Implementation and Verification of an Explicit Overset Grid Method

Sébastien Lemaire[*†], **Guilherme Vaz**[*], **and Stephen Turnock**[†]
[*]MARIN, Wageningen/Netherlands, [†]FSI Group, University of Southampton, Southampton/UK
sebastien.lemaire@soton.ac.uk

1 Introduction

The design of modern ships capable of achieving complex manoeuvres and handle dynamic ocean environment requires a precise understanding of the flow around hulls, propellers and control surfaces. Computational fluid dynamics already allows complex simulations to be performed thanks to conventional mesh techniques. Moving, sliding and deforming grids allow for some dynamic simulation of moving bodies (Toxopeus and Bhawsinka, 2016). However, when considering more complex motions, e.g. 6 degrees of freedom of several bodies, the combination of previously mentioned conventional techniques is at its limit. The overset grid method (or overlapping grid method), allows large and arbitrary relative motion of bodies by overlapping several meshes. The overset method was first designed for the aerospace industry (Benek et al., 1985) but is now also used for maritime applications like ship motions (Carrica et al., 2013), submarine maneuvers (Martin et al., 2015), etc.

This paper presents an initial implementation of overset using the two libraries Suggar++ and DiRTlib to compute the domain connectivity information (DCI). Manufactured solutions and analytical solutions are used in order to quantitatively assess the discretisation errors.

The paper is organised as follows: section 2 presents the overset approach and its nomenclature. Details of the implementation are introduced in section 3. Section 4 describes the test cases used for the study. Results are presented in section 5 and conclusions are given in section 6.

2 The overset approach

The overset method, sometimes called *chimera technique* or *overlapping grid method*, works by transferring field data between meshes and by ignoring certain cells which are outside the domain or overlapped by another mesh. Each cell has a given geometrical status and solver status which can change during the calculation:

- *Hole cell*: a field cell that is outside the boundary of the domain or not used because it is overlapped by another mesh. It is ignored when solving the transport equations (ignore status).
- *Fringe cell*: a fringe cell is a cell adjacent to a *hole* or at the boundary of an embedded grid. It will act as boundary cell for its grid and get its field value from the interpolation of donor cells of another mesh (interpolated status).
- *Orphan cell*: an orphan cell is a *fringe* cell that does not have an acceptable donor to compute its interpolated value. Its field value is usually then computed as the average of its neighbouring cells.
- *In cell*: every cell that is an active solution cell and not a *hole*, a *fringe* nor an *orphan* cell is called *in* cell. The equations are solved only for these cells. No particular treatment related with the grid assembly is needed (solved status).

The *domain connectivity information* (DCI) contains the cell status information sometimes associated with the interpolation weights needed to compute fringe values from donor cells. When grids are moving or being modified, the DCI needs to be recomputed. From the DCI, the flow solver will modify its equations and matrix system to ignore *hole* cells, get interpolation data on *fringe* and *orphan* cells and solve normally *in* cells.

3 Overset implementation

ReFRESCO (`refresco.org`) is a community-based open-usage CFD code for the maritime world. It solves multiphase incompressible flows using the Navier-Stokes equations, complemented with turbulence models, cavitation models and volume-fraction transport equations for different phases. The equations are discretised using a finite-volume approach with cell-centered collocated variables.

