Numerical improvement and validation of a naval hydrodynamics CFD solver in view of performing fast and accurate simulation of complex ship-wave interaction

CSI(Comité de Suivi Individuel) report

Kim Young Jun

Supervisor: prof. David Le Touzé

Ecole Centrale de Nantes

Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA)

Nantes, 09/2019

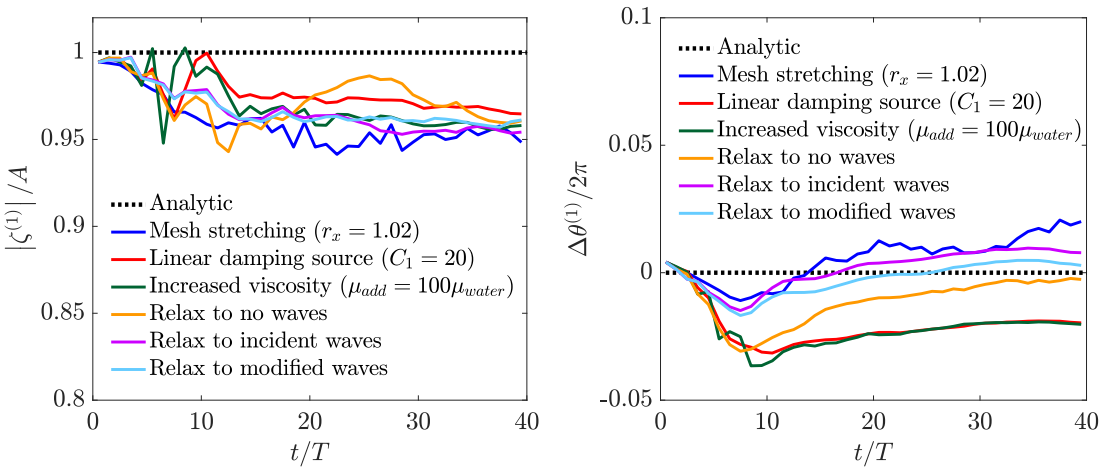
1. Summary

This article gives a summary of the ongoing PhD work done by Young Jun Kim. The objective of this PhD thesis is to have improvement (or development) of fast but accurate (efficient) CFD solver for naval application. The methodology is based on the finite volume method (FVM) and open source code package 'OpenFOAM' is used. An additional in-house code library `foamStar` is also employed. Following summary and the appendices give some explanations on the attempts to achieve efficient CFD solver for naval application.

1.1. Wave propagation & absorption

The waves are one of the most important and critical factor in naval hydrodynamic. Thus, proper methodology for generation and absorption of waves in the numerical wave tank (NWT) is the foundation of naval CFD application. Appendix 01 present a preliminary work on the NWT, and short summary of Appendix 01 is given in this section.

The relaxation scheme is a commonly used to generate or absorb the wave for the NWT simulation. Present study applied the explicit blending scheme for the generation of the waves and the absorption of the waves to incident wave. The explicit blending scheme blends the computed solution with target flow fields in relaxation zones, while, implicit scheme gives the weight and the constraints to the governing equation.



(a) Amplitude **(b)** Phase differences

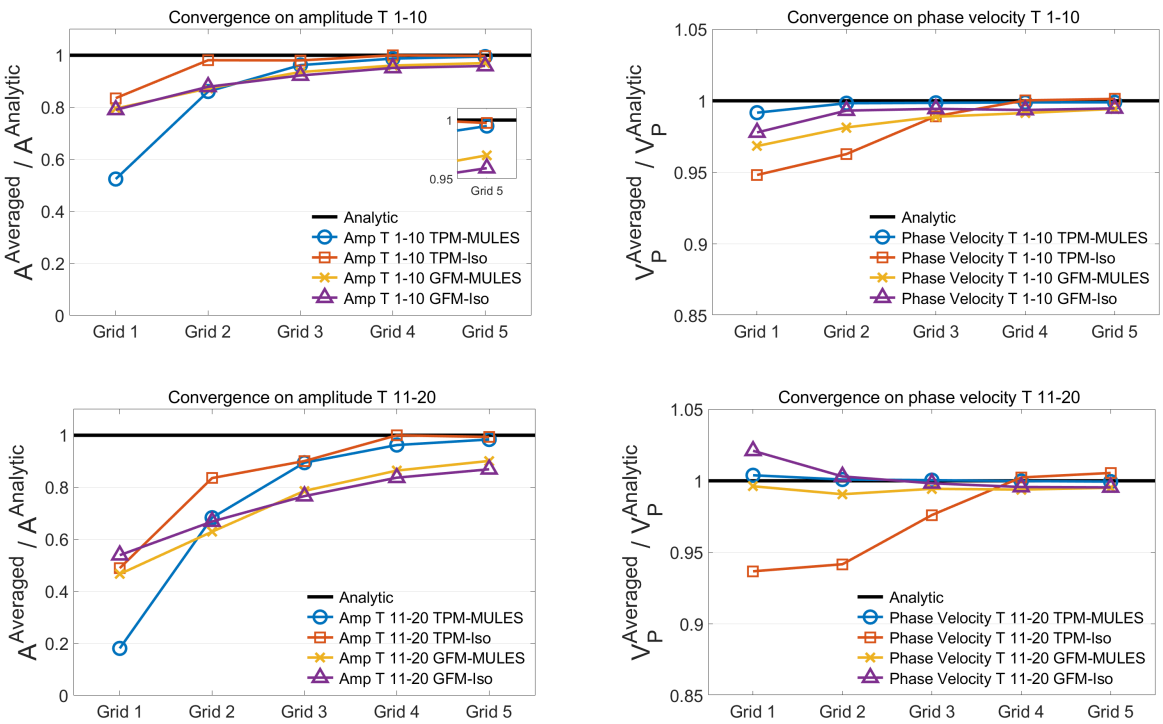
Figure 1.1. The first-order harmonic amplitudes and phase differences with respect to different outlet.

