V International Conference on Computational Methods in Marine Engineering
MARINE 2013
B. Brinkmann and P. Wriggers (Eds)

LOCAL GRID REFINEMENT FOR FREE-SURFACE FLOW SIMULATIONS IN OFFSHORE APPLICATIONS

PETER VAN DER PLAS*, HENRI J. L. VAN DER HEIDEN*, ROEL LUPPES* AND ARTHUR E. P. VELDMAN*

*Institute for Mathematics and Computer Science
University of Groningen (RuG)
P.O. Box 407, 9700 AK Groningen, The Netherlands
Corresponding author: P.van.der.Plas@rug.nl

Key words: Computational Methods, Marine Engineering, Local Grid Refinement, Free-Surface Flow

Abstract. A local grid refinement approach is presented for free-surface flow simulations. A semi-structured approach is used based on rectangular refinement regions, inside which the grid is locally structured, allowing for the application of efficient solution methods. At refinement interfaces a simple data structure facilitates the look-up of neighbouring cells. Near refinement interfaces, a modified discretization stencil is introduced for the Navier-Stokes equations. To allow for efficient grid configurations, the interface scheme also needs to perform well near objects and free-surface boundaries. Therefore, special attention is paid to designing a compact interface scheme that can perform well in a wide variety of industrial applications. Numerical results are presented for flow around a square cylinder as well as the simulation of a breaking dam.

1 INTRODUCTION

In offshore applications, extreme events of wave impact on rigid and floating structures are of high interest. In the past the CFD simulation tool ComFLOW [1, 2] has been successfully used for these purposes. For accurate prediction of wave run-up and wave loading on offshore structures high resolution is only required in the areas of interest, whereas in the far field coarse grids are sufficient. Up to now, further reduction of grid points was only possible by means of grid stretching which typically results in large deformation of grid cells and due to its poor locality is not very efficient. In the ComFLOW-3 project one of the aims is to increase numerical efficiency by introducing local grid refinement.

2 DISCRETIZATION OF THE NAVIER–STOKES EQUATIONS

An excellent model for incompressible fluid flow is provided by the Navier-Stokes equations. The set of equations consists of the continuity equation

$$\mathcal{M}\mathbf{u} = 0, \tag{1}$$

where $\mathcal{M} = \nabla\cdot$ is the divergence operator, and the momentum equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathcal{C}(\mathbf{u},\mathbf{u}) + \mathcal{G}p - \mathcal{D}\mathbf{u} = \mathbf{f}, \tag{2}$$

based on the convection operator $\mathcal{C}(\mathbf{u},\mathbf{v}) = \mathbf{u}\cdot\nabla\mathbf{v}$, the pressure gradient operator $\mathcal{G} = \nabla$, the diffusion operator $\mathcal{D}(\mathbf{u}_h) = \nabla\cdot\nabla\mathbf{u}_h$ and forcing term $\mathbf{f}$.

The continuity equation (1) is discretized at the 'new' time level $n+1$ to give

$$M\mathbf{u}_h^{n+1} = -M^\Gamma\mathbf{u}_h^{n+1} \tag{3}$$

where M acts on the internal of the domain and M^Γ acts on the boundaries of the domain.

Convection and diffusion are discretized explicitly in time. The divergence and the pressure gradient are discretized at the new time level. If we denote the diagonal matrix containing the fluid volumes of the momentum cells by Ω, the discretized momentum equation is given by

$$\Omega\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^n}{\Delta t} = -C(\mathbf{u}_h^n)\mathbf{u}_h^n + D\mathbf{u}_h^n - G\mathbf{p}_h^{n+1}. \tag{4}$$

Finding the solution to the system of equations (3) & (4) is split in two steps. First an auxiliary variable $\mathbf{u}_h^*$ is defined by the equation

$$\Omega\frac{\mathbf{u}_h^{*} - \mathbf{u}_h^n}{\Delta t} = -C(\mathbf{u}_h^n)\mathbf{u}_h^n + D\mathbf{u}_h^n. \tag{5}$$

