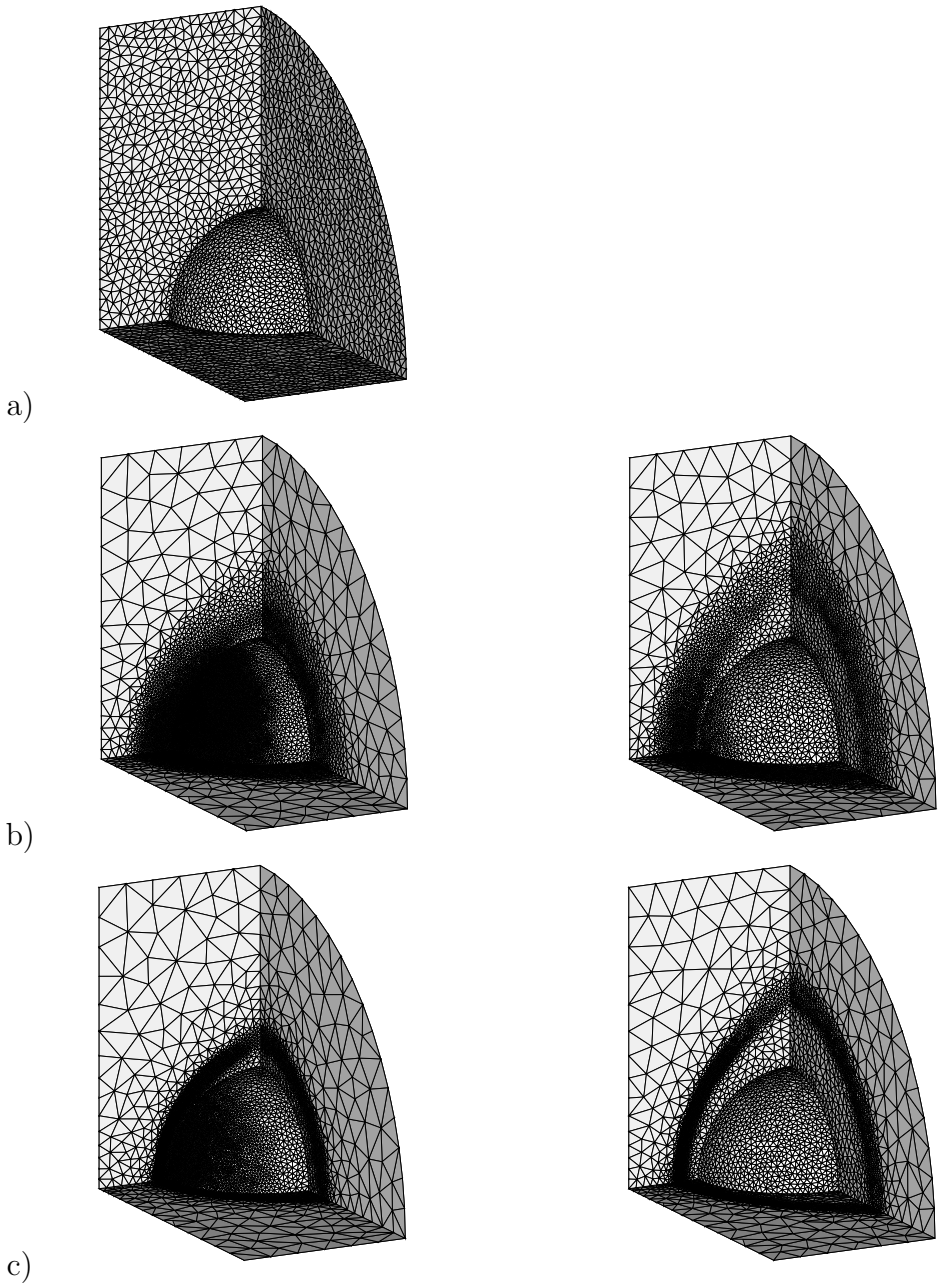


Figure 7: The grids for different levels of adaptations for the unsteady flow past the sphere (left column $t = 0.003$, right column $t = 0.019$) :
a) without adaptation - uniform grid
b) after first adaptation
c) after third adaptation

