

Benchmarking Delft3D FM on HPC systems for real-life problems in surface water

Menno Genseberger¹

Deltares, P.O. Box 177, 2600 MH Delft, The Netherlands,
Menno.Genseberger@deltares.nl

Abstract. This paper reports recent benchmarks on current hardware of the simulation software Delft3D FM for surface water. Benchmarks were performed before to have more efficient and scalable high performance computing (HPC). As both hardware, simulation software, and models further developed, new benchmarks were required to get an actual status. Also background is given of the test cases from real-life problems. Benchmark results are followed by a global analysis. This will be a basis to further improve Delft3D FM.

Keywords: shallow-water equations · high performance computing · real-life applications · modelling and simulation · scientific computing · complex systems.

1 Introduction

Forecasting of flooding, morphology and water quality in coastal and estuarine areas, rivers, and lakes is of great importance for society. To tackle this, the Delft3D Flexible Mesh Suite (Delft3D FM) [1, 2] has been developed. Delft3D FM is used worldwide. There is need to have Delft3D FM more efficient and scalable for high performance computing (HPC). For that purpose, new benchmarks were performed to get an actual status of the parallel performance on HPC systems for current typical real-life applications. Outcomes of benchmarks are of practical use for actual applications to see what can be expected for the available functionality of simulation software and/or hardware being used. They may also lead to required and/or possible model adaptations, given the current Delft3D FM simulation software and/or hardware on a short time scale. And, on a longer time scale, analysis of benchmark results serve as a basis for further development and/or adaptation of simulation software, given the trends in applications and/or hardware.

This paper is organized as follows. §2 gives an overview of previous steps undertaken on HPC for Delft3D FM. Then, §3 describes the current numerical methods of Delft3D FM, integral part of this is the parallelization approach. The background of the real-life applications from which test cases are selected is given in §4. Results of the new benchmarks and a global analysis are reported in §5. §6 ends with concluding remarks and outlook for a next stage.

2 Delft3D FM and HPC

The scope of Delft3D FM is broad, both in application range and the variety of hardware being used: from laptops, local clusters to remote and large scale computational resources. Therefore, optimization of this multi-purpose simulation software for HPC on specific hardware is not practical. With this in mind, previously, two European projects worked on making Delft3D FM more efficient and scalable for HPC. The first project considered two-dimensional model applications [3]. The second project focussed on three-dimensional model applications. In both projects, the basis was formed by benchmarks for test cases from real-life applications. The corresponding Delft3D FM models are further developed and applied in various running projects. Also hardware changed in the mean time. Therefore, there is a need for new benchmarks to get an actual status of the parallel performance with the most recent Delft3D FM software versions.

3 Numerical background of Delft3D FM

Delft3D FM solves the shallow-water equations with the spatial discretisation being achieved by a staggered finite volume method on an unstructured orthogonal mesh of cells of varying complexity. The discretised shallow water equations for the water levels are solved implicitly in time, with momentum advection treated explicitly. The velocities and fluxes are then obtained by back substitution, a procedure comparable with a pressure correction approach. After linearisation of the temporal discretisation, the resulting systems are solved with a semi-implicit method. This involves a linear system which is currently solved by a minimum degree algorithm to reduce system size and a Conjugate Gradient iterative solver with block Jacobi preconditioner and ILU(0) factorization on the blocks as implemented in PETSc [4]. For parallelization the domain is decomposed/partitioned via METIS [5] with halo regions on a minimal overlapping part of adjacent subdomains. These halo region are necessary for a proper representation of discretized stencils at the interfaces between subdomains. Fig. 4 in [6] gives an illustration of such a decomposition for a triangular grid on Lake Marken, this grid has more resolution where needed and less where not needed, see Fig. 1 in [6]. Due to the compulsory use of relative small time steps (a stability requirement resulting from the explicit elements in the applied time integration scheme) and $\frac{1}{\Delta t^2}$ on the diagonal of the linear system, a simple Dirichlet-Dirichlet coupling already gives fast convergence; no need to consider a more elaborate domain decomposition coupling like Robin-Robin. On the other hand, simulations typically require many time steps. Implementation of the communication between the subdomains is via MPI [7].

