

# Modeling and Simulation of the interaction between oil and rotating gear within Final drive volume

E. Avdeev<sup>1,2</sup>, V. Ovchinnikov<sup>2</sup>

<sup>1</sup>Samara National Research University, 34 Moskovskoe Shosse, 443086, Samara, Russia

<sup>2</sup>Laduga Automotive Engineering, 71 Mozhayskoe shosse, 143000, Odintsovo, Russia

---

## Abstract

An adaptive mesh refinement (AMR) method based on discretization matrix metric is described. The computational algorithm is implemented using OpenFOAM parallel library. This open C++ library provides data structures and routines to work with the finite volume method and adaptive mesh. The method was used for oil flow in final drive simulation. 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 approach showed greater efficiency in cases of low rotational frequencies and less efficiency in case of high frequency.

*Keywords:* mesh adaptation; lubrication modeling; final drive modeling; bearing modeling

---

## 1. Introduction

In this paper we consider a design of automobile final drive. One of the problems arising during the automobile final drive design is the lubricity problem. In particular, the authors of this paper solved the problem of the oil flow simulation created by rotating gear wheel of final drive. Then, the calculation results are transferred to design engineer, who will correct the shape of the final drive body accordingly that the oil flow will reach the stuffing box (see Fig.1).

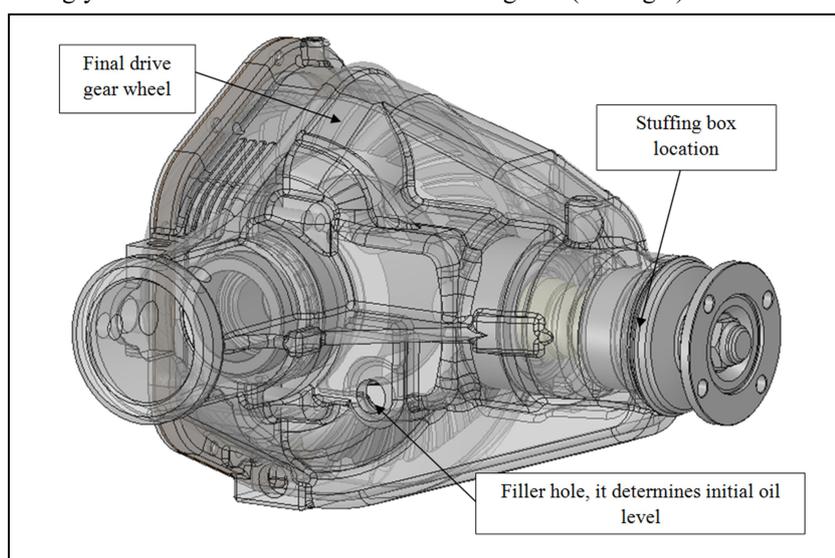


Fig. 1. The original geometry and the basic elements of the final drive.

To simulate oil flow 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]. Oil flow simulation is quite complex and requires a lot of computational time. 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). AMR procedure implemented in the OpenFOAM library [3], but the original library does not support AMR for rotating mesh. In this paper, we conducted the OpenFOAM library modification for final drive lubricity modeling.

## 2. Mesh adaptation method

Currently, the mesh adaptation technologies are widely used in numerical problems solving. 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 usually based on minimization of some selected functional. It is achieved by refinement or coarsening of mesh elements (h-adaptation) or mesh nodes moving (p-adaptation).

Mesh adaptation allows to reduce computational cost, to correct mesh in more complex areas, to handle moving surfaces, phase transitions and other areas of high gradients. Mesh adaptation approaches was successfully implemented in many commercial and non-commercial software packages such as Star-CCM+, 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.

In this work as scalar field we use field, based on discretization matrix eigenvalues estimation. This method described in more detail in [6].

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.

$$\mathbf{F}_i = |m_{ii}| + \sum_{i \neq j} |m_{ij}|,$$

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

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

Current version of OpenFOAM-v4.1 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 [7].

### 3. Discussed Problems

In this section we briefly consider the mathematical model, which describes oil distribution during final drive gear wheel rotation.