Using this variable in (4) gives

$$\Omega\frac{\mathbf{u}_h^{n+1} - \mathbf{u}_h^{*}}{\Delta t} = -G\mathbf{p}_h^{n+1}. \tag{6}$$

Substitution of equation (6) in the continuity equation (3), gives rise to the following system of equations

$$\Delta t M\Omega^{-1}G\mathbf{p}_h^{n+1} = M\mathbf{u}_h^{*} + M^\Gamma\mathbf{u}_h^{n+1}, \tag{7}$$

which is often referred to as the discrete pressure Poisson equation, as it can be viewed as a discretization of the equation $\mathcal{M}\circ\mathcal{G}\mathbf{p} = \mathcal{M}\mathbf{u}$. Note, however, that we are not directly discretizing the composed operator $\mathcal{M}\circ\mathcal{G}$ here, but its separate parts $\mathcal{M}$ and $\mathcal{G}$. Hence, of sole importance is the accuracy of the discretization of the divergence and gradient operators $\mathcal{M}$ and $\mathcal{G}$, respectively. This should be kept in mind when assessing the accuracy of the method.

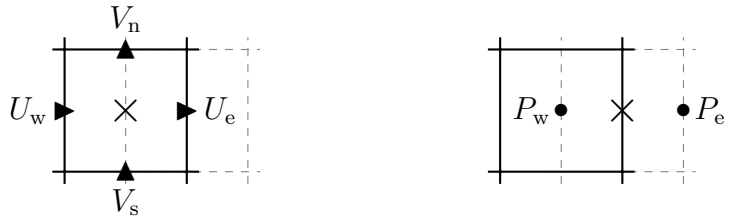

Figure 1: Regular discretization stencil for $M\mathbf{u}_h$ (left) and $G\mathbf{p}_h$ (right, horizontal component only) applied at the location indicated with $\times$.

3 LOCAL GRID REFINEMENT

The Navier–Stokes equations are discretized on an Arakawa C-grid as illustrated in fig. 1. For brevity the third dimension, which is treated similarly, is omitted. The subscript ℓ is used to indicate the local refinement level, where $\ell = 0$ refers to the unrefined base grid (indexing is discussed in section 3.1.)

In the regular parts of the grid the divergence operator is discretized as follows (for the subscript convention consult fig. 1)

$$M\mathbf{u}_h|_\ell = \Delta y_\ell (U_{\mathrm{e};\ell} - U_{\mathrm{w};\ell}) + \Delta x (V_{\mathrm{n};\ell} - V_{\mathrm{s};\ell}) \tag{8}$$

In order to let the discrete operators satisfy the adjoint condition,

$$G = -M^* \tag{9}$$

the pressure gradient is discretized as $G\mathbf{p}_h = -M^*\mathbf{p}_h$. This gives the following second-order central discretization:

$$\frac{P_{\mathrm{e};\ell} - P_{\mathrm{w};\ell}}{\Delta x_\ell} \qquad\qquad \frac{P_{\mathrm{n};\ell} - P_{\mathrm{s};\ell}}{\Delta y_\ell} \tag{10}$$

3.1 Refinement approach

A semi-structured approach is followed in which a cell (i, j) at refinement level ℓ is *replaced* by a set of $r_i \times r_j$ smaller cells at refinement level $\ell+1$ having indices $(2i+m, 2j+n)$ at offsets $0 \leq m < r_i, 0 \leq n < r_j$. The semi-structured indexing system is illustrated in fig. 2. On block-shaped refinement regions the method is locally structured, hence the computational efficiency of the original array-based solution methods can be exploited as much as possible. Only at the boundaries of the refinement regions where the actual refinement takes place a new treatment is required.

