ParCFD'2020
32^{nd} International Conference on Parallel Computational Fluid Dynamics
May-11-13 2020, Nice FRANCE

DEVELOPING DYNAMIC LOAD BALANCING MECHANISM FOR WAVE STRUCTURE INTERACTION APPLICATIONS USING OPENFOAM

Xiaohu Guo[*] AND Scott Brown[†] AND Deborah Greaves[†] AND Edward Ransley[†]

[*] Hartree Centre,
Science and Technology Facilities Council,
Daresbury Laboratory, Warrington WA4 4AD UK e-mail: xiaohu.guo@stfc.ac.uk, web page:
https://guoxiaohu.github.io/index.html

[†]School of Engineering,
University of Plymouth, Plymouth, PL4 8AA, UK

Key words: Wave Structure Interaction; Dynamic Load Balancing; parMETIS; OpenFOAM; MPI;

Abstract. In all current release of the OpenFOAM, there are no support of dynamic load balancing. However, most wave structure interaction applications track the motion of the structure using mesh deformation. The initial static domain decomposition methods implemented in the current version of OpenFOAM have no way to resolve load balancing issues that caused by the dynamic moving/deformable meshes. In this work, we have developed a dynamic load balancing library based on ParMETIS decomposition class, that considers all of the relevant parameters is required for efficient simulation and optimal parallel performance of WSI problems. Benchmarking of the new functionality shows that the load imbalance is a major performance bottleneck for dynamic mesh applications. The substantial improvements in computational efficiency (up to 5 times speed up) and parallel scaling have been observed through use of the new library for WSI applications with more than several thousands MPI tasks. As far as the authors aware, this hasn't been done before with OpenFOAM dynamic mesh applications in the literature.

1 INTRODUCTION

Offshore and coastal engineering fields are using increasingly larger and more complex numerical simulations to model wave-structure interaction (WSI) problems, in order to improve understanding of safety and cost implications. Therefore, an efficient multi-region WSI toolbox, wsiFoam[1], is being developed within an open-source community-serving numerical wave tank (NWT) facility based on the computational fluid dynamics (CFD) code OpenFOAM®, as part of the Collaborative Computational Project in Wave Structure Interaction (CCP-WSI). The CCP-WSI software framework adds a much wider range of expression-based wave theories to choose from, including the option to generate focused waves using superposition of linear components and has previously been successfully applied in a range of WSI problems such as wave impacts with fixed structures [2]; wave energy convertors [3]; floating tidal platforms [4]; and waves breaking on a beach [5]. Furthermore, `waves2Foam` provides the relaxation zone functionality which absorbs waves (and improves the inlet signal) by blending analytical and simulated values using a user-specified weighting function. All these applications involves complex geometries with adaptive/moving meshes. The simulations for high fidelity regions are very time consuming and hence there is a urgent requirement to further improve the efficiency of these high-fidelity codes based on OpenFOAM®.

Although OpenFOAM® is parallel ready, historically the MPI performance has been considered sub-optimal, but recent developments have led to a number of performance improvements being implemented in OpenFOAM® with version 5.x[7] above (OpenFOAM® v5.x+) along with new parallel I/O functionality. The developments have led to a significant performance benefit when employing a large number of processors, and it is vital that existing code is updated to be compatible with OpenFOAM® v5.x+ in order to utilise this functionality. However, OpenFOAM® v5.x+ still only offers static domain decomposition, which limits the choice of parallel load balancing methods to algorithms which can only take basic user defined arguments to aid in load balancing. These methods typically use blocking MPI communications that only consider the number of mesh cells and their spatial distribution (i.e. they assume homogeneity of load per mesh cell). As typical WSI simulations are often inhomogeneous with respect to the mesh, due to properties such as mesh motion and wave 'relaxation', these decomposition methods based purely on mesh resolution are likely to be sub-optimal in terms of computing resource usage.

In this work, we have developed a dynamic load balancing library based on `ParMETIS` decomposition class, that considers all of the relevant parameters is required for efficient simulation and optimal parallel performance of WSI problems. Benchmarking of the new functionality shows that the load imbalance is a major performance bottleneck for dynamic mesh applications but substantial improvements in computational efficiency (up to 5 times speed up) and parallel scaling have been observed through use of the new library for WSI applications with thousands of MPI tasks.

2 Results and Conclusions

The dynamic load balancing library is verified using an existing test case in the OpenFOAM® distribution for the `interDyMFoam` solver modified to use the `wsiFoam` with a single incompressible region. The simulation is a 3D dam break within a box-shaped domain ($1 \times 1 \times 1\,\mathrm{m}$), with an obstacle (a cuboid with dimensions $0.25 \times 0.25 \times 0.25\,\mathrm{m}$) at the centre of the domain. The top of the domain is considered to be an atmospheric boundary condition ($p = 1\,\mathrm{bar}$), and the remaining sides and the obstacle are considered as solid walls with no-slip conditions applied. The initial dam break is a block of water ($0.6 \times 0.1875 \times 0.75\,\mathrm{m}$) located in one corner of the domain (see Figure 1a), that is released at time $t = 0$. The initial mesh is uniformly discretised with an aspect ratio of 1 (cubic cells), and automatic mesh refinement is applied based on the values of two fields: the free surface (refined up to octree level 2); and the non-hydrostatic pressure (refinement up to octree level 3). This ensured that the mesh is only refined when it is required, both temporally and spatially, and consequently the total number of cells is constantly changing.

