Adaptive grid refinement for two-phase flow applications

Peter van der Plas[*], Arthur E. P. Veldman[*], Joop Helder[†], Ka-Wing Lam[†]
[*]University of Groningen, Netherlands, [†]MARIN, Wageningen/Netherlands
P.van.der.Plas@rug.nl

1 Introduction

In a previous study (van der Plas et al., 2015) we introduced an adaptive grid refinement method for free-surface flow simulations and presented preliminary results for various offshore applications. In a follow-up study, the grid refinement method has been extended to include support for two-phase flow (Wemmenhove et al., 2016) and to enable the inclusion of moving objects (Fekken, 2004). In this paper, we highlight some of its properties and its applications.

2 Outline of the numerical method

In the past, the CFD simulation tool ComFLOW (Veldman et al., 2016) has been successfully applied in a wide range of offshore applications, involving for example wave simulations and impact calculations. In many of these calculations the area of interest comprises a small part of the domain and remains fixed in time, which allows for efficient grid refinement by means of grid stretching or static local refinement. However, when trying to accurately resolve the surface dynamics and kinematics of irregular and breaking waves, the resolution requirements are strongly time-dependent and difficult to predict in advance. Efficient grids can then only be obtained by means of time-adaptive refinement.

A Cartesian block-based refinement approach is followed which allows for efficient grid adaptation, with moderate overhead. Upon satisfying a refinement or coarsening criterion grid blocks can be split or merged. Employing fixed-size grid blocks allows for a computationally efficient implementation of the numerical algorithms. Instead of an oct-tree, an array-based data structure is employed which exploits the semi-structured nature of the Cartesian block grid. A simple relation exists between grid indices of a parent block at level L and its underlying children at level L+1 as shown in Figure 1(a). Missing data (e.g. pressure, velocity, liquid distribution) are reconstructed in so-called guard layers, colored green and blue in Fig. 1(b), which allows to extend the regular discretization methods up to and including the boundaries of the refinement grids.

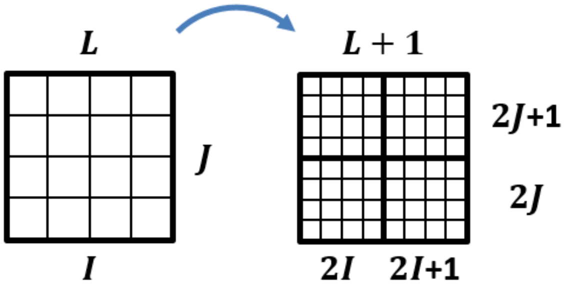

Fig 1(a): illustration of semi-structured indexing on a locally refined Cartesian grid

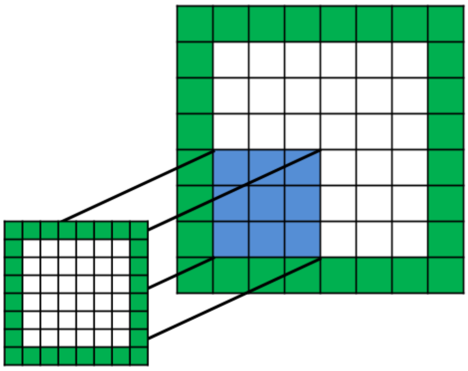

Fig 1(b): guard layers on a block-based adaptive grid

The method was introduced previously for static local grid refinement and proved useful in a variety of offshore applications (van der Plas, 2013). The modified grid data structure and the numerical discretization at the refinement interfaces is described in full detail in (van der Plas, 2017). Special attention was paid to the modified treatment of the Volume-of-Fluid scheme near refinement interfaces. At refinement interfaces the accuracy of the numerical is scheme is lower than in regular parts of the grids (first-order versus second-order). In particular if the cells are cut by the geometry or the free surface the accuracy can degrade even further. This poses no problems as long as the grid interfaces can be placed in smooth parts of the solution, however, depending on the problem at hand this may be difficult to realize. Increasing the accuracy of the interpolation operators at refinement interfaces near cut-cell geometry is a difficult task, in particular if conservation properties are to be maintained. As an alternative, we decided to adapt the computational grid using a surface- and geometry-tracking criterion which ensures that a uniform resolution is maintained around the free surface and the geometry.

3 Wave simulations

An important application area of ComFLOW is the simulation of irregular waves. One example could be the reconstruction of wave conditions as observed in an experimental test setup. In that manner CFD can be used to numerically investigate modifications in the test setup. An iterative procedure is applied in which after each run, the incoming wave signal is modified in order to better reproduce the measured water elevations inside the domain. Once a good representation is obtained, the signal is imposed on a 3-D domain including all features of the experimental setup. A more detailed description of this procedure can be found in reference (Bunnik et al., 2015).