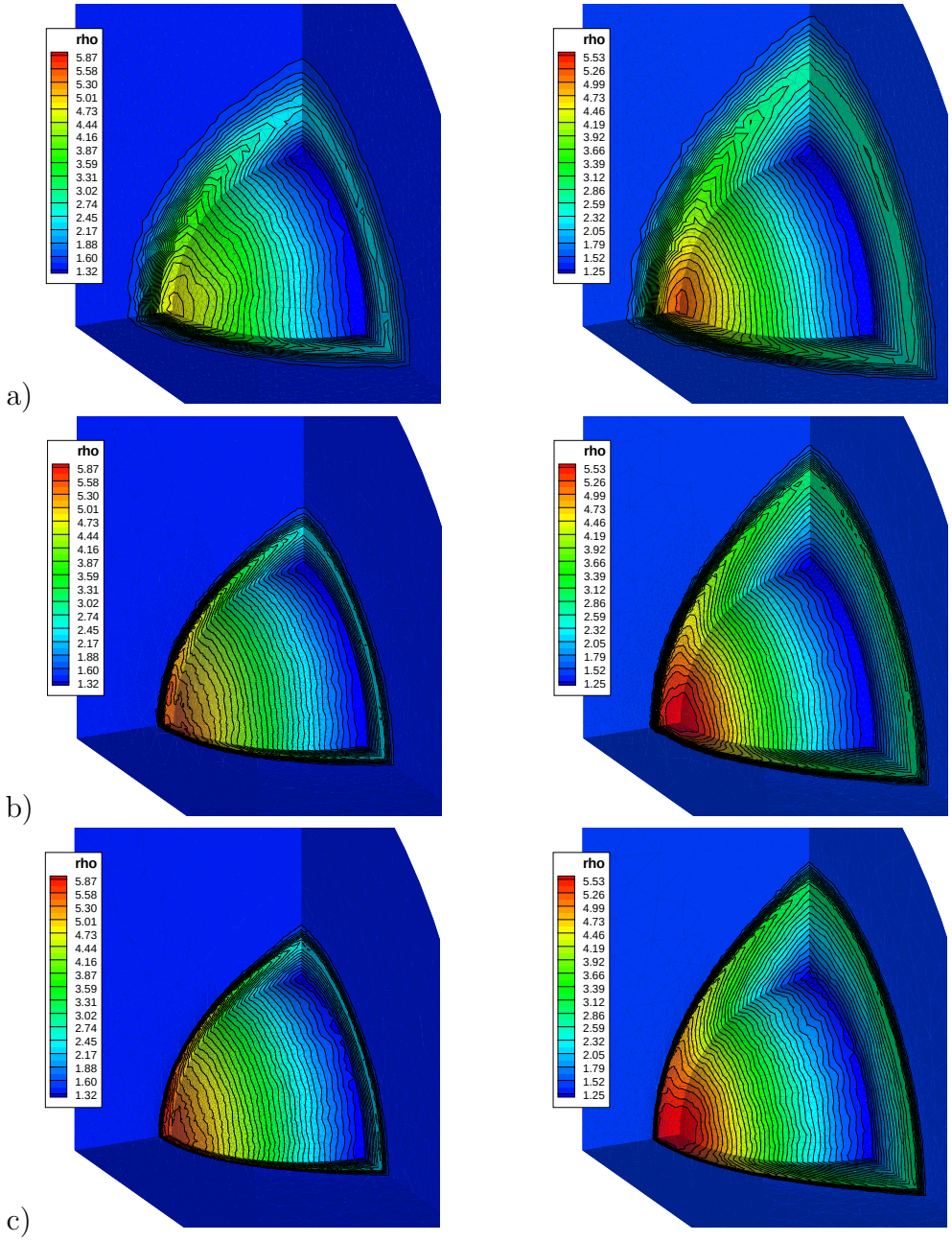


Figure 8: The density field for different levels of adaptations for the unsteady flow past the sphere (left column $t = 0.003$, right column $t = 0.019$) :
a) without adaptation
b) after first adaptation
c) after second adaptation