

An Adaptive Mesh Refinement Method, Based on Gershgorin Circle Theorem

Avdeev Evgeniy¹, Fursov Vladimir¹, Minaev Evgeniy¹, and Ovchinnikov Valeriy²

¹Department of supercomputers and general informatics, Samara State Aerospace University, 34, Moskovskoye Shosse, Samara, 443086, Russia.

²Laduga Ltd., 71, Mozhaiskoe Shosse, Moskovskaya oblast, Odintsovo, 143000, Russia.

Keywords: Adaptive mesh refinement, Gershgorin circle theorem, OpenFOAM.

Abstract. A new algorithm of dynamic local Adaptive Mesh Refinement (AMR) with support for arbitrary unstructured meshes, as well as for two and three dimensional computation is developed. The new mesh adaptation based on the discretization matrix eigenvalues estimation by Gershgorin Circle Theorem. The implementation of the algorithm is done within the framework of the OpenFOAM library for Computational Continuum Mechanics (CCM) using C++ programming language with modern policy based design for high program code modularity. It is possible to combine mesh adaptation criteria. Two numerical examples illustrate the effectiveness of new adaptive mesh refinement algorithm.

Introduction

Finite Volume Method (FVM) is a mesh based method where the computational domain is subdivided into several cells, together creating a mesh. When performing a computation for a general Computational Fluid Dynamics (CFD) case it is impossible to know a priori how to design an optimal mesh, i.e. a 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, this becomes most problematic since a static uniform mesh would need to be very fine to satisfy the error tolerance throughout the whole simulation. To solve this dilemma 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 [1], [2]. Our algorithm based on discretization matrix eigenvalues estimation [3], which will mark cells to be refined or likewise mark cells to be unrefined. This approach is a more effective use of cells and thereby lower computational cost.

Discussed Problems

After having used numerical methods such as FVM to discretize governing equations of unknown variables, we usually obtain the form of the system of algebraic equations as shown in the equation below.

$$\mathbf{Ax} = \mathbf{b}, \quad (1)$$

where \mathbf{A} – square sparse $n \times n$ discretization matrix, \mathbf{x} – $n \times 1$ column vector of unknown variable, \mathbf{b} – $n \times 1$ right-hand column vector.

As discussed by Voevodin [4] error estimation

$$\delta \mathbf{x} \leq \frac{K(\mathbf{A})}{1 - K(\mathbf{A})} (\delta \mathbf{A} + \delta \mathbf{b}), \quad (2)$$

where $\delta \mathbf{A} = \|\Delta \mathbf{A}\|/\|\mathbf{A}\|$, $\delta \mathbf{b} = \|\boldsymbol{\varepsilon}\|/\|\mathbf{b}\|$ and $\Delta \mathbf{A}$ - perturbation matrix \mathbf{A} , $\boldsymbol{\varepsilon}$ - perturbation right-hand column vector of equation (10), $K(\mathbf{A}) = \|\mathbf{A}^{-1}\| \|\mathbf{A}\|$ - so-called condition number. Also, we will use equation

$$K(\mathbf{A}) = \frac{\lambda_{\max}}{\lambda_{\min}}, \quad (3)$$

where λ_{\max} , λ_{\min} - maximum and minimum eigenvalues of \mathbf{A} , respectively. From equations (2), (3) we can see that condition number plays important role in (1). Condition number shows how much errors are propagated. cost.

The Estimation Algorithm of Eigenvalues

The eigenvalues estimation algorithm, which we use in our work, based on Gershgorin Theorem and Gershgorin disk definition [5].

Every eigenvalue of matrix \mathbf{A}_{nm} satisfies:

$$|\lambda - \mathbf{A}_{ii}| \leq \sum_{i \neq j} |\mathbf{A}_{ij}|, \quad (4)$$

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

Let $d_i = \sum_{j \neq i} |\mathbf{A}_{ij}|$. Then the set $D_i = \{z \in \mathbb{C} : |z - \mathbf{A}_{ii}| \leq d_i\}$ is called the i th Gershgorin disc of the matrix \mathbf{A} . 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 \mathbf{A}_{ii} , the imaginary part of \mathbf{A}_{ii}).

In our work as a grid refinement criterion was used a scalar field

$$\mathbf{F}_i = a_{ii} + \sum_{i \neq j} a_{ij}, \quad (5)$$

where $i \in \{1, 2, \dots, n\}$. Then mesh adaptation was made based on this scalar field. The normalization of field \mathbf{F} is performed before each iteration:

$$\mathbf{F}_i^{\text{normalised}} = \frac{\mathbf{F}_i}{\max(\mathbf{F}_i)} \quad (6)$$

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

Computational Examples and Analysis

In order to illustrate feasibility and effectiveness of the algorithm, we use following two examples [6]. 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 \mathbf{T} , i.e. based on temperature gradient minimization (AMR T) and AMR based on described above scalar field \mathbf{F} (Eq. 6) - (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.