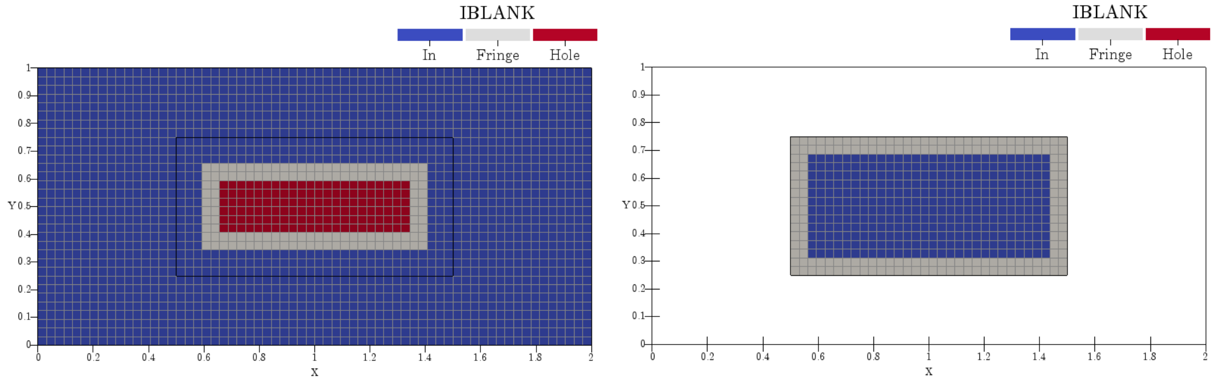
The implementation of the overset conducted here is using two external libraries called Suggar++ (Noack and Boger, 2009) and DiRTlib (Noack, 2005). Suggar++ computes the DCI and interpolation weights from the grid geometry and DiRTlib ease the treatment and use of the output from Suggar++. In this first implementation, Suggar++ provides cell status information (*fringe/hole/orphan/in* cells), donor associated to each *fringe* cell as as well as the interpolation weights via a `.dci` file. DiRTlib then provides to the flow solver an IBLANK array, an integer field data where the status of each cell is defined. Each outer loop, *fringe* cells get interpolated data from their donor cells currently using a first order inverse distance scheme provided by Suggar++. The coupling with the flow solver is done explicitly. Interpolated values are set at the right hand side of the system, and a unitary diagonal is applied on the lines corresponding to the *fringe* cells. Brunswig et al., 2010 compared several coupling approaches for overset method, and concluded that an explicit coupling needs more outer loops to converge to the same level as an implicit one and is more challenging to make mass conservative. However an implicit coupling demands more modifications on the flow solver code, hence an explicit method has been implemented as a first approach. When the continuity equation is used, the SIMPLE algorithm is employed to correct the pressure field. In this method, an estimated pressure is used to solve the momentum equation, then a pressure correction is solved to compensate for mass imbalance. In the overset explicit coupling implemented here, the pressure on the *fringe* cells is known (from interpolated data from the donor cells), and no pressure correction is performed on these cells. To implement this, a unitary diagonal is set for each *fringe* cells in the pressure correction matrix, together with a value of zero in the right hand side, and therefore the pressure field on the *fringe* cells is directly interpolated from its donor cells. Several methods have been described in Völkner et al., 2017 and Hadžić, 2005 to enforce mass conservation on the fringe of the overlapping grids, but none of them are used here.

4 Test cases definition

The overset implementation have been tested on three different analytical/manufactured 2D cases: *Advection* only case, *Advection-diffusion* case and *Poiseuille* flow case. Only the latter two will be presented here. The same five grid assemblies were used on the two test cases. It consists of a background grid and a foreground one, both being 2D Cartesian meshes with a 2:1 ratio. The foreground grid is twice smaller in both direction compared to the background grid and is positioned in the middle of the background grid. The cell size in between the two grids is the same, hence in the overlapping region the two grids are perfectly aligned, leading to a one-to-one match between cells in this area. Table 1 shows the different cell counts.

	Background	N_i	Foreground	N_i^f
G1	32x16	512	8x4	32
G2	64x32	2048	16x8	128
G3	128x64	8192	32x16	512
G4	256x128	32768	64x32	2048
G5	512x256	131072	128x64	8192

Table 1: Details of the Cartesian grids used for the two test cases and cell count (N_i)



(a) Background grid

(b) Foreground grid

Fig. 1: G2 grids and DCI colored thanks to the IBLANK array. The overlap region is composed of five “layers” of cells.

Figure 1 shows the grid assembly, which has been precomputed by Suggar++. Two layers of *fringe* cells are used and one layer of *in* cells is defined between the *fringe* cells of the background and foreground grids. This leads to an overlap of five cells between the two grids of the assembly. Cells in the middle part of the background mesh, which overlap the foreground one, are *hole* cells. Each *fringe* cell has one donor from the other grid. Such grid assembly was designed as a first step to remove the influence an interpolation scheme can have on results.

