Final drive lubrication modeling

E. Avdeev[a,b]1, V. Ovchinnikov[b]

[a]Samara University, [b]Laduga Automotive Engineering

Abstract. In this paper we describe the method, which is the composition of finite volume method (FVM) and adaptive mesh refinement (AMR). We use it to solve car final drive inner volume oil lubrication problem. The computational algorithm is implemented using OpenFOAM parallel library that provides data structures and routines to work with the finite volume method and adaptive mesh. This library supports parallelism through OpenMPI. The paper presents the results of numerical simulation.

Keywords: lubrication modeling, adaptive mesh refinement, final drive modeling, bearing modeling;

1 Introduction

In recent years automotive industry becomes increasingly competitive and global in nature. This forces the car manufacturers to optimize components and assemblies, in order to reduce the cost of their production, but without performance reducing.

This paper considers a design of automobile final drive. Final drive design is based on the fulfillment of the technical requirements and carried out by conducting separated modeling cases: stress analysis, kinematic analysis, thermal analysis, manufacturability analysis, oil flow analysis. One of the problems arising during the final drive design is related to the oil lubricity analysis. In particular, the authors of this paper solved the problem of the oil flow simulation created by rotating gear wheel of final drive. Also Adaptive Mesh Refinement (AMR) was used to reduce mesh cells amount and consequently amount of memory and CPU time. Results obtained with AMR were compared with results obtained with uniform mesh refinement.

The calculation results are transferred to design engineer, who will update the shape of the final drive body accordingly to technical requirements. In particular, oil flow has to reach the stuffing box (see Fig.1).

2 Problem Formulation

The final drive inner volume oil lubricity problem is to simulate oil distribution under the gear wheel, shaft and bearings rotation. After the technical requirements analysis we decide to perform lubrication modeling for the following selected shaft rotational frequencies: 551, 800, 1600, 2400 rpm. This frequencies set describes final drive basic operating modes.

---

* Corresponding author. Tel.: +7-927-216-4396.
E-mail address: j-avdeev@yandex.ru

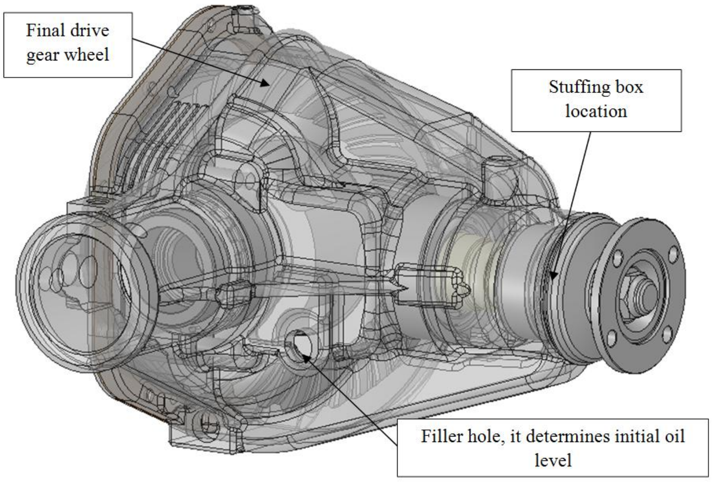

Fig. 1. The original geometry and the basic elements of the final drive.

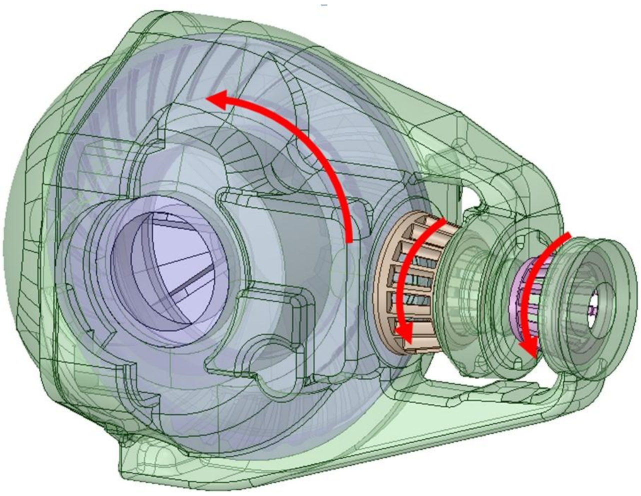

Fig. 2. The final drive internal volume, gear wheel and bearings rotation directions.