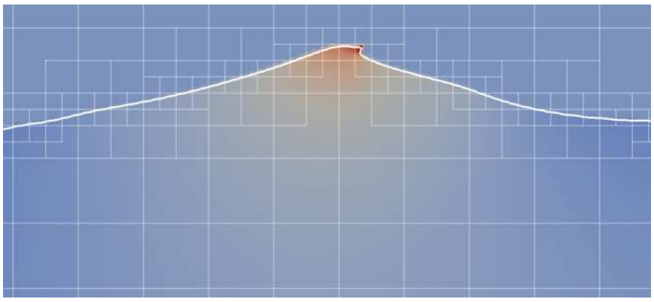

Fig. 2: Spilling breaker on an adaptive grid. The cross section is colored by the horizontal velocity.

The efficiency of wave simulations can be increased considerably by coarsening the grid away from the free surface. For moderate wave conditions, static local grid refinement can be applied straightforwardly by inserting one or more refinement strips along the entire surface. For these cases, the overhead of grid adaptivity would not pay itself back. However, under more violent wave conditions, e.g. focusing and breaking waves, static refinement becomes less efficient and grid adaptivity starts to become more interesting (see e.g. Figure 2).

Block-based (adaptive) grid refinement reduces the number of grid cells, but also introduces overhead due to data exchange and grid adaptation. On block-based grids, data has to be exchanged between adjacent blocks, even if located on the same refinement level. This means that the overhead increases with the relative number of grid interface cells, hence becomes more noticeable when smaller block sizes are used. In order to reduce the overhead of refinement interfaces, also a combined approach was investigated in which static refinement patches are applied on the lower refinement levels and adaptive grid blocks on the higher refinement levels.

For illustration purposes consider the simulation of a spilling breaker in a 2-D domain using five different types of grids as outlined in Table 1(a). Similar outcomes were obtained on all five grids.

	uniform	*patch*-based static	*block*-based static	*block*-based adaptive	*combined* static/adaptive
avg. no. liquid cells	48k	11k	11k	3.8k	4.2k
wall clock time [s]	**1143**	**311**	**543**	**237**	**201**
time per grid point [s]	0.024	0.028	0.049	0.062	0.049

Table 1(a): Computation times for a 2-D wave simulation on different types of grids. All grids have identical resolution around the free-surface. The block size (if applicable) was set to 6x6. The grids in columns 2 and 3 are effectively identical. The resolution at the free surface was set to 1x1 m.

As shown in Table 1a, employing a block-based grid is more expensive than employing a patch-based grid (compare columns 2 and 3). However, grid adaptivity compensates for this overhead and altogether yields better performance (columns 4 and 5). The best performance is obtained when using a combination of static grid patches and adaptive grid blocks (last column in Table 1a). When the resolution requirement at the free surface is increased, the differences become even more evident, as shown in Table 1b.

	uniform	*patch*-based static	*block*-based static	*block*-based adaptive	*combined* static/adaptive
avg. no. liquid cells	193k	40k	40k	9.8k	11k
wall clock time [s]	**9124**	**1612**	**2758**	**739**	**674**
time per grid point [s]	0.047	0.040	0.069	0.077	0.069

Table 1(b): Results for the same simulation as in Table 1(a), but with an extra level of refinement around the free surface. The resolution at the free surface was set to 0.5x0.5 m.

It is expected that the efficiency can be improved further by optimizing the interface treatment and by detailed code profiling. Increased efficiency can also be obtained, in particular on larger grids, by combining adjacent grid blocks into larger patches so as to reduce the number of interfaces.

The above considered 2D example only moderately benefits from introducing grid adaptivity. It is worth noting that larger (3D) problems or problems with more variation in time, show increasing benefit from grid adaptivity. One such example will be given in the following section.

4 Simulation of a free-fall lifeboat drop

In this section, we investigate the applicability of the numerical method by considering the simulation of a free-fall lifeboat drop. Measurements were performed at MARIN, which will be used for comparison. Adaptive grid refinement is used to facilitate grid setup and reduce computational time.

Currently we only consider lifeboat drops in still water, but ultimately it can be combined with incoming (irregular) waves, for example using the method described in the previous section.