4.1 Advection-diffusion case

The *Advection-diffusion* case was generated using a manufactured solution. To construct a manufactured solution case, a field solution is defined. Here, the advection of a Gaussian profile in a circular velocity field was chosen (Eq. 3).

$$\nabla \cdot (\rho U \phi) - \nabla \cdot (\mu \nabla \phi) = q \quad . \tag{1}$$

$$U_x = y \quad , \quad U_y = 1 - x \quad . \tag{2}$$

$$\phi(r) = e^{-2r} sin^2(\pi r) \quad , \quad r = \sqrt{(1-x)^2 + y^2} \quad . \tag{3}$$

Eq. 1 is being solved with the velocity field enforced with Eq. 2. The source term q is chosen in order to have ϕ (Eq. 3) as the analytical solution of the advection-diffusion equation (Eq. 1). This method allows to construct test cases where the exact analytical solution is known. The exact solution is enforced at the boundary thanks to Dirichlet boundary conditions. The steady computation is stopped when the infinity norm of the residuals is below 10^{-12}. A QUICK scheme is used for the discretisation of the convective fluxes.

4.2 Poiseuille flow test case

In the *Poiseuille* flow case an analytical solution is known. The flow is bounded between two walls (top and bottom) and driven by an axial pressure gradient. The analytical solution is presented in Eq. 4 with L the length of the channel (X direction), and h the distance between the walls.

$$U_x(y) = \frac{Ph^2}{2\mu}\left(1 - \frac{y^2}{h^2}\right) \quad , \quad U_y = 0 \quad , \quad \frac{\partial p}{\partial x} = -P \quad , \quad p(x) = -P(x - L) \quad . \tag{4}$$

In this test case the momentum equation and the pressure correction via the SIMPLE method are solved. The Reynolds number, based on the height of the channel, is $Re_h = 10$.

At the inlet, the analytical velocity profile is set and the outlet the pressure is enforced. The top and bottom of the domain uses non slip wall boundary conditions. Convergence is ensured by an iterative criterion of 10^{-12} for the infinity norm of the residuals. A second order central difference scheme is used for the discretisation of convective fluxes.

5 Results and discussion

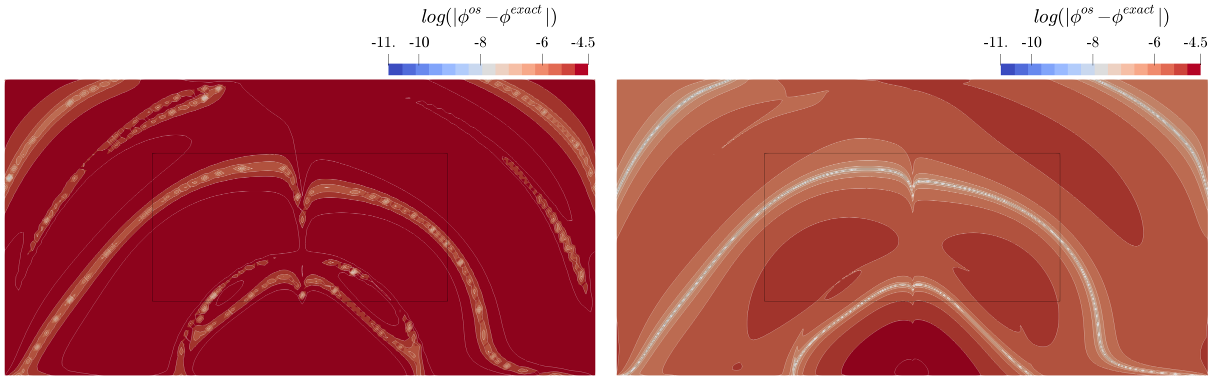
For both test cases, simulations with overset as well as simulations done using only the background grid were performed (without overset). Comparison with the exact analytical solution is also available.

Grid	$\|\phi^{os} - \phi^{exact}\|_\infty$	order	$\|\phi - \phi^{exact}\|_\infty$	order
G1	$0.787738275518 \cdot 10^{-02}$		$0.787738275519 \cdot 10^{-02}$	
G2	$0.225550008727 \cdot 10^{-02}$	1.80	$0.225550008725 \cdot 10^{-02}$	1.80
G3	$0.590247617471 \cdot 10^{-03}$	1.93	$0.590247617400 \cdot 10^{-03}$	1.93
G4	$0.149722446897 \cdot 10^{-03}$	1.98	$0.149722446791 \cdot 10^{-03}$	1.98
G5	$0.375999643849 \cdot 10^{-04}$	1.99	$0.375999643008 \cdot 10^{-04}$	1.99

Table 2: *Advection-diffusion* convergence with (left) and without (right) overset. Similar orders of convergence and errors are observed.