To simulate lubricity we decided to use a two-phase liquid-air model without taking into account the compressibility, heat transfer and miscibility. For phase separation we use VOF method, such as Lemfeld [1], Chunfeng [2].

We consider the mathematical model, which describes oil distribution during final drive gear wheel rotation.

Oil distribution is described by the following equations [3]:

$$\frac{\partial \alpha_\varphi \overline{U_\varphi}}{\partial t} + \nabla \cdot \left(\alpha_\varphi \overline{U_\varphi}\,\overline{U_\varphi} \right) + \nabla \cdot \left(\alpha_\varphi \overline{R}_\varphi^{eff} \right) = -\frac{\alpha_\varphi}{\rho_\varphi} \nabla \overline{p} + \alpha_\varphi g + \frac{\overline{M_\varphi}}{\rho_\varphi} \quad (1)$$

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot \left(\overline{U_\varphi} \alpha_\varphi \right) = 0 \quad (2)$$

where φ – phase, α – phase fraction, $\overline{R}_\varphi^{eff}$ is combined Reynolds (turbulent) and viscous stress, $\overline{M_\varphi}$ – averaged inter-phase momentum transfer term, $\overline{U_\varphi}$ – averaged transport velocity, p – pressure, t – time discretization step size, g – acceleration due to gravity, ρ_φ – phase density.

Combining equation (2) for two phases with $\varphi = a$ and b yields the volumetric continuity equation for the mixture, which will be utilized to formulate an implicit equation for the pressure. The volumetric continuity equation reads:

$$\nabla \cdot \overline{U} = 0 \quad (3)$$

where $\overline{U} = a_\alpha \overline{U_\alpha} + a_\alpha \overline{U_b}$.

The averaged equations representing the conservation of mass and momentum for each phase.

For more efficient use of computing resources, we decided to use an approach based on the use of adaptive mesh refinement/coarsening (Adaptive Mesh Refinement - AMR). Adaptive mesh allow to reduce computational cost, to correct mesh in complex areas, to handle moving surfaces, phase transitions and other areas of high gradients.

3 Adaptation method

There is a large amount of literature which deals with dynamic mesh and mesh adaptation methods. One of the first works on dynamic mesh application were investigations of Miller [4] and Yanenko[5]. Mesh adaptation methods can be classified according to the mesh movement strategy into two groups [6]: refinement or coarsening of mesh elements (h-adaptation) or mesh nodes moving (p-adaptation). Currently, h-adaptation is the most popular method in computational fluid dynamics (CFD) applications.

Mesh adaptation methods usually based on minimization of some selected functional - mesh quality metric field. Most popular mesh quality metrics can be divided by two groups: geometric metrics [7, 8] and mathematical metrics [9]. Geometric metrics include adjacent cells volume difference, cell faces non-orthogonality and cell faces skewness. Mathematical metrics include discretization matrix conditioning and discretization matrix maximum and minimum eigenvalues.

In practice, mathematical metrics are not used, due to the too high computational costs in comparison with geometric metrics. Geometric metrics are rather recommendations to the user.

In this work as mesh quality metric we use scalar field, which based on discretization matrix eigenvalues estimation. This method described in more detail in our previous work [10].