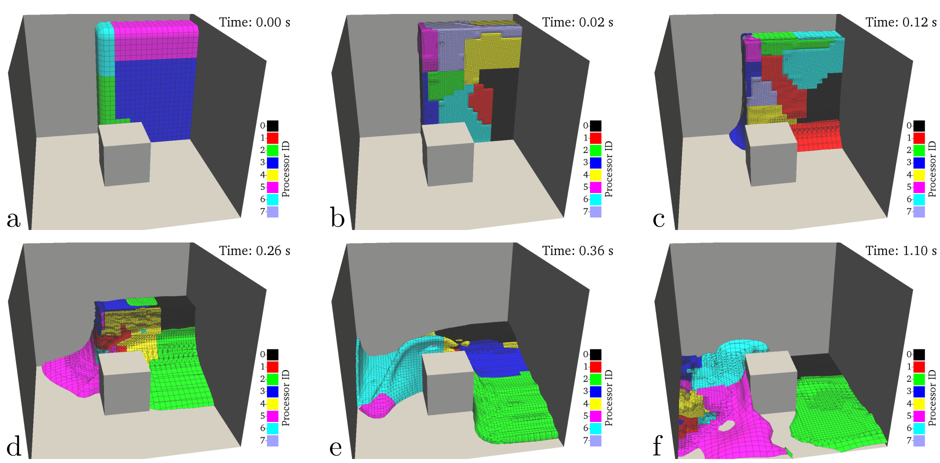

Figure 1: Snapshots of the volume of fluid isosurface ($\alpha_1 = 0.5$) from the dam break with obstacle case using eight processors, adaptive mesh and dynamic load balancing. The colours represent the sections being solved by each processor, and the grey lines show the mesh discretisation.

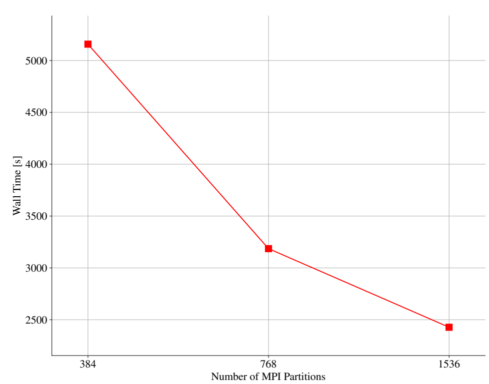

Figure 2: Scaling of the 'dam break with obstacle' test case using `wsiFoam` solver with 17M mesh cells

A new dynamic load balancing library has been developed using the number of mesh cells per MPI partition as the weight function to balance the load. This approach lends itself to simulations where the number of mesh cells is constantly changing, and is therefore highly suited to adaptive mesh refinement cases, which deploy refinement/unrefinement techniques. A test case using this adpative mesh functionality was used to benchmark the new library and shows a substantial increase in both the wall time (4-5 times speed up) and the scalibility (greater than one thousand cores with more than 50% efficiency) of the `wsiFoam` solver. However, restricting use of the dynamic load balancing library to cases with adaptive mesh refinement is non-generic and rules out a large number of current WSI problems. Typically, WSI simulations of floating objects in OpenFOAM® use mesh deformation to track the motion of the structure. Since, as standard, OpenFOAM® is only capable of utilising one form of dynamic mesh at a time, these cases are incompatible with adaptive mesh refinement, although recently progress has been made in this area [8]. To utilise dynamic load balancing in generic WSI simulations it is imperative that the weighting function is developed to work with cases with fixed numbers of mesh cells in the future, based on a timing-based technique.

Acknowledgement

The authors would also like to acknowledge the funding support under the embedded CSE programme of the ARCHER UK National Supercomputing Service (http://www.archer.ac.uk). The authors would also like to thank the EPCC eCSE support team for their help throughout this work

REFERENCES

[1] P. Martínez Ferrer, D. Causon, L. Qian, C. Mingham, and Z. Ma, "A multi-region coupling scheme for compressible and incompressible flow solvers for two-phase flow in a numerical wave tank." *Computers and Fluids*, vol. 125, pp. 116–129, 2016.

[2] S. Brown, P.-H. Musiedlak, E. Ransley, and D. Greaves, "Numerical simulation of focused wave interactions with a fixed FPSO using OpenFOAM 4.1," in *Proceedings of the 28th International Ocean and Polar Engineering Conference*, Sapporo, Japan, 2018, pp. 1–8.

[3] E. Ransley, D. Greaves, A. Raby, D. Simmonds, M. Jakobsen, and M. Kramer, "RANS-VOF modelling of the wavestar point absorber," *Renewable Energy*, vol. 109, pp. 49–65, 2017.

[4] E. Ransley, S. Brown, D. Greaves, S. Hindley, P. Weston, E. Guerrini, and R. Starzmann, "Coupled RANS-VOF modelling of floating tidal stream concepts." in *Proceedings of the 4th Marine Energy Technology Symposium (METS)*, Washington, D.C., USA, 2016, p. 5.

[5] S. Brown, D. Greaves, V. Magar, and D. Conley, "Evaluation of turbulence closure models under spilling and plunging breakers in the surf zone," *Coast. Eng.*, vol. 114, pp. 177–193, 2016.

[6] H. Weller, C. Greenshields, W. Bainbridge, M. Janssens, and B. Santos, "OpenFOAM 5.0," 2017. [Online]. Available: https://openfoam.org/release/5-0/

[7] C. Eskilsson and J. Palm, "Simulations of floating wave energy devices using adaptive mesh refinement," in *Advances in Renewable Energies Offshore*, G. Soares, Ed. Taylor and Francis Group, 2018, pp. 431–438.