For describing the grid layout an auxiliary array is introduced storing only one integer for each potentially occuring cell $(i, j; \ell)$ pointing at the memory location of the subgrid in which it is contained (or null if the cell does not exist). Along the lines of [3] a data structure results that allows for fast and efficient look-up when compared with typical tree-based storage methods.

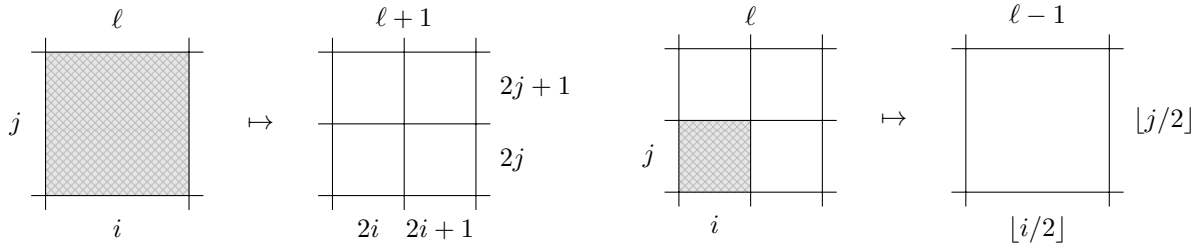

Figure 2: Illustration of semi-structured indexing for refinement ratios $r_i = 2, r_j = 2$. *Left*: From coarse to fine indices. *Right:* From fine to coarse indices.

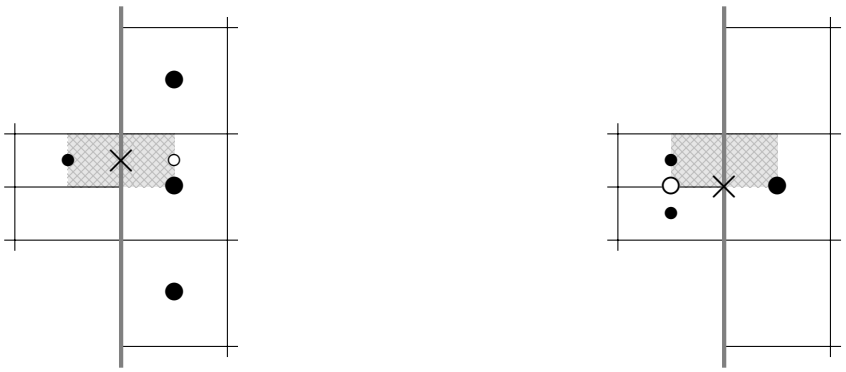

Figure 3: *Left*: refinement approach based on interpolation of missing refined pressure variable, *Right*: refinement approach (as followed here) with shifted pressure gradient which is equal for both refined cell faces. Missing variables are denoted by "○".

3.2 Poisson equation near interfaces

Near refinement interfaces the discretization stencil is incomplete due to missing coarse or fine grid variables. Typically, a large stencil is used for the approximation of missing pressure or velocity variables along the refinement interface. Interpolation of missing variables increases the number of non-zero coefficients in the pressure Poisson matrix, which might result in a non-symmetric matrix, putting higher demands on the solver. Most authors use a non-overlapping interface and apply linear (or even higher-order) interpolation for missing variables on the other side of the interface [4]. Another approach is to apply linear interpolation inside an overlapping interface [5]. In all cases the discretization results in a non-symmetric system of equations.

In the present approach, a compact discretization scheme is designed (in particular for the implicit part), which results in a small and symmetric scheme for the discrete composition of M and G. This makes it possible to employ an efficient linear solver. Furthermore, it facilitates the use of adjacent refinement regions as well as the interface discretization near objects and free-surface boundaries.