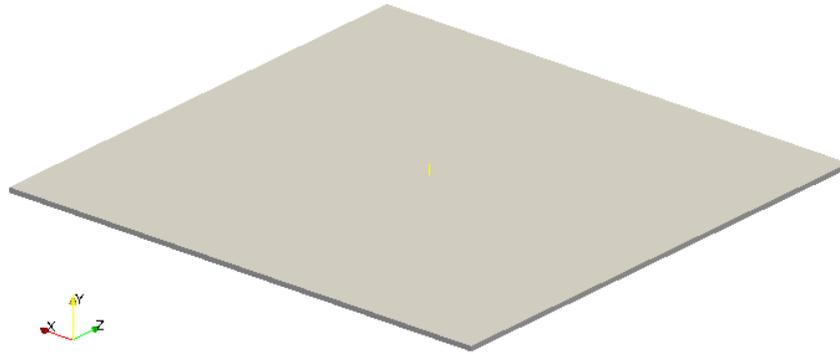


Fig. 1. Geometry of first example case.

Boundary conditions: on the surface $xy, z=0$: temperature $T = 1 \text{ }^\circ\text{C}$; 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:

$$\frac{\partial T}{\partial t} - \alpha \nabla^2 T = 0, \tag{7}$$

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

All dynamic mesh settings are described in OpenFOAM dynamicMeshDict.

```
dynamicRefineFvMeshCoeffs
{
  refineInterval 1;           // How often to refine
  field          normalisedF; // Field to be refinement on
  lowerRefineLevel 0.5;      // Refine field inbetween lower..upper
  upperRefineLevel 1.5;
  unrefineLevel 0.1;        // If value < unrefineLevel unrefine
  nBufferLayers 2;          // Have slower than 2:1 refinement
  maxRefinement 2;          // Refine cells only up to maxRefinement levels
  maxCells 8000;           // Stop refinement if maxCells reached
}
```

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.

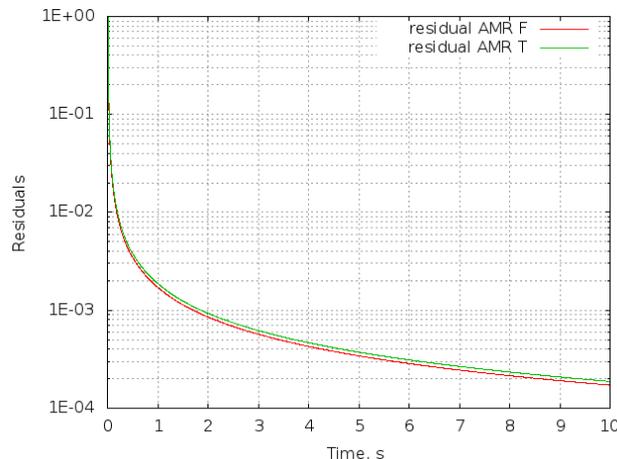


Fig. 2. First example, AMR F and AMR T residuals.

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.

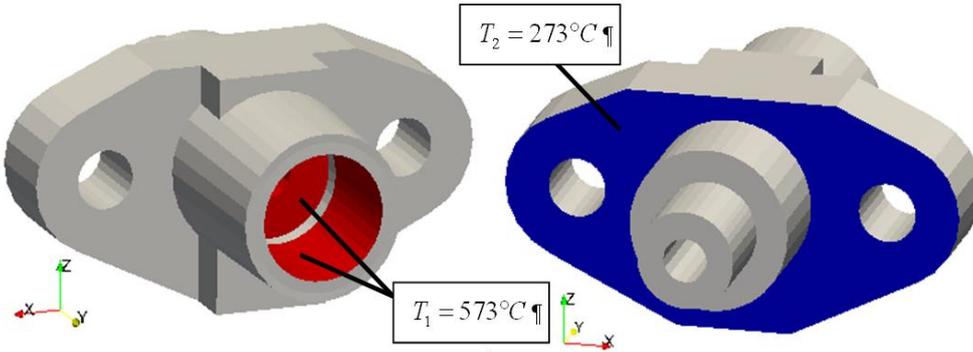


Fig. 3. Second example, geometry and boundary conditions.

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 \mathbf{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 \mathbf{b} of Eq.1, but almost no effect on discretization matrix \mathbf{A} .

