

## An adaptive mesh refinement in the finite volume method

Avdeev E.V., Fursov V.A.,

Samara State Aerospace University

Ovchinnikov V.A.

Laduga Automotive Engineering Ltd.

**Abstract.** In this paper we describe the method of adaptive mesh refinement, based on the estimation of eigenvalues of discretization matrix. This estimation based on Gershgorin Circle Theorem. This method can be used for unstructured meshes in two-dimensional problems as well as and in three-dimensional. The implementation of the grid adaptation algorithm was made within OpenFOAM open source library of continuum mechanics. This library consists of a set of modules for computational needs, modules written in C++. We give two numerical examples, which show the effectiveness of the proposed method of mesh adaptation.

**Keywords:** adaptive mesh refinement, Gershgorin Circle Theorem, OpenFOAM

**Citation:** Avdeev E.V., Fursov V.A. Ovchinnikov V.A. An adaptive mesh refinement in the finite volume method. Proceedings of Information Technology and Nanotechnology (ITNT-2015), CEUR Workshop Proceedings, 2015; 1490: 234-241. DOI: 10.18287/1613-0073-2015-1490-234-241

### 1. Introduction

Finite volume method (FVM) is a mesh method based on differential equations approximation or on integral equations, which correspond to balance relations. An important property of finite-volume methods is that the balance principles, which are the basis for the mathematical modelling of continuum mechanical problems, per definition, also are fulfilled for the discrete equations conservativity [1].

In the FVM the computational domain is divided into a number of control volumes.

The values are calculated at cell centers. The values of fluxes on the cell interfaces are determined through interpolation of values at the cell centers. As a result we obtain the system of algebraic equations [2].

As in the finite element method, in finite volume method a mesh is constructed, which consists in a partition of the domain where the space variable lives. The elements of the mesh are called control volumes. The most compelling feature of the FVM is that the resulting solution satisfies the conservation of quantities such as mass, momentum, energy, and species. This is exactly satisfied for any control

volume as well as for the whole computational domain and for any number of control volumes.

One of the advantages of the finite volume method over finite difference methods is that it does not require a structured mesh. Finite volume methods are especially powerful on coarse nonuniform grids and in calculations where the mesh moves to track interfaces or shocks.

In general case it is impossible to know a priori how to design an optimal mesh, i.e. mesh with minimal number of cells still satisfying the defined tolerance of the computational error. For transient problems, where the flux is unsteady and the points of interest can reposition during the simulation, uniform mesh becomes ineffective and would need to be very fine to satisfy the error tolerance throughout the whole simulation. To solve this problem a scheme where the mesh self-adapts its structure upon some criteria can be used.

The refinement based commonly on the physical quantity gradient field (see [3] and [4]). Our algorithm based on discretization matrix eigenvalues estimation, which mark cells to be refined or likewise mark cells to be unrefined. This approach has a more effective use of cells and thereby lower computational cost.

## 2. Problem Formulation

Suppose that after equations have been discretized by FVM we obtain the system of algebraic equations, which can be written in matrix form as

$$Ax = b \quad (1)$$

where  $A$  – square  $n \times n$  discretization matrix,  $x$  –  $n \times 1$  column vector of unknown variable,  $b$  –  $n \times 1$  right-hand column vector. The problem consists in finding vector  $x$  whose elements are the values of physical quantity in the cell centers.

It is known that the accuracy of the solution of the problem is largely connected with conditionality of the matrix  $A$ . Perform the following transformation with the expression (1).

$$0 = -Ax + b$$

$$x = (I - A)x + b, \quad (2)$$

Where  $I$  – is the identity matrix. Then rewrite the expression (2) taking into account that we use iterative method for solving, i.e.

$$x_{k+1} = (I - A)x_k + b \quad (3)$$

Note that error of the  $k$ -th and  $(k+1)$ -th iteration respectively can be written as follows:

$$e_k = x - x_k, \quad e_{k+1} = x - x_{k+1}. \quad (4)$$

Subtract (3) from (2)

$$\mathbf{x} - \mathbf{x}_{k+1} = (\mathbf{I} - \mathbf{A})(\mathbf{x} - \mathbf{x}_k)$$

or using (4)

$$\mathbf{e}_{k+1} = (\mathbf{I} - \mathbf{A})\mathbf{e}_k$$

$$\mathbf{e}_{k+1} = \mathbf{M}\mathbf{e}_k \tag{5}$$

Expression (5) shows that for the sequence error reduction at each iteration, it is necessary that for all eigenvalues of the matrix performs:

$$|\lambda(\mathbf{M})| < 1$$

Thus we see that the eigenvalues of the matrix  $\mathbf{M} = \mathbf{I} - \mathbf{A}$  depend on the matrix  $\mathbf{A}$  elements and play an important role in achieving the required accuracy. In particular, they show whether the problem under these conditions converge, converge to the correct solution and how fast.