3.2.1 Spatial discretization

Remark (i) *As an example we consider refinement interfaces in the "x=constant"-plane where the refined cells are located to the left of the interface. Five other interface orientations are possible, which are treated similarly. To further simplify discussion we assume a base grid with uniform grid spacings Δx_0 and Δy_0.*

Remark (ii) *In the current discussion we take refinement ratios $r_i = 1$ and $r_j = 2$. In words, no refinement is applied perpendicular to the refinement interface and local refinement is only applied along the refinement interface. For the grid spacings this implies $\Delta y_{\ell+1} = \Delta y_\ell/2$ and $\Delta x_{\ell+1} = \Delta x_\ell$. Since $r_i = 1$, for the latter, the level subscript will be omitted.*

Extending the discretization to the three-dimensional case, non-uniform grids and other refinement directions or refinement ratios is straightforward.

There are two ways of obtaining a first-order accurate discretization of the pressure gradient. Either by using a linear interpolation for the missing pressure variable outside the refinement region (see left of fig. 3) or by slightly shifting the location of the pressure gradient (see right of fig. 3). Both approaches result in a first-order accurate discretization scheme introducing an error term that is proportional to respectively $\Delta\frac{\partial^2 p}{\partial y^2}$ and $\Delta\frac{\partial^2 p}{\partial y \partial x}$, where for brevity we use Δ which has the same order of magnitude as Δx and Δy. However, the first approach results in a relatively large stencil whereas the second approach uses a smaller interpolation stencil consisting of pressure variables that already form part of the regular stencil.

For this reason, the second approach is followed, which can be described as "using a constant pressure gradient along a refined cell face" (see e.g. [6, 7]). Correspondingly, we use a uniform velocity across the entire refined cell face and only place coarse computational velocity variables at the interface.

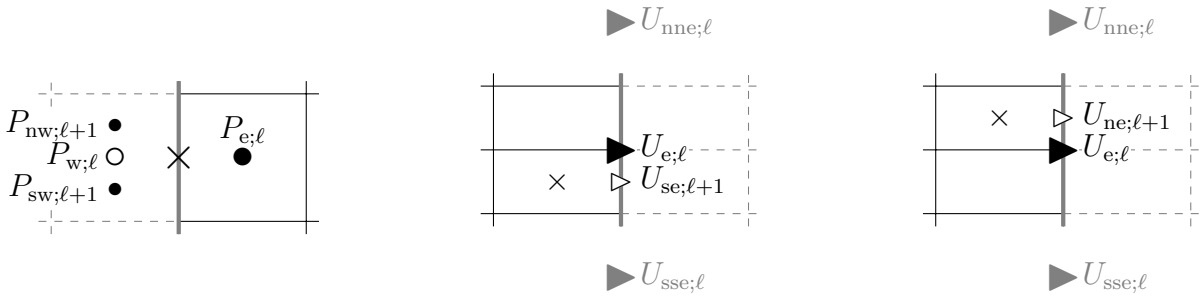

Figure 4: *Left:* Missing *coarse* pressure variable (○) for the gradient operator applied at the location indicated with ×. *Right:* Missing *refined* velocity variable (▷) for the divergence operator applied at the location indicated with ×, together with the variable used for constant extrapolation (▶) and linear correction (▶).

For the missing coarse pressure variable $P_{\mathrm{w};\ell}$ (see fig. 4) a simple average of the neighbouring fine pressure values is used. This results in the following discretization of the pressure derivative at the refined cell face

$$\frac{2P_{\mathrm{e};\ell} - P_{\mathrm{sw};\ell+1} - P_{\mathrm{nw};\ell+1}}{2\Delta x} = \frac{\partial p}{\partial x}(\mathbf{x}_\ell^u) + O(\Delta).$$

Note that the approximation for the missing pressure variable is second-order accurate, but one order of accuracy is lost due to the loss of symmetry in the central scheme. Hence the approximation of the pressure gradient across the refinement interface is first-order accurate.