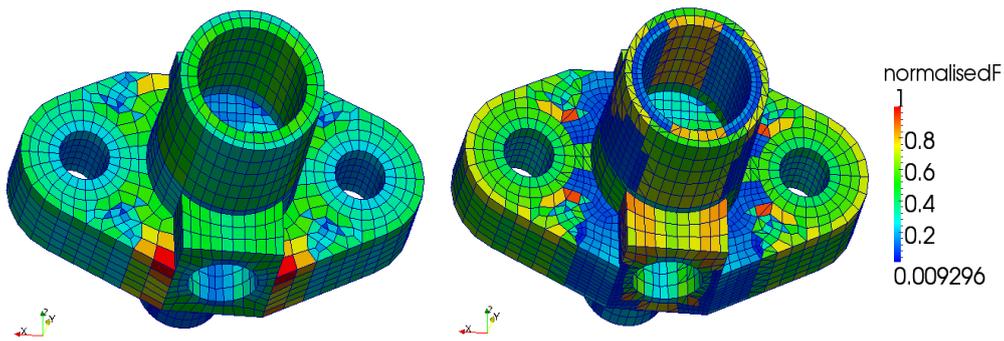


Fig. 4. AMR F detects and refines too large cells.

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

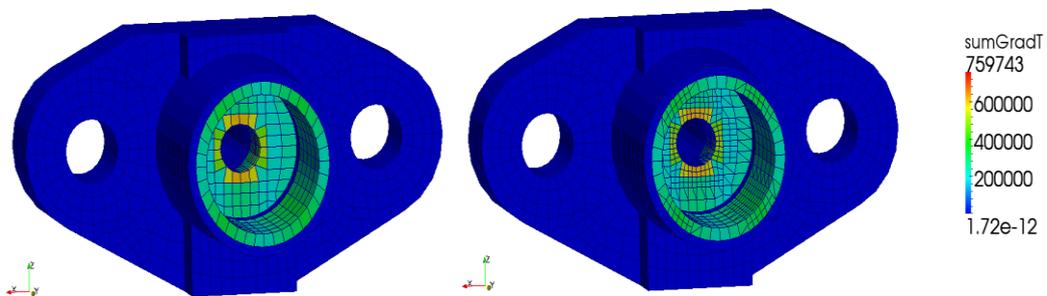


Fig. 5. AMR T takes into account the boundary conditions.

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).

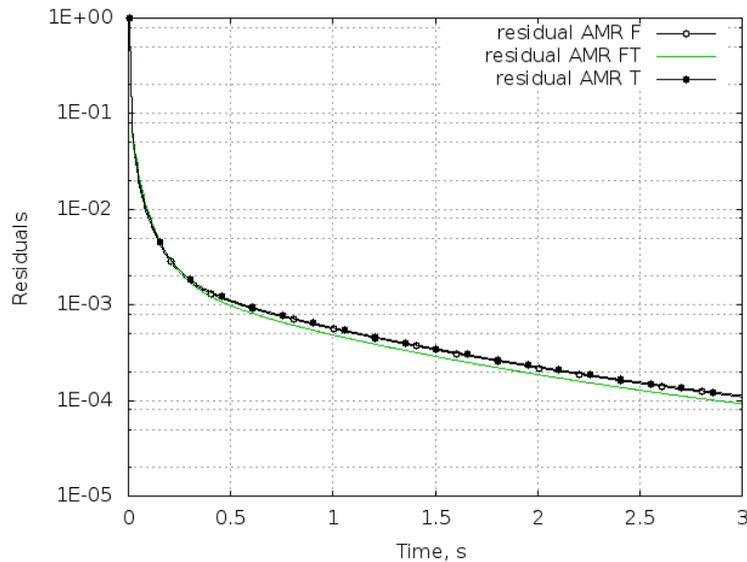


Fig. 6. Comparison of AMR F, AMR T and AMR FT.

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

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

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

References

- [1] Joubarne E., Guibault F., Braun O., Avellan F. Numerical capture of wing tip vortex improved by mesh adaptation. *International Journal for Numerical Methods in Fluids*; volume 67 (1), pp. 8-32, 2011
- [2] Escobar J. A., Ramirez S., Jimenez R.A., Giraldo A.M., Silva C., Lopez O.D., 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, New Orleans, 2012
- [3] Avdeev E.V., Fursov V.A. Discretization Matrix Condition Estimate in Finite Volume Method. *Science and Education*; volume 11, BMSTU, 2014
- [4] Voevodin V.V. *Vichislitelnie osnovi lineynoi algebr.* Moscow: Nauka, 1977
- [5] Brakken-Thal P. Gershgorin's Theorem for Estimating Eigenvalues. Retrieved from <http://buzzard.ups.edu/courses/2007spring/projects/brakkenthal-paper.pdf>
- [6] Retrieved from <https://github.com/j-avdeev/laplacianFoamF>