4 Test cases of real-life applications

The test cases are selected from current typical real-life applications, the corresponding locations are shown in Fig. 1. They have been used before in bench-

Table 1. Order of magnitude of active elements in each testcase.

Delft3D FM model	active elements in millions
Lake Marken	< 0.5
Lake IJssel	> 11.5
River Waal	9
RMM	> 6.5
North Sea	> 12.5

marks of two European PRACE projects (see §2). Table 1 indicates the number of active elements per case. The test cases involve different processes:

- Lake Marken and River Waal: physical processes for depth averaged shallow water flow,
- Lake IJssel and RMM: physical processes for three-dimensional shallow water flow and transport of substances, salinity, and temperature,
- North Sea: physical processes for three-dimensional shallow water flow and transport of substances, salinity, temperature, and chemical and biological processes for sediments, nutrients, and primary/secondary production.

Some background of each test case will be given now.

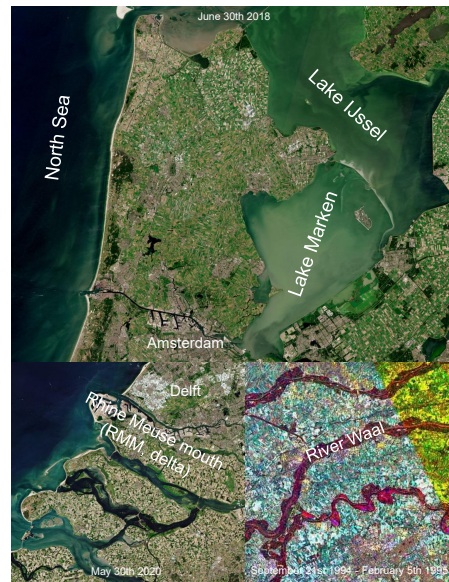


Fig. 1. Locations of the real-life applications, with top and lower left images from Copernicus Sentinel-2, ESA [8] and lower right image from ERS-1, ESA [9].

4.1 Lake Marken

For Lake Marken, a depth averaged Delft3D FM model is used as a testcase [6]. The simulation period contains a historical storm for which mainly wind driven flow of water leads to high water levels near the dikes around the lake. The computational grid is of intermediate size with 335.141 elements. Reported results for this test case are obtained by dividing the wall clock for the full simulation period of four days by a factor four. Almost the same model was used before in [3].

4.2 Lake IJssel

For Lake IJssel, a three-dimensional Delft3D FM model that simulates water flow, salinity, and temperature is further developed in [10]. Here, flow of water is both driven by wind and in- and outgoing sources of water. For application to salt intrusion in Lake IJssel, the horizontal computational grid was locally adapted near a dike in the north, where more saline water enters the lake via the locks and sluices from the Wadden Sea. The resulting horizontal computational grid is relatively large with 426.343 elements. In the vertical direction so-called Z-sigma layers (upper boundary fitted) are used with 27 sigma layers on top and $\Delta Z = 0.125$ m for the Z layers below. Typical simulation periods for impact assessments of salt intrusion or other water quality studies are one year. Here, for benchmarking the simulation period is shortened to one day. Reported results for this test case are the wall clock times for this simulation period of one day.

4.3 River Waal

The image on the lower right in Fig. 1 shows the flooding of three major rivers in the Netherlands in February 1995, see [9]. For River Waal, a schematic depth averaged model was developed to study the effect of lowering the groynes on the design flood level [11]. The computational grid contains 9.000.000 rectangular elements. Due to the simple shape and relative large size, it forms a good starting point to study parallel performance. Reported results for this test case are the wall clock times for a simulation period of three hours. The same model was used before in [3].