For the discretization of the divergence at the fine side of refinement interfaces, an approximation is needed for the missing fine velocities. As a first approach, a discretization is obtained by means of the adjointness condition (9), so the divergence operator is defined as the negative transpose of the gradient operator. This implies that missing velocities are simply approximated using constant extrapolation (see fig. 4)

$$U_{\mathrm{se};\ell+1} \quad := U_{\mathrm{e};\ell} = u(\mathbf{x}^U_{\mathrm{se};\ell+1}) + O(\Delta), \qquad U_{\mathrm{ne};\ell+1} := U_{\mathrm{e};\ell} = u(\mathbf{x}^U_{\mathrm{ne};\ell+1}) + O(\Delta).$$

The corresponding divergence operator for interface cells is then given by

$$\begin{aligned}
\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{s};\ell+1} &= \Delta y_{\ell+1}\left(U_{\mathrm{e};\ell} - U_{\mathrm{sw};\ell+1}\right) + \Delta x\left(V_{\mathrm{c};\ell+1} - V_{\mathrm{s};\ell+1}\right) \\
\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{n};\ell+1} &= \Delta y_{\ell+1}\left(U_{\mathrm{e};\ell} - U_{\mathrm{nw};\ell+1}\right) + \Delta x\left(V_{\mathrm{n};\ell+1} - V_{\mathrm{c};\ell+1}\right)
\end{aligned}$$

Taking a uniform velocity along the refined cell face is conform the earlier remark of using a uniform pressure gradient along the refined cell face. However, it can be seen that the scheme for the divergence operator is not consistent yet because the velocities in the central difference are not well aligned in the y-coordinate.

$$\begin{aligned}
\frac{1}{\Delta x\Delta y_{\ell+1}}\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{s};\ell+1} &= \frac{\partial u}{\partial x}(\mathbf{x}^P_{\mathrm{s};\ell+1}) + \frac{\partial v}{\partial y}(\mathbf{x}^p_{\mathrm{s};\ell+1}) + \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\frac{\partial u}{\partial y}(\mathbf{x}^u_{\mathrm{e};\ell}) + O(\Delta) \\
\frac{1}{\Delta x\Delta y_{\ell+1}}\left(\overline{M}\mathbf{u}_h\right)_{\mathrm{n};\ell+1} &= \frac{\partial u}{\partial x}(\mathbf{x}^P_{\mathrm{n};\ell+1}) + \frac{\partial v}{\partial y}(\mathbf{x}^p_{\mathrm{n};\ell+1}) - \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\frac{\partial u}{\partial y}(\mathbf{x}^u_{\mathrm{e};\ell}) + O(\Delta)
\end{aligned}$$

The inconsistency can be resolved by adding corrections for the observed error terms. In order to satisfy mass conservation it is important that these corrections sum up to zero for each refined cell face. For this we can make use of the symmetry observed in the above error terms and correct the operator $\overline{M}$ with the following linear correction terms:

$$\frac{1}{\Delta x\Delta y_{\ell+1}}\left(M^+\mathbf{u}_h\right)_{\mathrm{s};\ell+1} = -\frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\left[\delta_y\mathbf{u}_h\right]_{\mathrm{e};\ell} \tag{12}$$

$$\frac{1}{\Delta x\Delta y_{\ell+1}}\left(M^+\mathbf{u}_h\right)_{\mathrm{n};\ell+1} = \frac{1}{2}\frac{\Delta y_{\ell+1}}{\Delta x}\left[\delta_y\mathbf{u}_h\right]_{\mathrm{e};\ell} \tag{13}$$

where δ_y is a central differencing operator which is applied along the refinement interface (see fig. 4)

$$[\delta_y \mathbf{u}_h]_{\mathrm{e};\ell} = \frac{U_{\mathrm{nne};\ell} - U_{\mathrm{sse};\ell}}{2\Delta y_\ell} \tag{14}$$

The divergence operator with correction, i.e. $\overline{M} + M^+$ is now first-order accurate.