From Table 2 one can conclude that the *Advection-diffusion* case is converging when refining the grid toward the exact analytical solution with an order 2. The level of the error is also similar to the simulation without overset (right column of the table). The error distribution at the end of the computation can be visualized on Figure 2, no particular artefact due to the overset assembly is visible: the error does not increase around the edges of the foreground grid. Figure 3 presents the differences between the simulations with and without overset, showing that the difference is below 10^{-10}

and are slightly impacted by the overset assembly. An *Advection* only case was also run and yielded very similar results to this *Advection-diffusion* case.



(a) G3 (b) G5

Fig. 2: Difference between exact solution and overset simulation for the *Advection-diffusion* case. The error reduces when refining the grid and no effect of the overset assembly is visible.

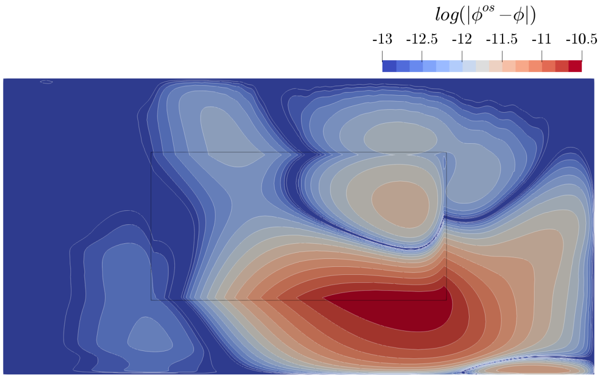
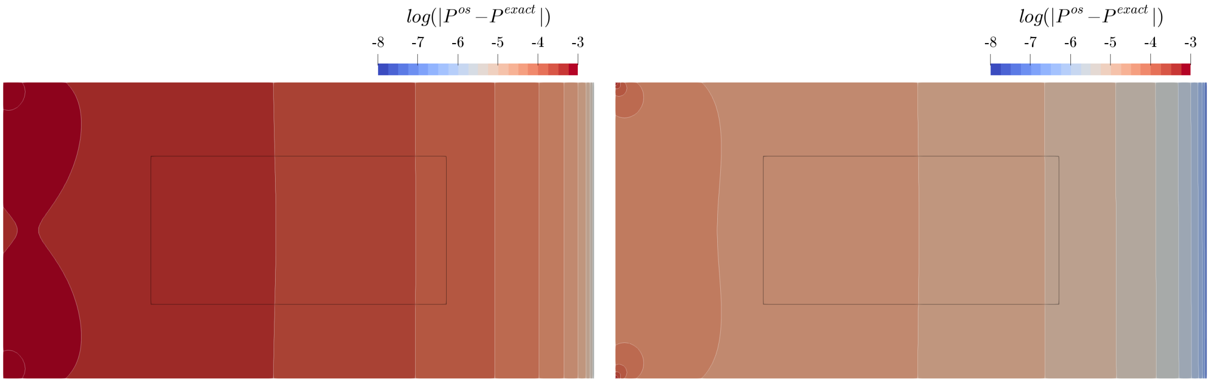


Fig. 3: Difference between with and without overset for the finest grid (G5) for the *Advection-diffusion* case. The difference is slightly impacted by the overset assembly.

	$\|\|P^{os} - P^{exact}\|\|_\infty$	order	$\|\|U^{os} - U^{exact}\|\|_\infty$	order	$\|\|P - P^{exact}\|\|_\infty$	order	$\|\|U - U^{exact}\|\|_\infty$	order
G1	$0.570951568 \cdot 10^{-01}$		$0.106355136 \cdot 10^{-01}$		$0.570951568 \cdot 10^{-01}$		$0.106355136 \cdot 10^{-01}$	
G2	$0.221288355 \cdot 10^{-01}$	1.37	$0.338702389 \cdot 10^{-02}$	1.65	$0.221290412 \cdot 10^{-01}$	1.37	$0.338700864 \cdot 10^{-02}$	1.65
G3	$0.934912126 \cdot 10^{-02}$	1.24	$0.944148824 \cdot 10^{-03}$	1.84	$0.934912218 \cdot 10^{-02}$	1.24	$0.944148756 \cdot 10^{-03}$	1.84
G4	$0.424055832 \cdot 10^{-02}$	1.14	$0.248345462 \cdot 10^{-03}$	1.93	$0.424054967 \cdot 10^{-02}$	1.14	$0.248345483 \cdot 10^{-03}$	1.93
G5	$0.201178414 \cdot 10^{-02}$	1.08	$0.637460767 \cdot 10^{-04}$	1.96	$0.201182292 \cdot 10^{-02}$	1.08	$0.637460542 \cdot 10^{-04}$	1.96

Table 3: *Poiseuille* case convergence with (left) and without (right) overset. Similar orders of convergence and errors are observed for the two sets of simulations.