Adaptive mesh refinement algorithm includes following steps:

1. Discretization matrix $\mathbf{A}$ initialization.
2. Matrix $\mathbf{M} = \mathbf{I} - \mathbf{A}$ calculation.
3. Eigenvalues estimation matrix calculation.

$$\mathbf{F}_i = \left|m_{ii}\right| + \sum_{i \neq j} \left|m_{ij}\right|, \qquad (3)$$

where m_{ii} and m_{ij} diagonal and off-diagonal elements of matrix $\mathbf{M}$.

4. Mesh cells refinement/coarsening, based on matrix $\mathbf{F}$.

4 Method implementation

Mesh adaptation approach was successfully implemented by developers of commercial and non-commercial software packages such as FlowVision, Abaqus, Ansys, OpenFOAM. In this study we used an OpenFOAM open library, which has complete modules for AMR implementation.

For AMR configuration in OpenFOAM user need to provide following information:

- mesh update frequency (update mesh on every first, second or subsequent iteration);
- scalar field, whose values will be used for the mesh refinement/coarsening;
- field values interval, defined by minimum and maximum values, at which we want to refine mesh;
- field threshold value, below which we want to start mesh coarsening;
- maximum cells refinement level relative to initial mesh cells;

- the maximum allowable mesh cells amount.

AMR procedure implemented in the OpenFOAM library [11]. Wherein the original OpenFOAM library does not support AMR usage simultaneously with the rotating mesh domains. So it was necessary to conduct OpenFOAM library modifications.

Current version of OpenFOAM-v1612 does not allow to use mesh adaptation (implemented by dynamicRefineFvMesh class) and rotation of the mesh (implemented by solidBodyMotionFvMesh class) simultaneously. Therefore, to achieve the desired functionality, we have created a new C ++ class solidBodyMotion dynamicRefineFvMesh by virtual inheritance. The sources available at [12].

5 Computational results and analysis

To determine the effectiveness of the AMR approach we used comparison of residuals for AMR and for static uniform mesh cases. In particular we used pressure residual. For mesh comparison we set maximum allowable mesh cells amount for AMR case 1 100 000 cells. Uniform mesh contained 1 045 637 cells.

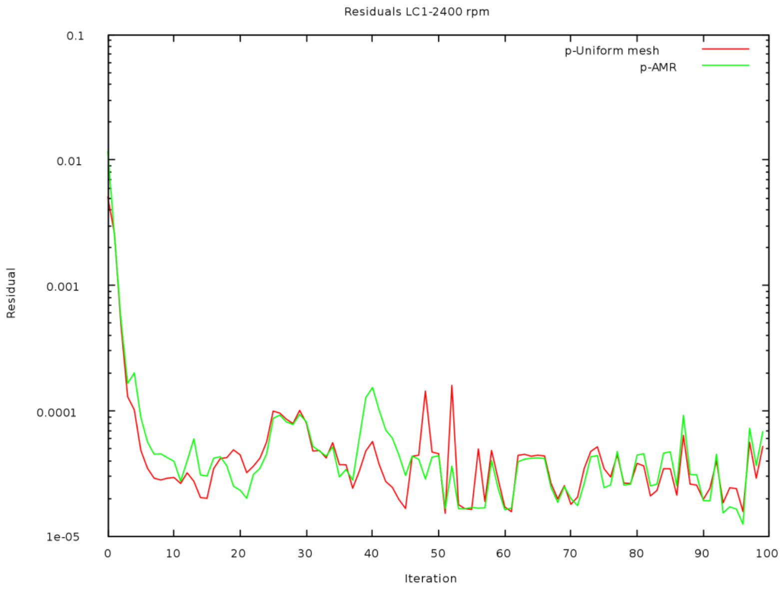

Fig. 3. p residuals comparison for uniform and AMR cases for 2400 rpm frequency.

As shown in Figure 3 and Figure 4 behavior of the residuals for high rotational frequency is the same (LC1 2400 rpm). In case of low rotational frequencies (LC4 551 rpm), the pressure residual converges faster for AMR. This can be explained by the fact that in case of low frequency the oil flow behavior less chaotic (see Figure 6) than in case of high frequencies.