We remark that the correction operator M^+ is similar for other interface orientations. Note that in the three-dimensional case the operator would also include a difference term in the secondary direction tangential to the interface.

4 NUMERICAL RESULTS

In order to investigate the performance of the local grid refinement scheme, two-dimensional simulations of flow around a square cylinder have been performed. All simulations at Reynolds numbers 10 and 100 have been performed using a second-order central discretization without the use of a turbulence model.

cyl.	grid	L	# pts	C_d
2×4	80×80	0	6k	3.1374
	40×40	1	3k	3.1598
3×6	120×120	0	14k	3.2590
	60×60	1	6k	3.2589
4×8	160×160	0	26k	3.2918
	80×80	1	19k	3.2917
	40×40	2	4k	3.2908
6×12	240×240	0	58k	3.3138
	60×60	2	9k	3.3133
	30×30	3	3k	3.3099
12×24	480×480	0	230k	3.3297
	60×60	3	12k	3.3290
	30×30	4	5k	3.3243
24×48	960×960	0	922k	3.3383
	60×60	4	19k	3.3373
	30×30	5	9k	3.3323

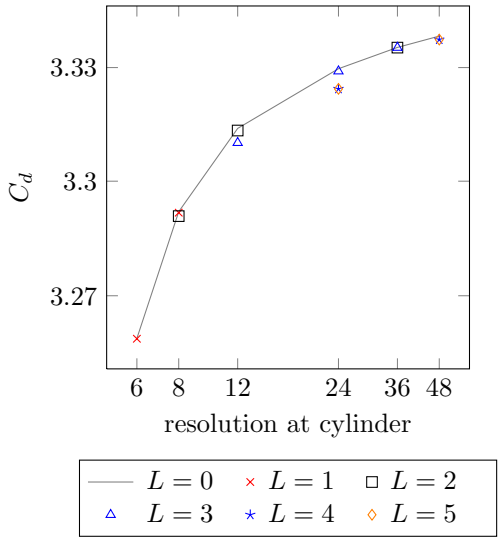

Table 1: Drag-coefficient predictions (C_d) for flow around a square cylinder (Re=10) on uniform, and locally refined grids. The column 'cyl.' displays the grid resolution at the boundary of the cylinder. The column 'grid' shows the resolution of the base grid to which the local grid refinement is applied. In all cases a refinement ratio of 2×2 is used and L indicates the number of local refinement regions. In the column '# pts' the total number of grid points is displayed.

In order to get a good view of the efficiency gain that is obtained with the local grid refinement approach, the analysis is best performed from a "coarsening" point of view. Near the object the grid resolution is kept constant while the grid is coarsened towards the boundaries of the domain.

4.1 Flow around a square cylinder

In this 2-D test case a square cylinder is placed of dimensions $[-0.5, 0.5] \times [-0.5, 0.5]$ in a computational domain covering the region $[-10, -10] \times [30, 10]$.

At a Reynolds number of 10 the flow readily converges to a steady-state solution. The resulting solution is smooth and is not expected to pose any difficulties for refinement interfaces. The numerical results presented in table 1 show that the drag-force predictions are accurate even on grids that are very coarse close to the boundaries of the domain. By accepting a 0.5% difference in the prediction of the drag force it is possible to reduce the computational time by up to factor of 100. The resolution of the grid close to the object is of main importance, and it is seen that good convergence behaviour is obtained when increasing the number of cells around the cylinder.

