

AN ANISOTROPIC ADAPTATION FOR SIMULATION OF COMPRESSIBLE FLOWS

J. MAJEWSKI

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

Received September 20, 2001; revised December 21, 2001

ABSTRACT

This paper is dedicated to problems connected with an anisotropic adaptation used for simulation of compressible flows with FVM solver. The adaptation is based on error indicator defined by 2D interpolation error. The comparison between isotropic and anisotropic adaptation is shown for typical test case of supersonic flow in a channel.

1. INTRODUCTION

Adaptation of a grid is a very powerful tool for optimization of flow calculations in both cost and memory usage. Many implementations are based on ad hoc ideas which do not take into account a directional properties of a solution.

The anisotropic adaptation has been first presented in [1; 2; 5; 6]. This technique is more powerful because grid is refined only in direction which demands better resolution. As it will be shown in the paper, it is possible to create a grid with smaller number of grid nodes than with isotropic adaptation, keeping the error of the solution at the similar level.

The error indicator used for adaptation follows algorithms presented in [1; 5]. Generation of the new grid is based on an original algorithm which uses an anisotropic Delaunay triangulation of a set of new points obtained from modifications of the grid from the previous adaptation step. To avoid solver problems when dealing with very thin triangles present in anisotropic meshes the author uses Finite Volume Method coupled with the WENO (Weighted Essentially Non-Oscillatory) reconstruction [4].

2. ADAPTATION ALGORITHM

Adaptation algorithm can be described in the following way:

1. Generate initial grid G^k ($k = 0$) which may not reflect any flow features yet.
2. Solve equations on the grid G^k .
3. Check the criterion to stop the adaptation loop (e.g., number of grid nodes between G^k and G^{k-1} does not vary significantly). If this criterion is satisfied, adaptation loop is finished.
4. Calculate the metric tensor $\mathcal{M}(\mathbf{x})$ in the flow domain using error estimator based on 2D interpolation error for G^k . The metric is stored in a discrete way at each node of grid G^k . This, by means of interpolation, allows us to find the value of \mathcal{M} at any point of the flow domain.
5. Generate a new grid G^{k+1} in a Riemann space with the metric defined in the point 4.
6. Return to point 2 setting $k \leftarrow k + 1$.

3. CALCULATION OF THE METRIC

Computation of the metric is based on the estimation of the interpolation error. For linear interpolation this error is proportional to a second derivative of an adapted function u .

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 as [5]:

$$\varepsilon_E \leq \max_{\mathbf{x} \in E} |(\mathbf{x} - \mathbf{x}_c)^\top |\mathbf{H}| (\mathbf{x} - \mathbf{x}_c)|, \quad (3.1)$$

where \mathbf{H} is a Hessian of u :

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial x \partial y} & \frac{\partial^2 u}{\partial y^2} \end{bmatrix} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^{-1}. \quad (3.2)$$

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

$$|\mathbf{H}| = \mathbf{R} \begin{bmatrix} |\lambda_1| & 0 \\ 0 & |\lambda_2| \end{bmatrix} \mathbf{R}^{-1}. \quad (3.3)$$

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

$$C = h^2 \mathbf{w}^\top |\mathbf{H}| \mathbf{w}, \quad (3.4)$$

where h denotes a length of a cell in the direction \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 \mathbf{e}_i , the constant C_i is equal to a global C . We introduce a scaled metric \mathcal{M} :

$$\mathbf{M} = \mathbf{C}^{-1} |\mathbf{H}|. \quad (3.5)$$

Calculation of edge length can be then performed with metric defined by (3.5). Grid should be constructed so that all edges \mathbf{e}_i have unitary lengths calculated from the formula:

$$\hat{h} = \sqrt{\mathbf{e}_i^\top \mathbf{M} \mathbf{e}_i}. \quad (3.6)$$

Function u used for adaptation is given in a discrete manner. Values of u are known only at cells centers. The calculation of a metric based on a Hessian of u is not a trivial task. Different approaches can be used, e.g., based on Green's formula. In this paper the approach based on the least squares method. The full algorithm of computation of the metric can be presented as follows:

1. Calculation of ∇u at grid nodes with the least squares method [3].
2. Calculation of $\mathcal{H} = \nabla(\nabla u)$ at grid nodes with the least square method [3].
3. Calculation of \mathcal{M} using equation (3.5).
4. Limiting the metric with prescribed values defining maximum and minimum sizes of edge length (h_{min} and h_{max}).