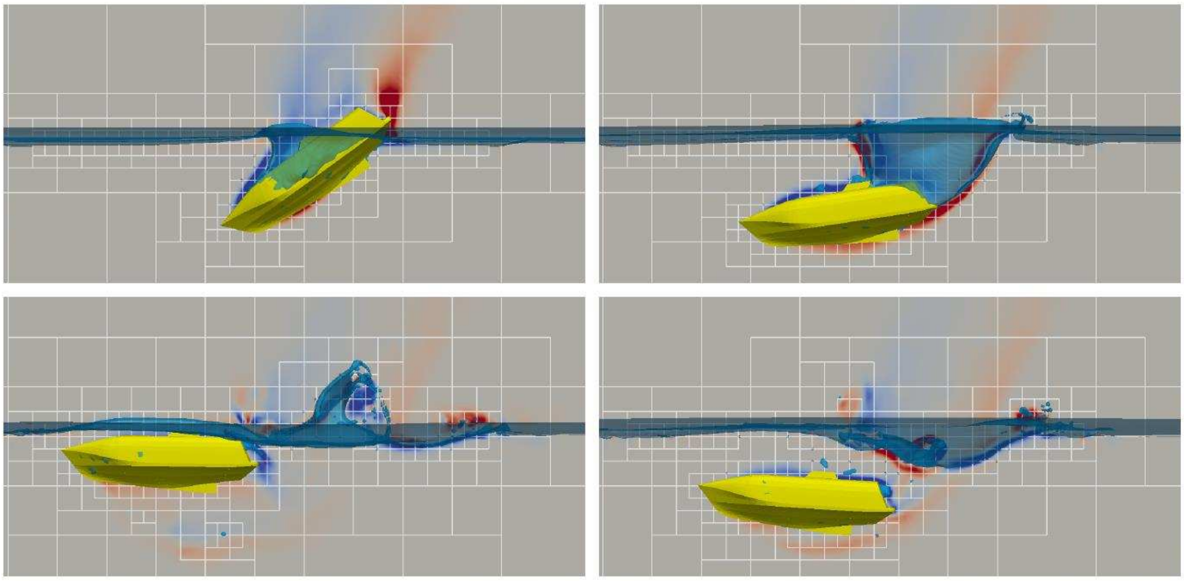

Fig. 3: Simulation of a free-fall lifeboat (snapshots are sorted clockwise, starting from the top-left corner). The grid is adaptively refined by means of a basic surface and geometry tracking criterion. Each visualized block corresponds to 6x6x6 grid cells. Coloring represents the flow vorticity in the *xz*-plane, clipped to the interval [-12; 12]. This figure was produced on an adaptive grid with a resolution of 0.125 m around both the surface and the geometry.

Here we only present the results obtained with the two-phase flow module and a second-order upwind discretization of convection. As a starting point, simulations were also performed using the one-phase flow module of ComFLOW and a first-order upwind discretization of convection. However, the results obtained with these settings resulted in a poor approximation of the measurements in the test basin. It was concluded that two-phase flow effects play a crucial role that cannot be neglected. Furthermore, it was observed that second-order upwind discretization of convection significantly improved results (we hope to further investigate the role of turbulence modeling in the near future).

grid	resolution around surface [m]	resolution around geometry [m]	#	total wall clock time [h]
2-1 / FF1000S-*coarse*	0.50	0.25	35k	3
3-2 / FF1000S-*medium*	0.25	0.125	90k	22
4-3 / FF1000S-*fine*	0.125	0.0625	320k	100

Table 2: Properties of the numerical simulations. The total **simulated** time was 7 s.

Of critical importance is the grid resolution around the object and the free surface. Various grid configurations were investigated with local resolutions ranging between 0.5 m and 0.06 m. The properties of three simulations are listed in Table 2. In the far-field, both in horizontal direction and towards the bottom of the domain, the grid resolution is coarsened towards a resolution of 2 m.

The obtained numerical results are shown in Figure 4, together with the measurements from experiment. Overall, a good agreement is observed between the numerical simulation and the measurements from experiment. Given the small differences between the solutions on the medium and fine grids it is expected that further grid refinement will only have minor effect on the solution. The remaining differences are most visible after 1.5 to 2 seconds, which is when the lifeboat is almost totally submerged and detaches from the air gap (see steps 2 and 3 in Figure 3). It is thought that the results may improve further, by modeling more of the physical aspects at play. One could think of turbulence modeling (in particular behind the lifeboat), surface tension, as well as compressibility (in particular in the region of air inclusion).