At a Reynolds number of 100 the flow is unsteady, as the flapping shear layer results in an oscillating drag and lift force on the cylinder. This test case clearly provides a more challenging test for the local grid refinement method. A local grid refinement ratio of 3×3 is used and the results of the locally refined grids are compared to their uniform counterparts. The results shown in table 2 illustrate again that the number of grid points can be reduced significantly while useful predictions for the drag and lift coefficients can still be obtained.

cyl.	grid	L	#pts	St	C_d	$C_\mathrm{l,rms}$
6×12	240×240	0	58k	0.152	1.6275	0.2567
	80×80	1	19k	0.153	1.6273	0.2554
9×18	360×360	0	129k	0.150	1.5687	0.2231
	40×40	2	12k	0.150	1.5687	0.2208
18×36	720×720	0	0.5M	0.150	1.5234	0.2009
	80×80	2	48k	0.150	1.5232	0.1998

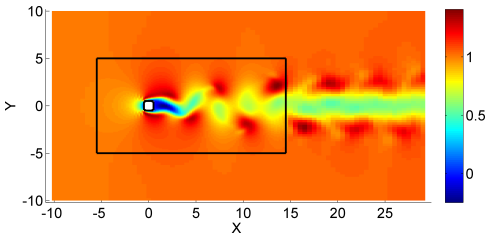

Table 2: Numerical predictions of the Strouhal number (St), mean drag force C_d, and the root-mean-square of the lift coefficient $C_\mathrm{l,rms}$ for flow around a square cylinder (Re=100) on uniform and locally refined grids. In all cases a refinement ratio of 3×3 is used. Again, the 'cyl.' denotes the grid resolution around the cylinder, 'grid' the resolution of the base grid, L the number of refinements, and '# pts.' the total number of grid points.

4.2 Dambreak experiment

To demonstrate the validity of the local refinement method for more practical cases, it is tested for the simulation of a breaking dam. ComFLOW has been used before for this classical test [1], therefore it provides good material for assessing the performance of the local refinement method.

As starting point a grid is used of $200 \times 36 \times 50$ points. In order to save computational time, the original resolution is only maintained around the block; to the right as well as towards the sides of the domain it is coarsened (as illustrated in fig. 5).

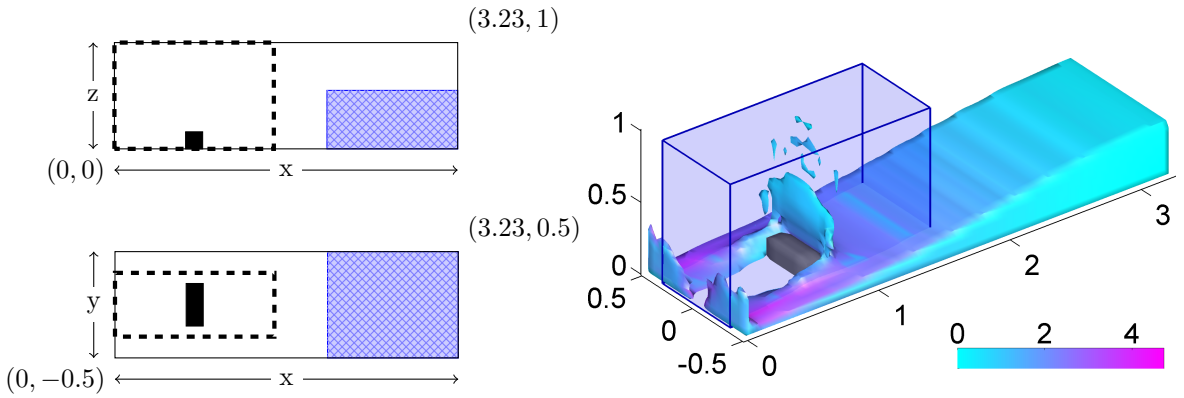

Figure 5: *Left:* Cross sections of the domain illustrating the grid configuration. The original resolution is only kept inside the dashed box. *Right:* Isosurface of the volume fractions colored by the absolute velocity of the free surface.