4.4 RMM (Rhine Meuse mouth)

Management of the Rhine Meuse mouth area (RMM) requires accurate prediction of salt intrusion and flow velocities. For these aspects the three-dimensional Delft3D FM model of RMM that simulates water flow, salinity, and temperature, is further developed in [12]. The resulting horizontal computational grid is large with 791.706 elements. In the vertical direction Z-sigma layers are used with 8 sigma layers on top and $\Delta Z = 1.0$ m for the Z layers below. Typical simulation periods for impact assessments of salt intrusion are one year. Here, for benchmarking the simulation period is shortened to one day. Reported results for this test case are the wall clock times for this simulation period of one day.

4.5 North Sea

For the North Sea focus is on integrated assessments of multiple impacts such as climate change, large-scale coastal and off-shore infrastructure developments, and development of large-scale aquaculture. For these purposes, a Delft3D FM model of the North Sea is further developed in [13]. The model consists of water flow, temperature, salinity, sediments, nutrients, primary production/algae bloom, and secondary production (predominantly shellfish species, such as mussels, razor clams etc.). It is a large three-dimensional Delft3D FM model with additional transport equations and processes for water quality and ecology. The horizontal computational grid consists of 618.919 elements. In the vertical direction Z-sigma layers are used with 20 sigma layers on top and $\Delta Z = 5.0$ m for the Z layers below. Here, for benchmarking the simulation period is shortened to one day. Reported results for this test case are the wall clock times for this simulation period of one day.

5 Benchmarks

5.1 Approach

As optimization of Delft3D FM on specific hardware is not practical (§2), benchmarks are run at typical modern hardware. Furthermore, due to the broad range of processes that can be simulated, computational complexity may vary. This makes it less easy to zoom in at specific aspects at first instance. Therefore, an approach with different stages is followed. Here, results are presented for a first stage. Benchmarks are for typical real-life applications that vary in size and type of processes (§4). Based on the differences of these properties for the testcases, results are analysed globally. The outcome of this analysis is used to decide which aspects should be investigated in more detail in a next stage.

5.2 Hardware

Benchmarks with the test cases were run at

- Cartesius (BullX Intel Xeon cluster, SURF The Netherlands): nodes with each 24 cores in 2 Intel Xeon E5-2690 v3 processors and 32 cores in 2 Intel Xeon E5-2697A v4 processors, InfiniBand, for Delft3D FM version 2021.04,
- Snellius (Lenovo AMD cluster, SURF, The Netherlands): nodes with each 128 cores in 2 AMD Rome 7H12 processors and nodes with each 192 cores in 2 AMD Genoa 9654 processors, InfiniBand, for Delft3D FM Apptainer container versions 2024.03 and 2023.01 (North Sea),
- DelftBlue (Fujitsu Intel Xeon cluster, Delft University, The Netherlands): nodes with each 48 cores in 2 Intel Xeon Gold 6248R processors and 64 cores in 2 Intel Xeon Gold 6448Y processors, InfiniBand, for Delft3D FM Apptainer container versions 2025.02 and 2023.01 (North Sea),
- HAL8 (Dell Intel Xeon cluster, Deltares, The Netherlands): virtual nodes with each 60 virtual cores in 2 Intel Xeon Platinum 8462Y+ processors, Gigabit Ethernet, for Delft3D FM versions 2024.03 and 2.27.22 (North Sea).

5.3 Setup

Although the simulation periods of the original models were shortened, the resulting test cases still contain a representative balance of computational work, communications, and I/O that can be expected for application in real-life. For each test case we focused on the time loop, the initialisation and finalisation phases are excluded from this. Strong scaling of the test cases is considered by increasing the number of computational cores. Timing of the wall-clock time has been performed one time for every configuration. Shown are double logarithmic plots for wall-clock time as a function of the number of computational cores. In case of linear parallel scaling, lines will have a downward slope of 45° .