The metric tensor is subsequently extended to the whole domain using simple linear interpolation [5].

4. ALGORITHMS FOR GENERATION OF A NEW GRID

Once the metric is defined, the grid can be created in various ways (e.g., it can be generated from a scratch by using anisotropic Delaunay triangulation). In the presented implementation, a new grid is created by modification of the grid from previous adaptation step, enabling refinement and coarsening of the grid.

4.1. Refinement of the grid

Refinement of the grid is equivalent to inserting new nodes in areas where estimated error is too big. In the present implementation it is necessary to define a threshold value $h_{p_{max}}$ used for comparison of edge length $d_{\mathcal{M}}$ calculated in the Riemann space. If a given edge satisfies

$$d_{\mathcal{M}} > h_{p_{max}}, \quad (4.1)$$

then a new node is introduced in the middle point of this edge. The node is inserted into a list of all nodes of the new grid. If the given edge is a part of the curved boundary, position of the new point must be corrected to allow for proper representation of the boundary.

4.2. Coarsening of the grid

Coarsening of the grid allows to remove nodes from areas where grid is too dense according to the error estimator. In the present implementation it is necessary to define a threshold value $h_{p_{min}}$ used for comparison of edge lengths $d_{\mathcal{M}}$ calculated in the Riemann space. If a specified number of edges connected to a given node satisfies

$$d_{\mathcal{M}} < h_{p_{min}}, \quad (4.2)$$

then the node is removed from a list of all nodes of the new grid.

4.3. Grid smoothing

Threshold values $h_{p_{min}} < h_{p_{max}}$ are chosen both close to 1. When $h_{p_{max}}$ is decreased more nodes are created during refining phase. When $h_{p_{min}}$ is increased more nodes are removed during coarsening phase. Once the refining and coarsening is finished a list of nodes of the new grid is created. All nodes from this list are connected using anisotropic Delaunay triangulation.

The last step consists in grid smoothing. The smoothing uses the algorithm based on spring analogy, in which every edge is treated as a spring connected to two nodes. The boundary nodes are allowed to move only along the boundary. The iterative algorithm is used to find the new equilibrium of the system [5].

5. NUMERICAL RESULTS

5.1. Supersonic flow in 15° wedge channel

The performance of the adaptation algorithm is presented for supersonic flow in a channel with a 15° wedge. Flow at the inlet is of Mach number 2. The structure of the flow is dominated by system of oblique shock waves and expansion fans. Near the top wall of the channel a slip line emerging from a triple Mach point can be seen. This is a very demanding test since shock waves form strong discontinuity in the solution and locally high level of refinement is required.

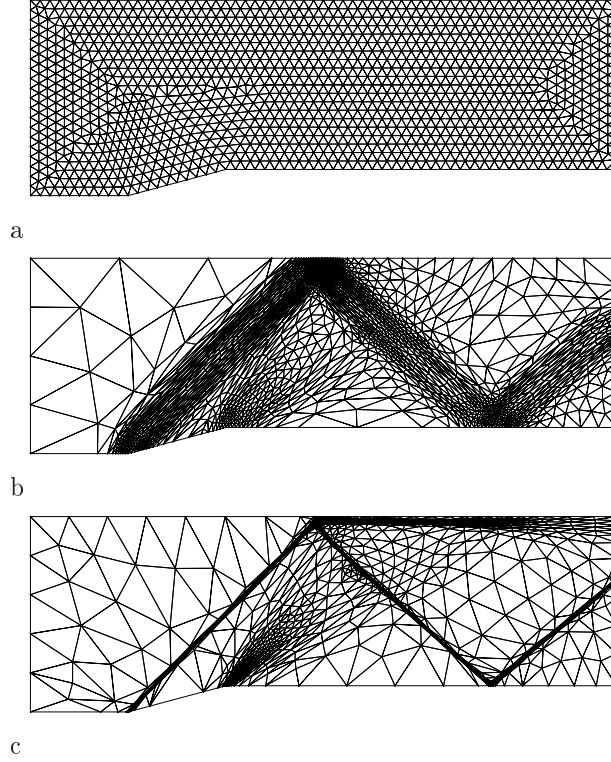


Figure 1. Grids used for calculation of supersonic flow in the channel with 15° : ($h_{p_{max}} \simeq 1.4$, $h_{p_{min}} \simeq 0.8$). a) initial grid (1300 nodes). b) grid after 1^{st} adaptation (1400 nodes). c) grid after 10^{th} and last adaptation (7700 nodes).