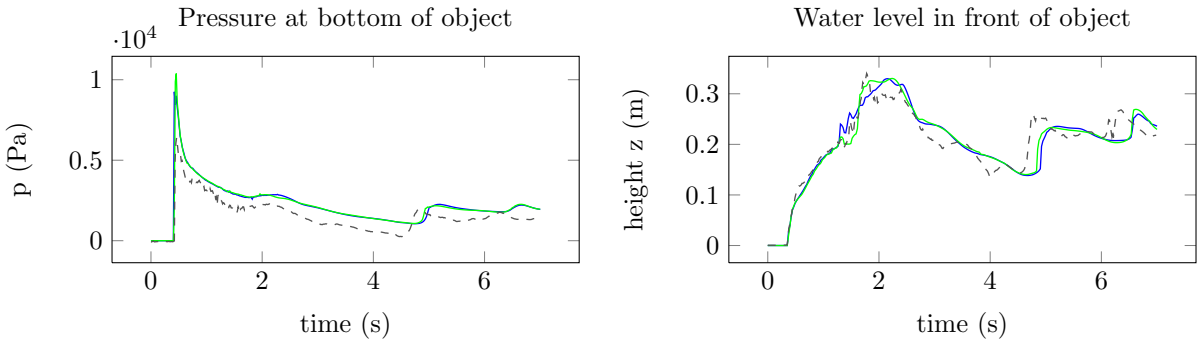

Figure 6: Numerical results and measurements for the 3-D dambreak experiment. *Legend:* —— coarsened grid; —— uniform grid; - - - experiment.

The simulation results show good correspondence to the measurements and it can be seen that coarsening in the region of the reservoir does not affect the prediction of the impact pressure. The differences between the locally coarsened and uniform grid are much smaller than the actual modelling error while the computational time has been reduced from 8h05 to 0h38. This illustrates that for typical "impact" problems a significant computational saving can be made by coarsening in the far away regions.

5 CONCLUSIONS

In this paper a local grid refinement approach has been presented for the simulation of free-surface flow. Special attention was paid to designing a compact stencil because the method needs to be accurate and robust in a wide variety of settings. In particular this facilitates the interface discretization near cut-cells and the modification of the Volume-of-Fluid scheme. Furthermore, the compact interface scheme is easily adapted to support refinement corners that occur when concatenating rectangular refinement regions, hence allowing for even more efficient grid configurations.

The local refinement approach has been succesfully applied to the simulation of turbulent flow, wave simulations and (wave) impact problems. Several simulation results were presented to verify and validate the method. In particular for calculating drag forces or (wave) impact forces the reduction of computational costs can be significant. Currently the refinement method is being extended to support two-phase flow and moving objects.

REFERENCES

[1] K.M.T. Kleefsman, G Fekken, A.E.P. Veldman, B Iwanowski, and B. Buchner. A volume-of-fluid based simulation method for wave impact problems. *Journal of Computational Physics*, 206(1):363–393, 2005.

[2] A.E.P. Veldman, R. Luppes, T. Bunnik, R.H.M. Huijsmans, B. Duz, B. Iwanowski, R. Wemmenhove, M.J.A. Borsboom, P.R. Wellens, H.J.L. van der Heiden, and P. van der Plas. Extreme wave impact on offshore platforms and coastal constructions. *30th Conf. on Ocean, Offshore and Arctic Eng. OMAE2011, Rotterdam, Netherlands, June 19-24*, 2011.

[3] M.D. de Tullio, P. De Palma, G. Iaccarino, G. Pascazio, and M. Napolitano. An immersed boundary method for compressible flows using local grid refinement. *Journal of Computational Physics*, 225(2):2098–2117, 2007.

[4] M.L. Minion. A projection method for locally refined grids. *Journal of Computational Physics*, 127(1):158–178, 1996.

[5] Q. Liu. A stable and accurate projection method on a locally refined staggered mesh. *International Journal for Numerical Methods in Fluids*, 67(1):74–92, 2011.

[6] E. Uzgoren, R. Singh, J. Sim, and W. Shyy. Computational modeling for multiphase flows with spacecraft application. *Progress in Aerospace Sciences*, 43(4):138–192, 2007.

[7] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 457–462. ACM, 2004.