On other hand, it is known that the elements of matrix  $\mathbf{A}$  and the right-hand side  $\mathbf{b}$  are functions of the mesh spacing. Consequently, there is a relation between conditionality and mesh discretization step. Identification of this relations using approximated equations is not easy task. Moreover, in closed source software, this details are hidden. Therefore we construct procedures of direct analysis of matrix  $\mathbf{M}$  to identify the relations of conditionality and solution accuracy with mesh and adaptive mesh refinement.

### 3. Adaptive mesh refinement algorithm

It is known that the solution of eigenvalues problem is significant and has high computational cost. Moreover, if the case is ill-conditioned, then small eigenvalues can be calculated with low accuracy. Therefore, we have sufficient interest to use simple estimates of eigenvalues, estimates which calculated by the matrix elements insensitive to its conditioning.

Gershgorin circles method [4] is a well-known method for the localization of eigenvalues. According to Gershgorin's Theorem every eigenvalues satisfies:

$$|\lambda - M_{ii}| \leq \sum_{i \neq j} |M_{ij}|, \tag{6}$$

where  $i \in \{1, 2, \dots, n\}$ .

Let  $d_i = \sum_{j \neq i} |M_{ij}|$ . Then the set

$D_i = \{z \in \mathbb{C} : |z - M_{ii}| \leq d_i\}$  is called the  $i$ th Gershgorin disc of the matrix  $\mathbf{M}$ . This disc is the interior plus the boundary of a circle. The circle has a radius  $d_i$  and is centered at (the real part of  $M_{ii}$ , the imaginary part of  $M_{ii}$ ).

Calculating of estimates of the eigenvalues based on equation (6) is the computationally simple task. However, sufficiently strong upper and lower estimations are possible only for diagonally dominant matrices:

$$|M_{ii}| > \sum_{i \neq j} |M_{ij}|.$$

If the matrix is not diagonally dominant, then lower and upper estimations of eigenvalues are undefined.

We offer to predict conditionality and mesh quality associated with conditionality through the right boundary of the Gershgorin circle:

$$F_i = |m_{ii}| + \sum_{i \neq j} |m_{ij}|. \quad (7)$$

This boundary can be easily calculated. The maximum value, which calculated among the boundaries for all Gershgorin circles, defines an upper bound for the maximum eigenvalue of the matrix. Because of change of mesh all eigenvalues “shift” together with appropriate Gershgorin circles. There is reason that changes in the mesh, which lead to increase  $F_i$  in (7), may lead to increase small eigenvalues.

Based on these assumption we have the mesh adaptation algorithm, which based on the analysis of scalar field  $F_i$   $i \in \{1, 2, \dots, n\}$ , which formed by calculating the values of (7) for all mesh nodes. Then mesh adaptation was made based on this scalar field. The normalization of field  $F$  is performed before each iteration:

$$F_i^{\text{normalised}} = \frac{F_i}{\max(F_i)} \quad (8)$$

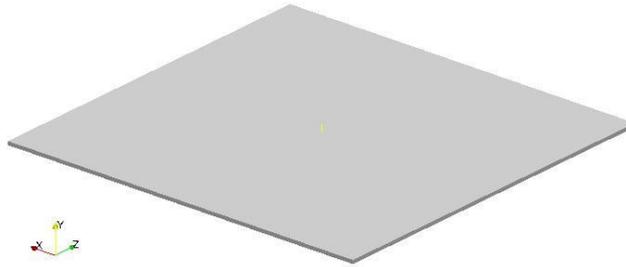
Thus the values of the field  $F$  are within the semi-interval (0;1] and it let us to set in OpenFOAM to refine mesh for cells with  $F_i$  close to 1 and unrefined cells with  $F_i$  around 0. Values of field  $F$  decrease while refining mesh and increase while coarsing mesh.

#### 4. Computational examples and analysis

In order to illustrate feasibility and effectiveness of the algorithm, we use following two examples [5]. Both examples use modified laplacianFoam OpenFOAM solver called laplacianFoamF. New solver has ability to work with dynamic mesh, which allows to adapt mesh.

We compare AMR based on temperature scalar field  $T$ , i.e. based on temperature gradient minimization (AMR  $T$ ) and AMR based on described above scalar field  $F$  (8) – (AMR  $F$ ). For every example we tried to get about the same amount of cells.

Initial geometry of first example, as shown on Fig.1, thin square plate. The length along the  $x$ -axis and  $z$ -axis is 100 meters, along  $y$ -axis is 1 meter.