Oil distribution is described by the following equations [8]:

$$\frac{\partial \alpha_\varphi \overline{U}_\varphi}{\partial t} + \nabla \cdot (\alpha_\varphi \overline{U}_\varphi \overline{U}_\varphi) + \nabla \cdot (\alpha_\varphi \overline{R}_\varphi^{eff}) = -\frac{\alpha_\varphi}{\rho_\varphi} \nabla \overline{p} + \alpha_\varphi \mathbf{g} + \frac{\overline{M}_\varphi}{\rho_\varphi} \quad (1)$$

$$\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot (\overline{U}_\varphi \alpha_\varphi) = 0 \quad (2)$$

where  $\varphi$  – phase,  $\alpha$  – 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,  $\mathbf{g}$  – acceleration due to gravity,  $\rho_\varphi$  – 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  $\overline{U} = a_\alpha \overline{U}_\alpha + a_b \overline{U}_b$ .

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

After the technical requirements analysis we decide to perform lubrication modeling for the following selected shaft rotational frequencies: 551, 800, 1600, 2400 rpm. These frequencies set describes final drive basic operating modes.

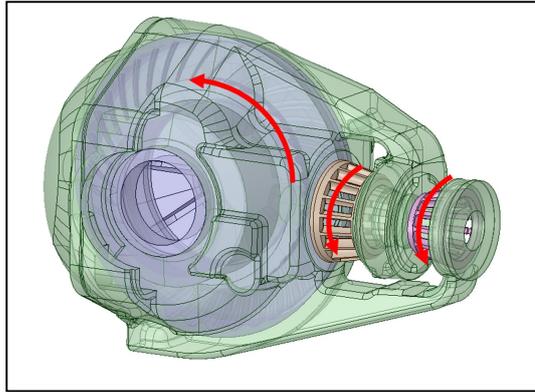


Fig. 2. The final drive internal volume, gear wheel and bearings rotation directions.

#### 4. OpenFOAM library parallelism

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. Algorithms parallelization is built-in OpenFOAM parallel library. 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. When post-processing cases that have been run in parallel the user has two options:

- reconstruction of the mesh and field data to recreate the complete domain and fields, which can be post-processed as normal;
- post-processing each segment of decomposed domain individually.

A decomposed OpenFOAM case is run in parallel using the openMPI implementation of MPI. openMPI can be run on a local multiprocessor machine very simply but when running on machines across a network, a file must be created that contains the host names of the machines. The file can be given any name and located at any path. OpenFOAM also allows to use other MPI implementation libraries.

All computations are performed on cluster “Sergey Korolev”. In particular, we used two blade 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 3.

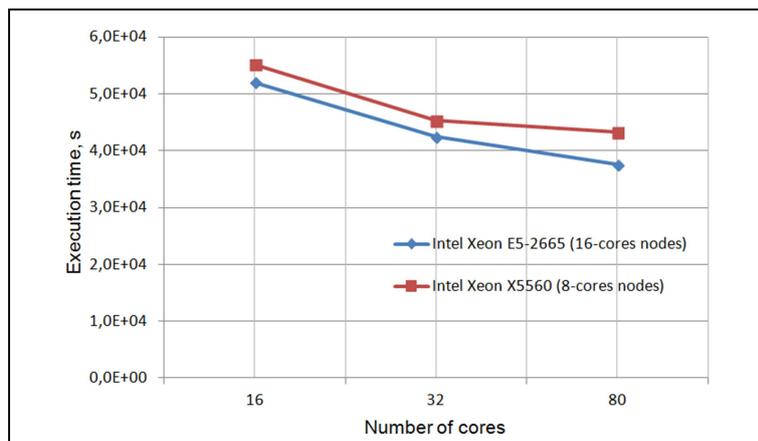


Fig. 3. Execution time for different number of cores and nodes types.

#### 5. Numerical simulation results

Figure 4 shows the oil-air free surface for wheel rotational frequency 2400 rpm, time = 0.278 s.

Figure 5 shows the oil distribution for wheel rotational frequency 551 rev/min, time  $t = 1.7$  s. It can be seen that in this case the oil flow reaches the stuffing box location.

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

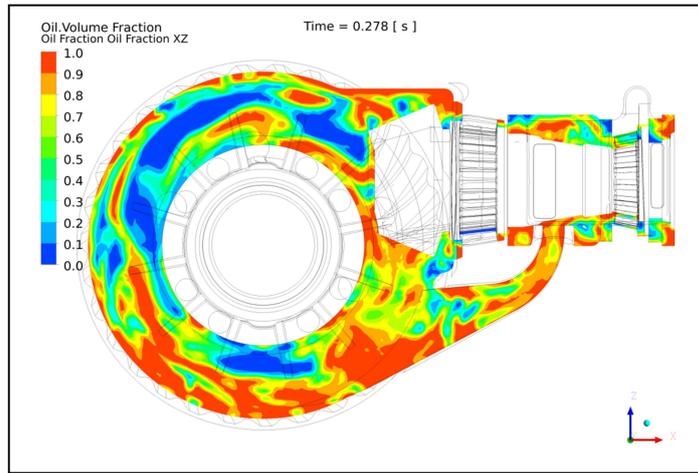


Fig. 4. Oil- distribution for wheel rotational frequency 2400 rpm, time  $t = 0.278$  s.