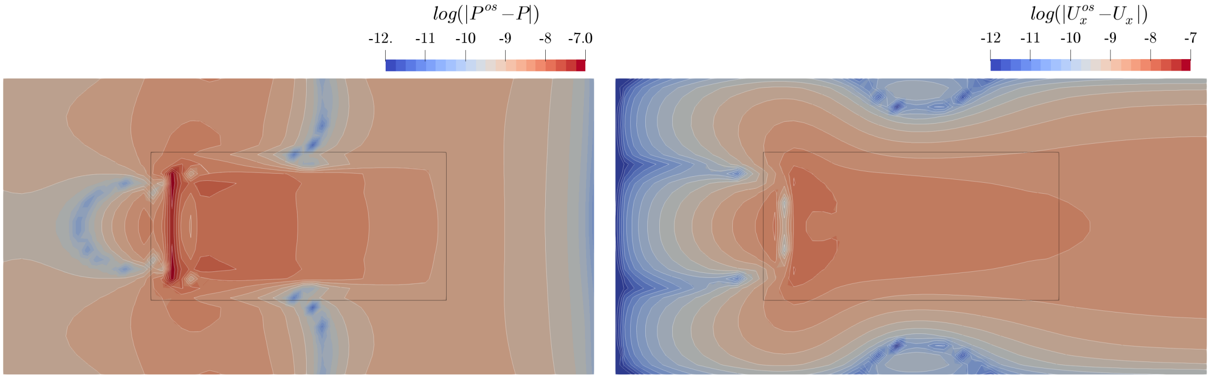
For the *Poiseuille* flow case, Table 3 shows that the order of convergence is second order for the velocity and first order for the pressure. Similarly to the *Advection-diffusion* case, comparable errors and thus convergence orders are observed when running a simulation without overset. Figure 4 displays the pressure difference between the exact solution and the overset simulation. The pressure being enforced at the outflow the error is null at that location. No variation of the error is visible around the boundary of the foreground grid. However, when comparing simulations with and without overset, some effects of the overlapping method are visible. Figure 5a shows the difference in pressure between simulations with and without overset, and a pressure variation is observed close to the inlet of the foreground mesh. The differences between the two simulations are higher in the foreground mesh when compared to the rest of the domain. Since pressure and momentum are coupled (via the SIMPLE method), a similar effect is visible on Figure 5b presenting the velocity difference between overset and non overset simulation. As presented in section 3, no particular treatment is being done at the fringe of overlapping grids to account for pressure correction. Another way of quantifying the error introduced by the overset mesh method is by assessing the mass fluxes difference between the inflow and outflow. Figure 6 presents the mass imbalance with and without overset for all grids. For both set of simulations, the mass imbalance decreases when refining the grids, going down to machine accuracy for the non overset simulations. Three to four orders of magnitude are however noticeable between the two sets of simulations. Figure 7 shows residuals convergence with and without overset simulations. Some oscillations are observed for the overset simulation, this is related to the under relaxation and the explicit coupling used. One can also note that the overset simulation does not converge slower than the simulation without overset.



(a) G3

(b) G5

Fig. 4: Pressure error for the overset simulation of the *Poiseuille* flow. The error is minimal at the outflow where it is enforced, and no effect of the overset method is visible.



(a) Pressure difference

(b) Velocity difference

Fig. 5: Pressure (left) and velocity (right) differences between simulations with and without overset for grid G2 (*Poiseuille* case). A pressure variation at the boundary of the foreground mesh leading to a velocity variation is visible.