5.4 Results and global analysis

The results of the benchmarks are shown in Fig. 2 (model of Lake Marken and Lake IJssel), Fig. 3 (schematic model of River Waal), and Fig. 4 (model of RMM and North Sea), respectively.

Fig. 2 shows almost linear parallel scaling up to 32 cores for Lake Marken. For more than 64 cores scaling breaks up, this may be due to the relative small size (Table 1). Similar to Lake Marken, the schematic model of River Waal only simulates the physical processes for depth averaged shallow water flow. There, almost linear parallel scaling holds up to 512 cores, see Fig. 3¹. This number is a factor 16 times higher than the 64 cores for Lake Marken. The models for RMM and Lake Marken differ with almost the same factor in the number of active elements (see Table 1).

Compared to Lake Marken, Lake IJssel has a bit larger horizontal computational grid, uses multiple layers in the vertical direction, and involves more physical processes for salinity and temperature. Fig. 2 shows that parallel scaling holds for larger core counts than Lake Marken, like the schematic model of River Waal. But now the slope is less steep which may be due to the additional algorithms for the other physical processes. Results of the RMM model in Fig. 4 are comparable to Lake IJssel, both models simulate the same physical processes (and the required memory per active element is the same). The North Sea model contains additional chemical and biological processes, with more advection-diffusion-reaction equations and file I/O². Fig. 4 shows that parallel scaling holds up to 512 cores, the slope seems to be a little bit more flattened compared to RMM and Lake IJssel.

For the AMD based Snellius, using all available cores within one node does not yield the fastest performance for Lake IJssel, RMM, and North Sea. Using 64 instead of all 192 cores per Genoa 9654 node makes computations 1.6 times faster for RMM, for Lake IJssel and North Sea this factor is about 2.6. This may be due to the memory used and RMM has less active elements than Lake IJssel and North Sea (see Table 1).

¹ The outlier for 2048 cores on Snellius Rome 7H12 nodes needs further investigation.

² On Delft Blue it was quite hard to start this model, this may be due to the many and large input files and needs to be further investigated.

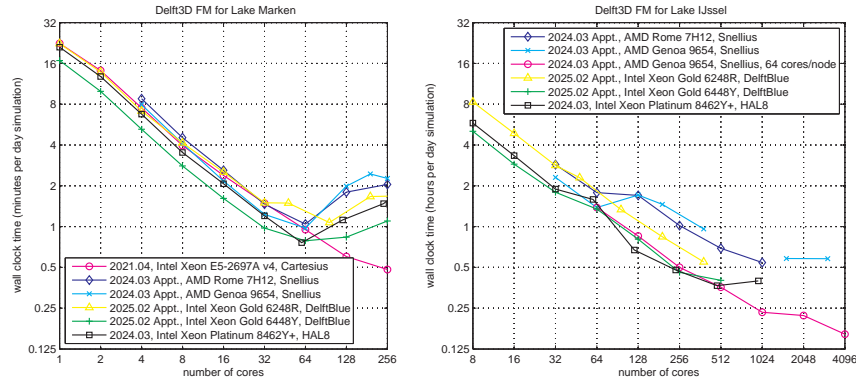


Fig. 2. Benchmark results of Delft3D FM model for Lake Marken (left) and Lake IJssel (right).

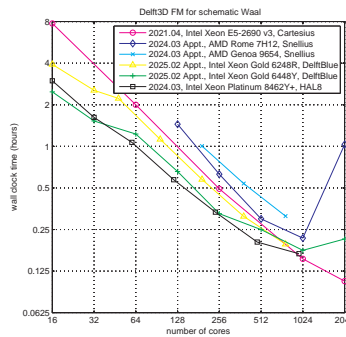


Fig. 3. Benchmark results for schematic Delft3D FM model of River Waal.