**Fig. 1.** – Geometry of first example case

Boundary conditions: on the surface  $xy$ ,  $z=0$ : temperature  $T = 1^{\circ}\text{C}$ ; on other 5 surfaces:  $\text{grad}(T) = 10$ . These boundary conditions were chosen for ease of estimation error of the final result. In this case, the temperature decreases linearly from the heated surface.

We want to find out temperature distribution. The heat transfer expressed by Laplace equation:

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0, \quad (9)$$

where for the mathematical treatment is sufficient to consider the case  $\alpha = 1$ .

For comparison, residual plots for both AMR are given (see Fig.2). As you can see from Fig.2, in case with AMR  $F$  adaptation the residual converges little faster than that of the AMR  $T$  case.

In second example we show the work of the AMR  $F$  on a more complex geometry. Geometry and boundary conditions are showed on Fig.3.

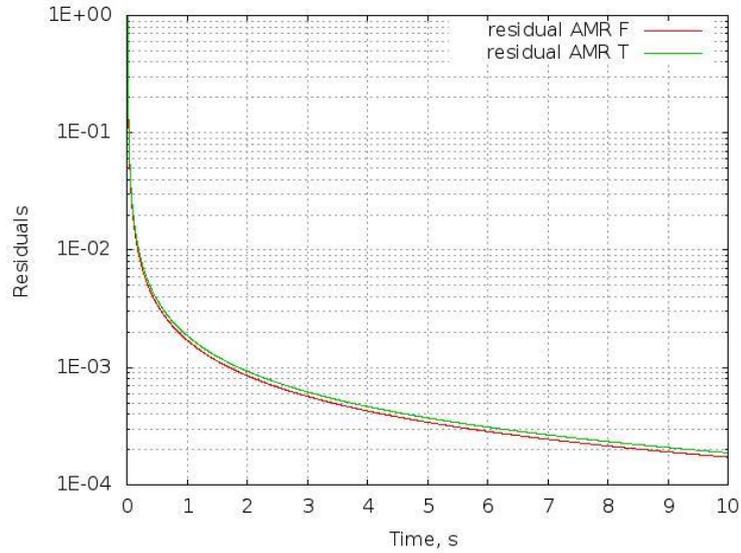
As shown in Fig.3 one surface of flange has temperature  $T_1 = 573^{\circ}\text{C}$  and second has temperature  $T_2 = 273^{\circ}\text{C}$ . On all other surfaces the temperature gradient is set to 0, that walls do not conduct heat (adiabatic walls).

During testing, we found that AMR based on scalar field  $F$  detects too large cells, but does not take into account the boundary conditions (see Fig. 4). This occurs due to the fact that the boundary conditions are contained in right-hand column vector  $b$  of Eq.1, but almost no effect on discretization matrix  $M$ .

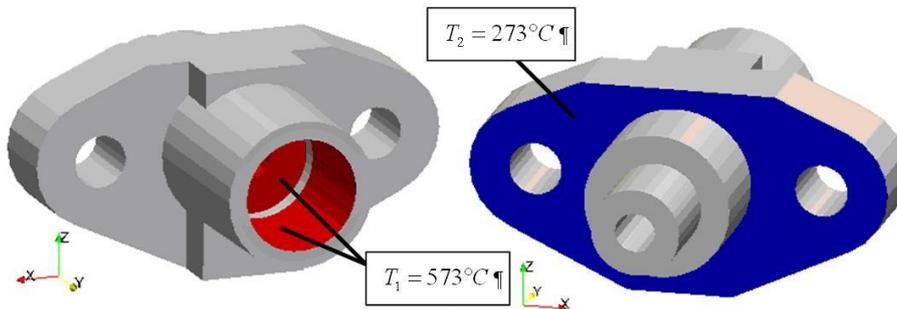
AMR based on temperature gradient field  $T$  vice versa takes into account the boundary conditions, but skips “bad” cells.

Therefore for comparison was made third “hybrid” variant, which include first five iterations of AMR  $F$  and after that it continues with AMR  $T$ . For comparison, residual plots of three cases are given (see Fig. 6).

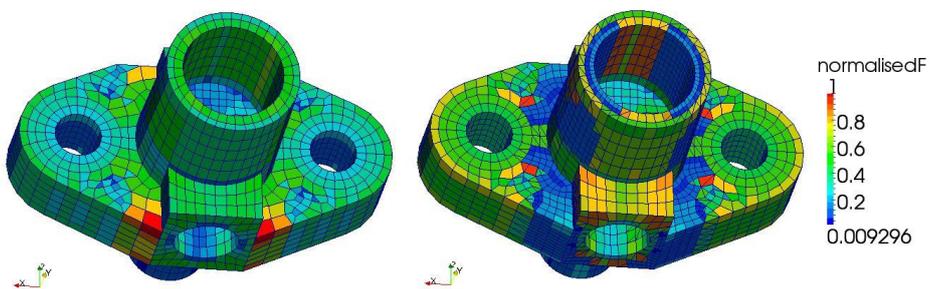
As can be seen from the figure, AMR  $FT$  has the best residual, AMR  $F$  and AMR  $T$  slightly worse.