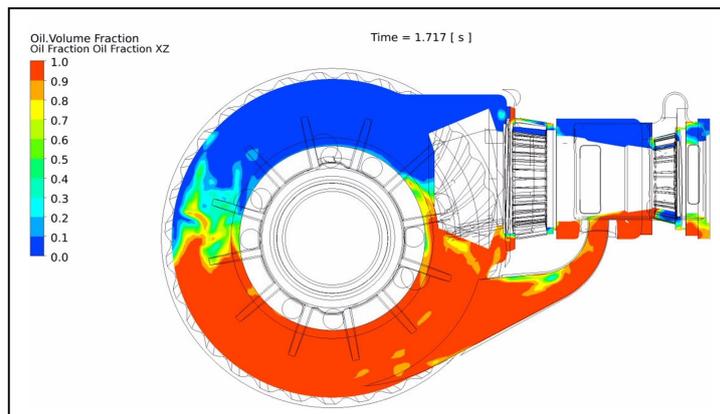


Fig. 5. Oil distribution for wheel rotational frequency 551 rev/min, time  $t = 1.7$  s.

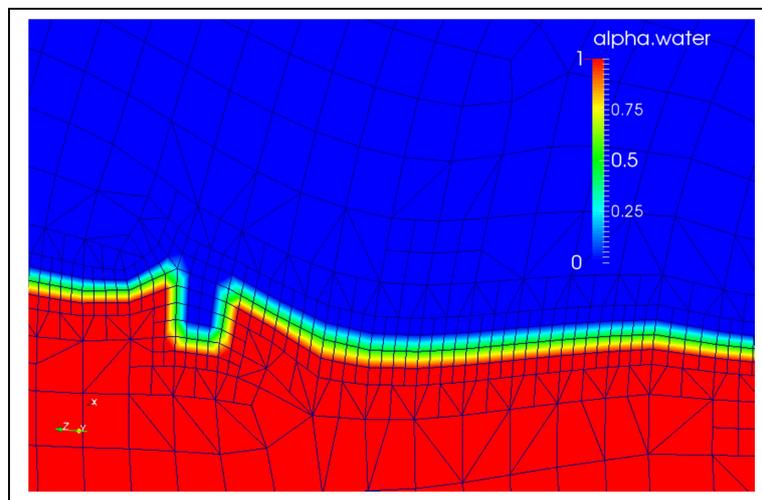


Fig. 6. Mesh fragment, wheel rotational frequency 551 rev/min, time  $t = 1e-6$  s.

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

## 6. Conclusion

Algorithms parallelization performed by built-in features of OpenFOAM parallel library. An implemented by OpenFOAM library model has shown efficiency and stability. Adaptive mesh refinement along with the ability to use parallel computing also provides computational costs reduction compared to the static mesh.

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, Fran K, Unger J. Numerical simulations of unsteady oil flows in the gearboxes. *Journal of applied science in the thermodynamics and fluid mechanics* 2007; 1: 1–5.
- [2] Vande V, Vierendeels J, Dick E. Flow simulations in rotary volumetric pumps and compressors with the fictitious domain method. *Journal of Computational and Applied Mathematics* 2004; 168(1-2): 491–499.
- [3] OpenFOAM UserGuide. URL: <http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf> (4.02.2017).
- [4] Miller K, Miller RN. Moving finite elements. *Journal on Numerical Analysis – SIAM* 1981; 18(6): 1019–1032.
- [5] Yanenko NN, Lisseikin VD, Kovenia VM. The method of the solution of gaz dynamical problems in moving meshes. *Computing Methods in Applied Sciences and Engineering* 1977; 91(2): 48–61.
- [6] Avdeev E, Fursov V, Ovchinnikov V. An adaptive mesh refinement in the finite volume method. *CEUR Workshop Proceedings* 2015; 1490: 234–241.
- [7] Final drive lubrication OpenFOAM case sources. URL: <https://github.com/j-avdeev/DriveLubrication> (4.02.2017).
- [8] Rusche H. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*. URL: <http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/HenrikRuschePhD2002.pdf> (4.02.2017).