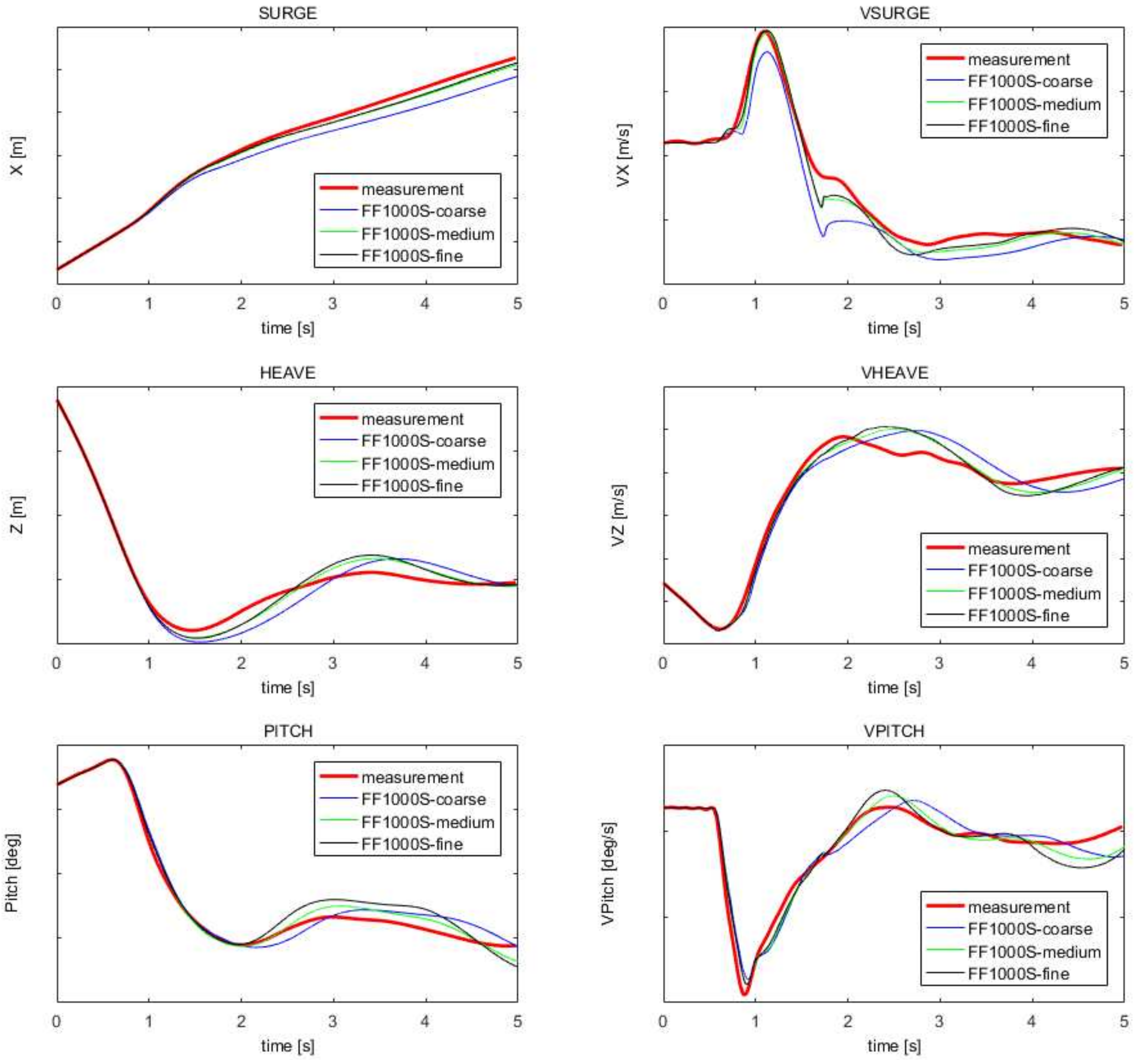

Fig. 4: Comparison of motion time traces obtained from experiment and simulation (positions on the left and velocities to the right).

Conclusions and outlook

Currently we are testing the numerical method to the simulation of lifeboat drops in waves. Ultimately our goal is to combine the functionality described in sections 3 and 4 in order to perform simulations of free-fall lifeboat drops in regular and irregular wave conditions. This poses several challenges such as accurately imposing the incoming waves and modifying the absorbing boundary conditions to support two-phase flow. Furthermore, the computational efficiency will be improved by means of MPI parallelization and further optimization of the adaptive grid refinement method. As mentioned in section 4, we also intend to investigate the role of other physical aspects, such as turbulence, surface tension and compressibility.

References

T. Bunnik, J. Helder and E-J. de Ridder (2015), Deterministic Simulation of Breaking Wave Impact and Flexible Response of a Fixed Offshore Wind Turbine, *Proceedings* OMAE 2015.

G. Fekken: Numerical simulation of free surface flow with moving rigid bodies. *PhD thesis*, University of Groningen (2004).

P. van der Plas, A.E.P. Veldman, et al. (2015), Adaptive Grid Refinement for Free-Surface Flow Simulations in Offshore Applications, *Proceedings* OMAE 2015.

P. van der Plas, et al. (2013), Local Grid Refinement for Free-Surface Flow Simulations, *Proceedings* NuTTS Symposium 2013.

P. van der Plas (2017), Local Grid Refinement for Free-Surface Flow Simulations, *PhD thesis*, University of Groningen

A.E.P. Veldman, R. Luppes, et al. (2016), Locally Refined Free-Surface Flow Simulations for Moored and Floating Offshore Platforms, *Proceedings* ISOPE 2016.

R. Wemmenhove, R. Luppes, A.E.P. Veldman and T. Bunnik (2015). Numerical simulation of hydrodynamic wave loading by a compressible two-phase flow method. Comp. Fluids, 114 (2015)