Comprehensive Computational Performance Analysis of a Maritime CFD Code

Guilherme Vaz[1,2,3], Sébastien Lemaire[3,4], and Menno Deij-van Rijswijk[4]

[1]HPC&Data-Science Team, WavEC-Offshore Renewables, Portugal
[2]ISMT, University of Duisburg-Essen, Germany
[3]Dep. of Civil, Maritime and Env. Engineering, University of Southampton, UK
[4]MARIN, Maritime Research Institute of the Netherlands, the Netherlands

November 12, 2020

Abstract

Most offshore ship operations, ocean platforms, offshore wind turbines (fixed or floating), current turbines and wave energy converters, are designed and analysed via classical potential flow or empirical tools. This is due to gained experience and trust on these type of tools, the generally lower influence of viscosity in wave-related phenomena in design conditions, but especially, due to the low computational cost. Nowadays, however, with the democratisation of viscous flow tools (CFD, or Computational Fluid Dynamics), and HPC (High-Performance Computing) hardware, all fields of the maritime world are seeing an increased use of these higher fidelity tools. Moreover, computational times are consistently decreasing due to better hardware (multiple cores, faster clock rates, memory and IO), new processor architectures, usage of accelerators (e.g. GPUs), parallelisation techniques and, last but not the least, improved and hardware-specific algorithms.

Usual maritime problems involve multi-physics (fluid and structure), moving structures, moorings, soil interaction, control systems, hydrodynamics and aerodynamics, unsteady calculations, multi-species/phases, turbulence modelling, turbulence simulation and complex geometries. This implies several equations need to be solved, large grids are used and this leads to considerable computational times. Moreover, in order to increase the confidence in CFD calculations, modern Verification & Validation (V&V) techniques have to be used, which implies several of these calculations in order to identify and estimate numerical uncertainties. HPC resources are of paramount importance.

In addition, modern CFD tools cannot be seen independent of the hardware to be used. Specifics of the tool's intrinsic models, discretisation techniques employed, external libraries used (e.g. linear solvers), memory and CPU needs, IO operations, communication paradigms, require optimal hardware configurations. For instance, a tool tailored for a CPU-based HPC cluster cannot be used, without modifications, in a GPU-based HPC machine, without a strong performance penalty; or a tool for shared-memory system cannot/should not be used in a distributed memory infrastructure.

Within this context, in the current work the viscous flow multi-phase community-based open source CFD code ReFRESCO [1] is considered. ReFRESCO resembles in several regards a general commercial CFD code, optimized, verified and validated exclusively for Maritime applications, Fig. (1).

ReFRESCO MPI+distributed-memory version will be analysed in terms of software and hardware computational performance on four different European CPU-mainly HPC clusters:

- BOB: MACC, Braga, Portugal.

  12,800 cores, dual Intel SandyBridge 2.7Ghz, 26TB RAM, 1.5 PB disk, 2019 (former part of STAMPEDE-1 of Texas Advanced Computational Center TACC, 2013).

- MagnitUDE: University of Duisburg-Essen, Germany.

  14,976 cores, dual Intel Broadwell 2.1Ghz, 56TB RAM, 480TB disk, 2017.

- Iridis5: University of Southampton, UK.

  20,000 cores, dual Intel Skylake 2.0GHz, 2.2 PB disk, 2017.

- Marclus4: MARIN, Wageningen, The Netherlands.

  4,080 cores, dual Intel Xeon CPU E5-2660 v3 2.60GHz, 16TB RAM, 0.5 PB disk, 2014.

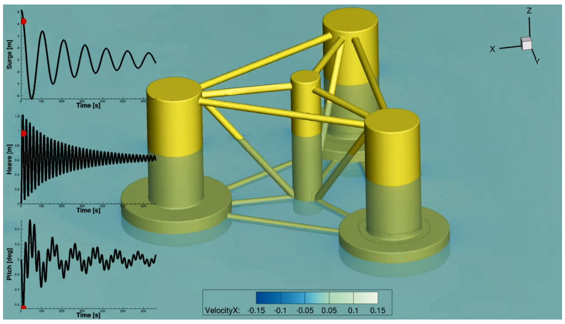

Figure 1: Unsteady non-linear RANS calculation of a Floating Offshore Wind Turbine Platform. Calculation computational time was approximately 3 weeks using 200 cores of MagnitUDE HPC cluster. For more details see [2]. Click here for animation.

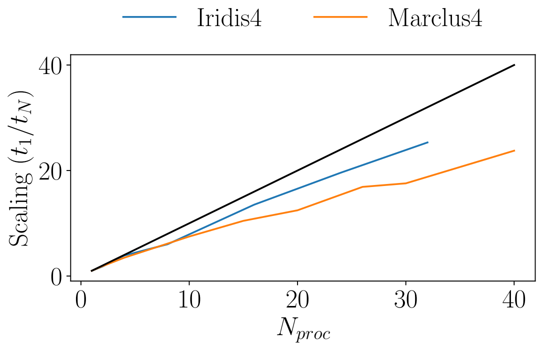

Figure 2: Scalability of a ReFRESCO CFD code solution on two different clusters using a small mesh (300k cells). N_{proc} is the number of MPI processes (or cores). The black line shows ideal scalability.

The existing performance test cases will be employed [3] and absolute CPU times, speed-up, communication time, memory needs, FLOPS bottlenecks will be studied. Strong scaling exercises will be performed [4], also illustrating the impact of some of the specific CFD code grid-handling features such as deforming, sliding and overset grids [5] on the overall computational performance. Fig. (2) presents some preliminary results for two clusters. In order to illustrate the effect of compilers, and the future capability for running in different CPU architectures (Intel, AMD, ARM), available in some of the above stated clusters, both Intel and GNU compilers are used, and their effect on the code and cluster combined performance analysed.

Keywords: CFD, HPC, Performance, Speed-Up, Strong scaling, MPI, Intel, GNU, Maritime Problems

References

[1] G. Vaz, F. Jaouen, and M. Hoekstra. Free-Surface Viscous Flow Computations. Validation oF URANS Code ReFRESCO. In *OMAE2009, Honolulu, Hawaii, USA*, Honolulu, Hawaii, USA, 2009.

[2] S. Burmester, G. Vaz, and O. El-Moctar. Towards Credible CFD Simulations for Floating Offshore Wind Turbines. *Ocean Engineering*, 209, 2020.

[3] G. Vaz, C. Klaij, and D. Rijpkema. On the Quality Control Of CFD code ReFRESCO using V&V. In *V&V2016, Las Vegas, USA*, Las Vegas, USA, May 2016.

[4] J. Hawkes, G. Vaz, A.B. Phillips, S. J. Cox, and S. R. Turnock. On the Strong Scalability of Maritime CFD. *Journal of Marine Science and Technology*, 2017.

[5] S. Lemaire, G. Vaz, and S. Turnock. On the Need for Higher Order Interpolation with Overset Grid Methods. In *22th Numerical Towing Tank Symposium (NuTTS)*, Tomar, Portugal, 2019.