The flow solver was based on 2D Euler Finite Volume cell-centered method supplemented with WENO reconstruction [4]. The estimated error of the Mach number field was used as an adaptation function. Fig. 1 shows grids used in calculations and Fig. 2 shows corresponding Mach number fields. It is clearly visible that adaptation results in big improvement of the solution at quite reasonable cost.

5.2. Comparison of isotropic and anisotropic adaptation

This test case was chosen for direct comparison between isotropic and anisotropic adaptation. Calculations for both cases were performed with the same adaptation tool. In the isotropic case the metric tensor \mathcal{M} was calculated from the same formulas (3.3) – (3.6) with both eigenvalues equal to:

$$\max (|\lambda_1|, |\lambda_2|). \quad (5.1)$$

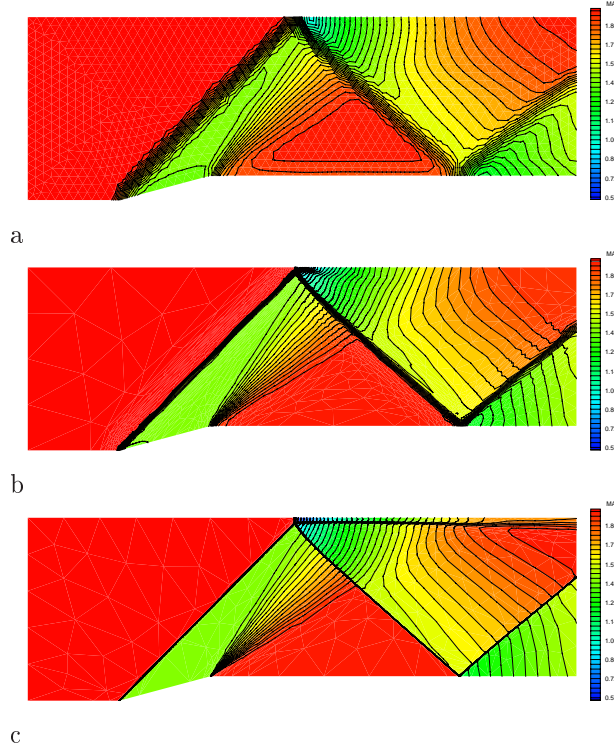


Figure 2. Mach number field for grids shown at Fig. 1 for supersonic flow in the channel with 15° : a) calculated using initial grid, b) calculated using grid after 1^{st} adaptation, c) calculated using grid after 10^{th} and last adaptation.

Three adaptation steps were performed both for isotropic and anisotropic cases. The grids after third adaptation for both cases are shown in Fig. 3 and corresponding Mach number fields are presented in Fig. 4. Details of the grids are presented in Fig. 5.

The Mach number fields obtained on both grids are almost identical. The isotropic grid consists of 28078 nodes while anisotropic one of 6403. In this case the use of anisotropic grid resulted in fourfold reduction of the computing time and memory usage. It is expected that this gain should be even larger for 3D cases.

6. CONCLUSIONS

It was shown in the paper that anisotropic adaptation is a useful tool to improve quality of the numerical solution. Compared with isotropic adaptation the anisotropic adaptation results in much smaller number of grid nodes, re-

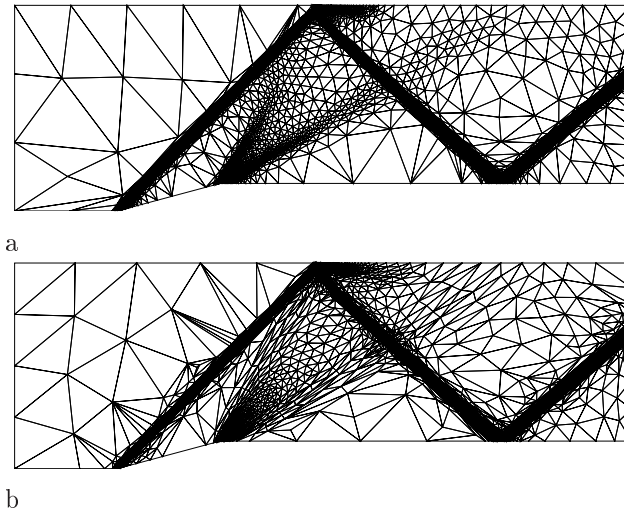


Figure 3. Comparison of isotropic and anisotropic adaptation.
a) grid after third isotropic adaptation (28078 grid nodes).
b) grid after third anisotropic adaptation - (6403 grid nodes).