**Fig. 2.** – First example, AMR *F* and AMR *T* residuals



**Fig. 3.** – Second example, geometry and boundary conditions



**Fig. 4.** – AMR *F* detects and refines too large cells

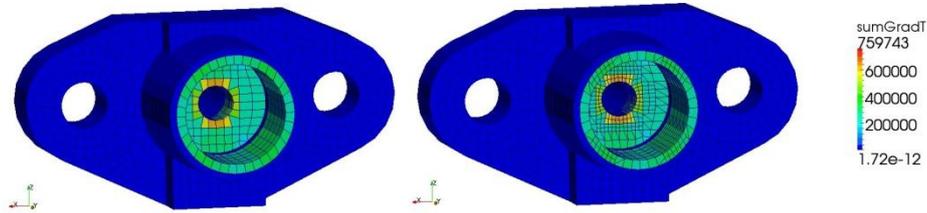


Fig. 5. – AMR *T* takes into account the boundary conditions

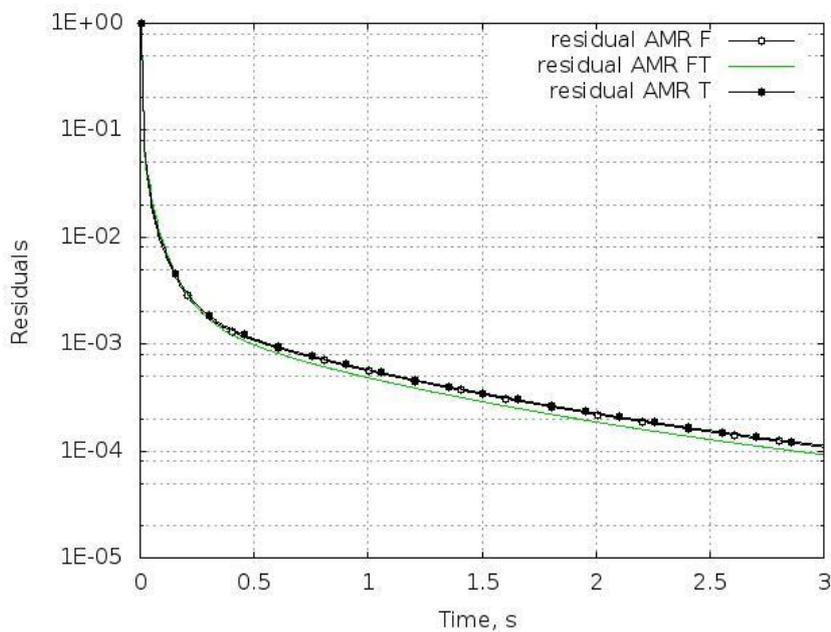


Fig. 6. – Comparison of AMR *F*, AMR *T* and AMR *FT*

**5. Conclusion**

In the proposed method mesh refinement based on discretization matrix conditioning. As can be seen from above two examples – the method does not provide a significant performance increase compared to AMR, based on temperature gradient minimization, but use AMR *T* not always convenient or possible. Such situations are possible in complex problems with dynamic geometry, multiphase flows, etc. Our proposed method allows to choose more suitable AMR settings than in case of AMR *T*.

**Acknowledgements**

In This work was supported by the Ministry of Education and Science of the Russian Federation (project №"2.2335.2014/K").

**References**

1. **Versteeg HK, Malalasekera W.** An introduction to computational fluid dynamics. 2ed., Pearson Ed., 2007; 517 p.
2. **Joubarne E, Guibault F, Braun O, Avellan F.** Numerical capture of wind tip vortex improved by mesh adaptation. *International Journal for Numerical Methods in Fluids*, 2011; 67(1): 8-32.
3. **Escobar JA, Ramirez S, Jimenez RA, Giraldo AM, Silva C, Lopez OD, Ochoa N, Mahecha J, Leguizamon S.** Numerical simulation of NASA wing-trap model as a Colombian Contribution to the High-Lift Prediction workshop. 30th AIAA Applied Aerodynamics Conference Special Session: CFD High Lift Prediction Workshop Follow-on II, 2012.
4. **Brakken-Thal P.** Gershgorin's Theorem for Estimating Eigenvalues. Source: <<http://buzzard.ups.edu/courses/2007spring/projects/brakkenthal-paper.pdf>>.
5. Source: <<https://github.com/j-avdeev/laplacianFoamF>>.