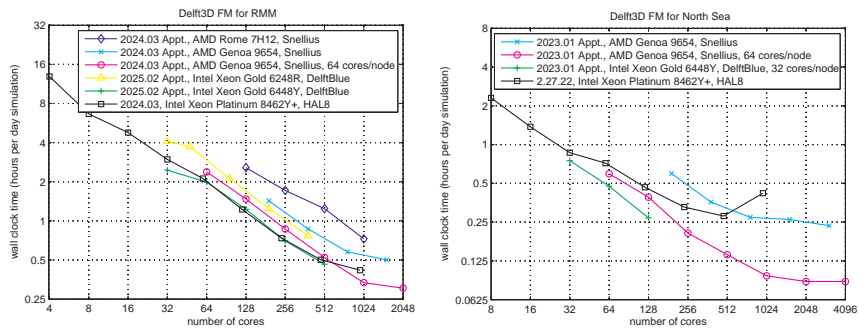


Fig. 4. Benchmark results of Delft3D FM model for RMM (left) and North Sea (right).

6 Concluding remarks and next stage

For the simulation software Delft3D FM new benchmarks were performed to get an actual status of parallel speed and efficiency. Optimization of Delft3D FM for HPC on specific hardware is not practical. Therefore, representative test cases were selected for the broad range of real-life applications and benchmarks were performed on typical modern hardware. Results were reported and followed by a global analysis. For small and large Delft3D FM models with only elementary parts for shallow water flow almost linear parallel scaling is observed. By adding more processes to a Delft3D FM model the slope of the parallel scaling becomes less steep. Some results indicate that memory usage affects computational times. In a specific case it was hard to start one of the models, this may be due to the many and large input files. Due to the followed approach, the results are a balanced outcome for the broad application range (without too detailed investigations and/or specific optimizations). In the next stage, this should be further investigated with performance analysis tools with focus on load imbalance, communication overhead, memory bandwidth, and constraints in file I/O.

Acknowledgments. This paper contains parts from projects financed by the Dutch Ministry of Infrastructure and the Environment. PRACE, SURF, and Delft University are acknowledged for access and support to Cartesius, Snellius, and DelftBlue.

References

1. Delft3D Flexible Mesh Suite. <https://www.deltares.nl/en/software-and-data/products/delft3d-flexible-mesh-suite>
2. Kernkamp, H. W. J. et al.: Efficient scheme for the shallow water equations on unstructured grids with application to the Continental Shelf. In: *Ocean Dynamics* **61**(8), 1175–1188 (2011)
3. Mogé, M. et al.: Scalable Delft3D Flexible Mesh for Efficient Modelling of Shallow Water and Transport Processes. PRACE White Paper 284 (2019)
4. <https://www.mcs.anl.gov/petsc>
5. Karypis G.: METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices. University of Minnesota (1998)
6. Genseberger M. et al.: Domain Decomposition in Shallow Water Modelling of Dutch Lakes for Multiple Applications. In: *Proc. 26th Int. Conf. on Domain Decomposition*. Springer (2022)
7. Gropp, W. et al.: MPI: The Complete Reference Vol. 2. MIT Press (1998)
8. Copernicus Sentinel-2. ESA. <https://sentinels.copernicus.eu/copernicus/sentinel-2>
9. ERS-1, ESA. <https://earth.esa.int/eogateway/gallery/flood-in-netherlands-1995>
10. Verbruggen, W. et al.: Ontwikkeling zesde-generatie 3D IJsselmeer model. Report 11210333-007-ZWS-0002 (in Dutch), Deltares (2024)
11. Yossef, M.F.M., Zagonjoli, M.: Modelling the hydraulic effect of lowering the groynes on design flood level. Report 1002524-000-ZWS-0009, Deltares (2010)
12. van der Kaaij, T. et al.: RMM 3D, een nieuw 3D model van de RijnMaasMonding in D-HYDRO. Report 11208053-005-ZWS-0002 (in Dutch), Deltares (2022)
13. van Duren, L.A. et al.: Ecosystem effects of large upscaling of offshore wind on the North Sea. Report 11203731-004-ZKS-0010, Deltares (2021)