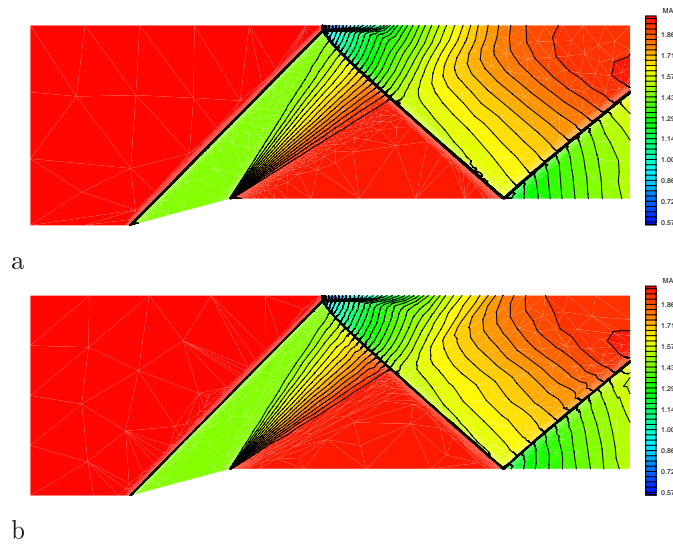


Figure 4. Comparison of isotropic and anisotropic adaptation.
a) Mach number field for grid after third isotropic adaptation.
b) Mach number field for grid after third anisotropic adaptation.

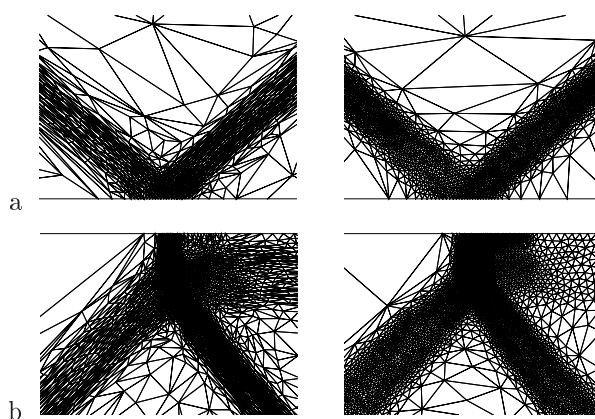


Figure 5. Comparison of isotropic and anisotropic adaptation - details of the grids. Figures on the left present details of the grid after isotropic adaptation and on the right after anisotropic adaptation: a) in the area of the bottom reflection of the oblique shock wave, b) in the area of the top reflection of the oblique shock wave and triple Mach point.

ducing significantly the overall computational effort. The algorithm can be extended to 3D cases in relatively easy manner.

REFERENCES

- [1] M.J. Castro-Diaz, F. Hecht and B. Mohammadi. New progress in anisotropic grid adaptation for inviscid and viscous flows simulations. In: *Proc. 4th International Meshing Roundtable, Sandia National Laboratories*, <http://www.andrew.cmu.edu/user/sowen/abstracts/Ca164.html>, 1995.
- [2] M.J. Castro-Diaz, F. Hecht, B. Mohammadi and O. Pironneau. Anisotropic Unstructured Mesh Adaptation for Flow Simulation. *International Journal for Numerical Methods in Fluids*.
- [3] L. Formaggia and S. Perotto. *Anisotropic Error Estimation for Finite Element Methods*. 31st Computational Fluid Dynamics, VKI Lecture Series 2000-05, 2000.
- [4] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, **144**, 194 – 212, 1998.
- [5] P.L. George and H. Borouchaki. *Delaunay Triangulation and Meshing – Application to Finite Element*. Hermes, ISBN 2-86601-692-0, Paris, 1998.
- [6] W.G. Habashi, M. Fortin and J. Dompierre. *Anisotropic Mesh Optimisation for Structured and Unstructured Meshes*. VKI Lectures Series on CFD VKI LS 1997-02, 1997.

Suspaudžiamų skysčių judėjimo modeliavimas naudojant anizotropinį adaptyvumą

J. Majewski

Darbe pasiūlyta nauja metodika, leidžianti sudaryti anizotropinį adaptyvųjį tinklą. Jame elementai smulkinami tik viena kryptimi. Dvimatis paklaidos indikatorius yra sudarytas remiantis aproksimacijos paklaidos įverčiu. Pateikti skaičiavimo eksperimento rezultatai, kuriame palyginti izotropiniai ir anizotropiniai tinklai.