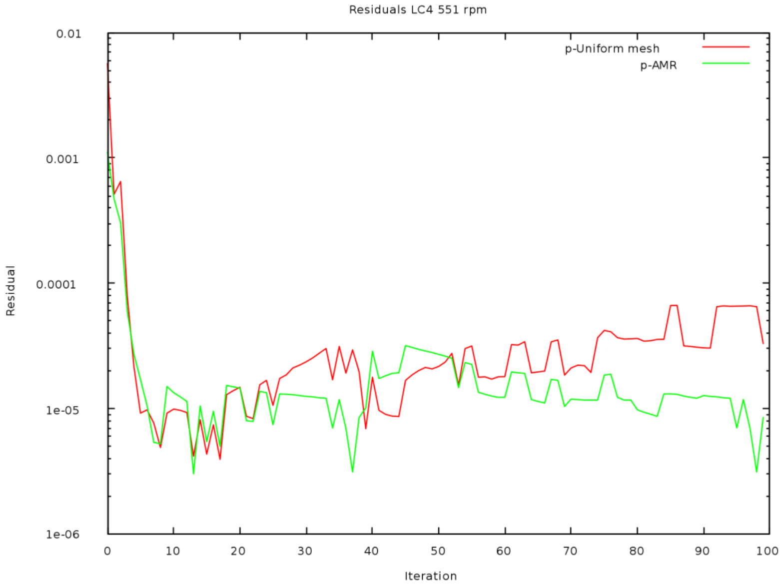

Fig. 4. p residuals comparison for uniform and AMR cases for 551 rpm frequency.

Figure 5 shows the oil-air free surface for wheel rotational frequency 2400 rpm, time = 0.278 s.

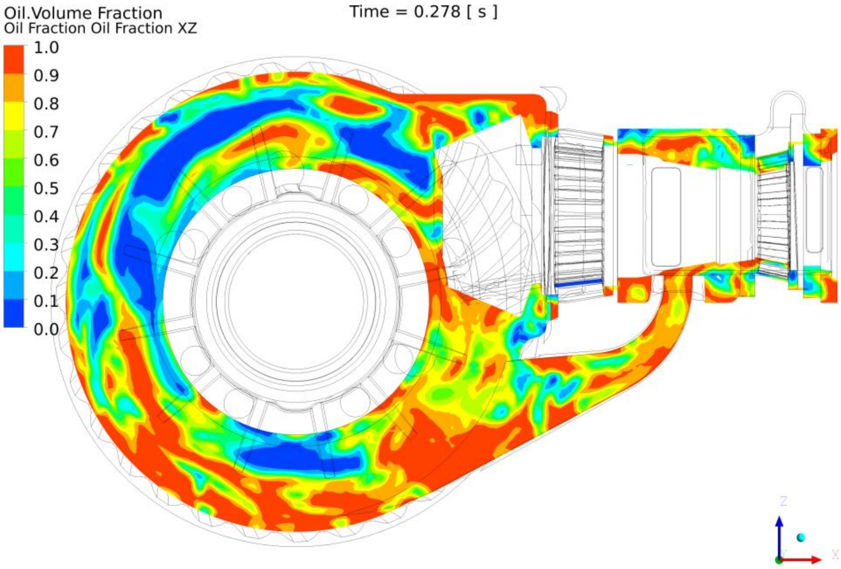

Fig. 5. Oil- distribution for wheel rotational frequency 2400 rpm, time t = 0.278 s.

Figure 6 shows the oil distribution for wheel rotational frequency 551 rpm, time t = 1.7 s. It can be seen that in this case the oil flow reaches the stuffing box location.