6 Concluding remarks

An initial explicit overset implementation has been tested and results from two different cases presented here. The use of manufactured and analytical solutions permitted the quantitative assessment of the errors involved, and check the implementation for bugs. Convergence towards the exact solution is not affected by the use of the overset method. The difference between simulations with and without overset is below 10^{-10} for the *Advection-diffusion* case and below 10^{-7} for the *Poiseuille* case. When solving the continuity equation, however, pressure variations at the boundary of the overlapping area are introduced. This leads to a mass imbalance between the inflow and outflow. Quantitatively, for the *Poiseuille* test case, the mass imbalance introduced is of about four orders of magnitude higher when using overset compared to simulations without overset. To overcome this, other CFD codes usually use methods to compensate for the loss or gain of mass at the grid interfaces (Völkner et al., 2017). Future developments will focus on

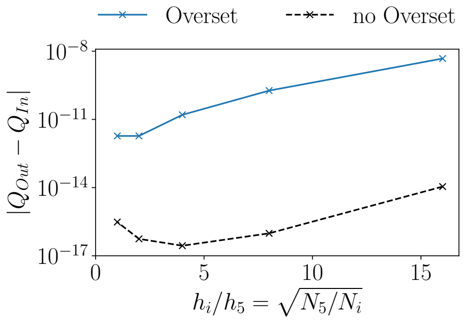


Fig. 6: Mass flux differences between the inflow and outflow of the domain for the *Poiseuille* case.

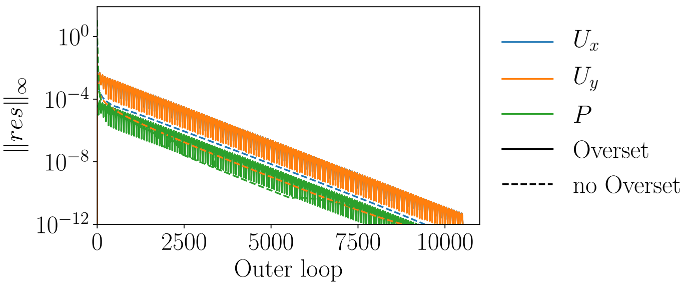


Fig. 7: Residual convergence for the finest grid (G5) for the Poiseuille test case. Oscillations related to the under relaxation are visible for the overset simulation.

various aspects. Different methods available in the literature to correct mass imbalance will be assessed. The impact of the interpolation scheme used for fringe field data will be examined. Finally, the influence of an implicit coupling toward mass conservation will also be investigated.

Acknowledgements

The authors would like to thank Dr. Ralph Noack for his help on the coupling with DiRTlib and for the computation of the DCI used in the presented test cases.

References

Benek, J., Buning, P. G., and Steger, J. (1985). A 3-D chimera grid embedding technique. In *7th Computational Physics Conference*, page 10, Reston, Virigina. American Institute of Aeronautics and Astronautics.

Brunswig, J., Manzke, M., and Rung, T. (2010). Explicit and implicit coupling strategies for overset grids. In *Proceedings of the 10th Symposium on Overset Composite Grids and Solution Strategies*, page 56. NASA AMES Research Center.

Carrica, P. M., Ismail, F., Hyman, M., Bhushan, S., and Stern, F. (2013). Turn and zigzag maneuvers of a surface combatant using a URANS approach with dynamic overset grids. *Journal of Marine Science and Technology (Japan)*, 18(2):166–181.

Hadžić, H. (2005). *Development and Application of a Finite Volume Method for the Computation of Flows Around Moving Bodies on Unstructured , Overlapping Grids*. PhD thesis.

Martin, J. E., Michael, T., and Carrica, P. M. (2015). Submarine Maneuvers Using Direct Overset Simulation of Appendages and Propeller and Coupled CFD/Potential Flow Propeller Solver. *Journal of Ship Research*, 59(1):31–48.

Noack, R. W. (2005). DiRTlib: A library to add an overset capability to your flow solver. In *17th AIAA Computational Fluid Dynamics Conference*, number June 2005, pages 1–20, Reston, Virigina. American Institute of Aeronautics and Astronautics.

Noack, R. W. and Boger, D. A. (2009). Improvements to SUGGAR and DiRTlib for Overset Store Separation Simulations. *47th AIAA Aerospace Sciences Meeting*, (January):5–9.

Toxopeus, S. and Bhawsinka, K. (2016). Calculation of Hydrodynamic Interaction Forces on a Ship Entering a Lock Using CFD. In *4th MASHCON-International Conference on Ship Manoeuvring in Shallow and Confined Water with Special Focus on Ship Bottom Interaction*, pages 305–314.

Völkner, S., Brunswig, J., and Rung, T. (2017). Analysis of non-conservative interpolation techniques in overset grid finite-volume methods. *Computers and Fluids*, 148:39–55.