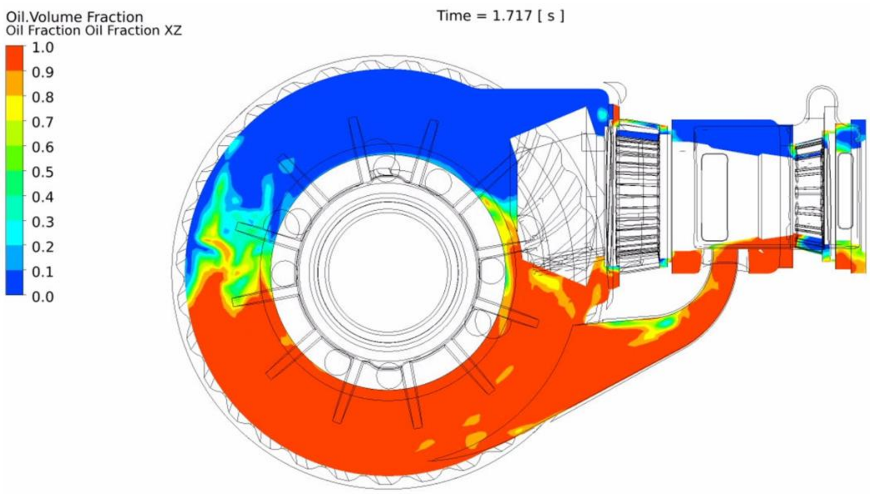

Fig. 6. Oil distribution for wheel rotational frequency 551 rpm, time t = 1.7 s.

Adaptive mesh refinement more effective in areas of constant oil flow form, less effective in areas with stochastic oil flow behavior.

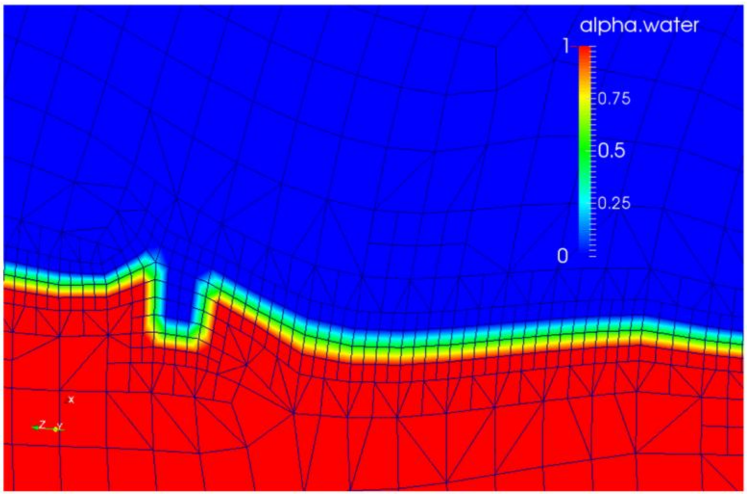

Fig. 7. Mesh fragment, wheel rotational frequency 551 rpm, time t = 1e-6 s.

Figure 7 shows mesh fragment for case of wheel rotational frequency 551 rpm, time t = 1e-6 s. More fine mesh formed in areas with a higher α phase fraction gradient, which reduces task computational cost.

6 OpenFOAM library parallelism

Algorithms parallelization performed by built-in features of OpenFOAM parallel library. The method of parallel computing used in OpenFOAM is based on the computational domain mesh and fields decomposition into separate parts, every single part is assigned to a separate computing core. Thus, the parallel calculation process includes the following steps: mesh and fields decomposition; parallel solver run; postprocessing after mesh and fields reconstruction or right in the decomposed form. OpenFOAM supports OpenMPI implementation of the standard message passing interface MPI by default, it is also possible to connect other MPI implementation libraries.

All computations are performed on cluster “Sergey Korolev”. In particular, we used two server types:

- HS22 blade servers, each of them has 2x CPU: Intel Xeon X5560, 4 cores;
- HS23 blade servers, each of them has 2x CPU: Intel Xeon E5-2665, 8 cores.

Execution time comparison for this server types showed on Figure 8.

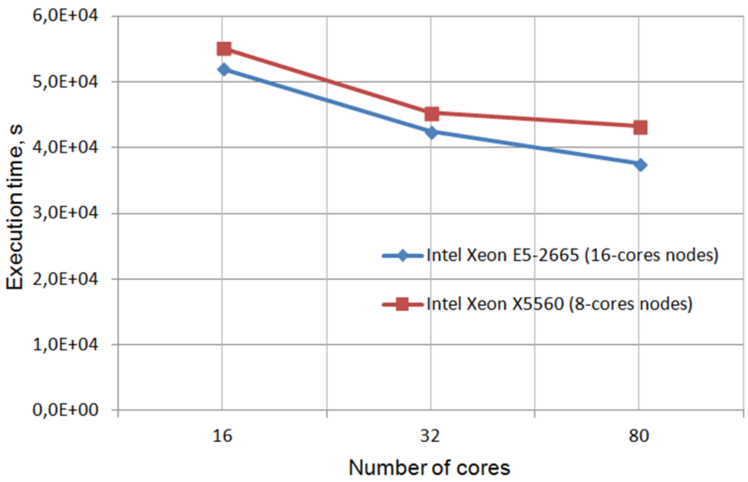

Fig. 8. Execution time for different number of cores and nodes types.

7 Conclusion

An implemented by OpenFOAM library model has shown efficiency and stability. Adaptive mesh refinement approach showed greater efficiency in cases of low rotational frequencies and less efficiency in case of high frequency. Computational complexity of the problem should be taken into account when deciding on the use of AMR. The additional costs of calculating the mesh quality metric field and mesh updating decrease efficiency of AMR using.

Acknowledgements. This work was supported by the Ministry of Education and Science of the Russian Federation.

References

1. Lemfeld, F. Numerical simulations of unsteady oil flows in the gearboxes / F. Lemfeld, K. Fran, J. Unger // Journal of applied science in the thermodynamics and fluid mechanics – 2007. – Vol.1. – P. 1-5.
2. Vande, V. Flow simulations in rotary volumetric pumps and compressors with the fictious domain method / V. Vande, J. Vierendeels, E. Dick // Journal of Computational and Applied Mathematics – 2004. – Vol.168(1-2). – P. 491-499.
3. Rusche, H., Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions [Elec-tronic resource]. – Access mode: http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/HenrikRuschePhD2002.pdf (4.02.2017).
4. Miller, K. Moving finite elements / K. Miller, R.N.Miller // Journal on Numerical Analysis – SIAM, 1981. – V. 18(6) . – P. 1019-1032.
5. Yanenko, N.N. The method of the solution of gaz dynamical problems in moving meshes / N.N. Yanenko, V.D. Lisseikin, V.M. Kovenia // Computing Methods in Applied Sciences and Engineering – 1977. – Vol. 91(2). – P. 48-61.
6. Ferziger, J.H. and Peric, M. Computational methods for fluid dynamics: Springer Verlag, Berlin-New York, 1995. doi: 10.1007/978-3-642-56026-2
7. Knupp, M.P.: “Remarks on Mesh Quality”, Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exhibit, pp. 1-10, 2007.
8. Hrvoje, J.: Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows, Phd thesis, 394 p., 1996.
9. Shewchuk, J.R.: “What is a Good Linear Element? Interpolation, Conditioning, and Quality Measures”, Proceedings of Eleventh International Meshing Roundtable, pp. 115-126, 2002.
10. Avdeev, E. An adaptive mesh refinement in the finite volume method / E. Avdeev, V. Fursov, V. Ovchinnikov // Proceedings of Information Technology and Nanotechnology (ITNT-2015), CEUR Workshop Proceedings – 2015. – Vol.1490. – P. 234-241.

11. OpenFOAM UserGuide [Electronic resource]. — Access mode: http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf (4.02.2017).
12. Final drive lubrication OpenFOAM case sources [Electronic resource]. – Access mode: https://github.com/j-avdeev/DriveLubrication (4.02.2017).