For some cases, the damping zone, which damps waves and all velocity components, has to be considered. To achieve this, linear damping source, quadratic damping source and increased artificial viscosity are considered. The quality of propagating waves are highly affected by the choice of strength of each

damping source and damping zone length. Thus, by using the reflection coefficient, the quality of waves are measured respect to the damping zone parameters.

1.2. Interface treatment schemes

Unlike one-phase flow model, incompressible two-phase flow model require proper modeling of the discontinuous properies at the interface (free-surface). Due to the existence of the interface, it is still challenging to generate the precise desired wave and to limit the dissipation in order to maintain the wave amplitude for the whole simulation. To avoid the instability and the dissipation, a proper interface treatment schemes are required. The interface treatment scheme includes interface capturing scheme, interface advection scheme and interface boundary condition. Interface capturing scheme determine how to numerically model the free-surface. Interface advection scheme define the methodology to transport the interface in time and interface boundary condition gives the free-surface boundary condition into the finite volume framework. This study reviewed some of the existing interface treatment schemes, and reported its capability for the wave propagation test case.



(a) Normailized amplitude **(b)** Phase velocity

Figure 1.2. The first-order harmonic amplitudes and phase differences with respect to different outlet.

The applied interface treatment schemes and its application results are shortly summarized. The volume of fluid (VOF) method is applied to capture the free-surface. A classical algebraic VOF convection scheme with an artificial compression source and MULES limiter (MULES) and new geometric VOF convection scheme (IsoAdvector [2]) are used for the convection of VOF. For the treatment of boundary

condition of free-surface, classical averaged two-phase model(TPM) and ghost fluid method (GFM [3]) model is considered. The combination of interface treatment schemes yields four different solvers, and its results for the wave propagation test case are illustrated at Fig. 1.2.

The result say that TPM-Iso showed best conservation of wave amplitude and this is expected since it is an explicit scheme. TPM-MULES always have moderate wave amplitude and best phase velocity. The method using GFM shows similar results regardless to its VOF convection scheme. This means that the mass flux do the most important role for the modeling of two-phase flow. GFM shows good results regardless to the resolution of mesh, but it also shows constant dissipation of wave amplitude in time regardless to mesh resolution.

1.3. Temporal discretization schemes

The application of the diagonally implicit Runge-Kutta (DIRK) scheme to two-phase incompressible solver is studied. With Implicit Runge-Kutta (IRK) temporal discretization, an arbitrary ordinary differential equation (ODE, Eqn. (1.1)) can be approximated into Eqn. (1.2) and Eqn. (1.3). Where, s is the number of Runge-Kutta iteration; h is the time-step; a_{ij} is the stage weight; b_i is the scheme weight; τ is the abscissa (sub-step) for Runge-Kutta iteration; $y(t^{(n,i)})$ is the sub-step output valid at $t^{(n)} + h\tau_{(i)}$.

$$\dot{y} = f(t, y(t)) \tag{1.1}$$

$$\begin{aligned} y(t^{(n,i)}) = y^{(i)} = y^{(n)} + ha_{ii} f(t^{(n,i)}, y(t^{(n,i)})) + h\sum_{j=1}^{i-1} a_{ij} f(t^{(n,j)}, y(t^{(n,j)})) \\ t^{(n,i)} = t^{(n)} + h\tau_{(i)} \end{aligned} \tag{1.2}$$

$$y^{(n+1)} = y^{(n)} + h\sum_{j=1}^{s} b_j f(t^{(n,j)}, y(t^{(n,j)})) \tag{1.3}$$

The coefficients of Eqn. (1.2) and Eqn. (1.3) define a Runge-Kutta method, and a set of the coefficients can be presented in a form of Butcher matrix. Table 1.1 and Table 1.2 show the families of four-stage DIRK method. These tables include sub-classes of DIRK, which are S, E and SA. The letter S denotes

Table 1.1. Four-stage DIRK, SDIRK and SDIRK_SA [5]

$$\begin{array}{c|cccc} \tau_1 & a_{11} & 0 & 0 & 0 \\ \tau_2 & a_{21} & a_{22} & 0 & 0 \\ \tau_3 & a_{31} & a_{32} & a_{33} & 0 \\ \tau_4 & a_{41} & a_{42} & a_{43} & a_{44} \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} \qquad \begin{array}{c|cccc} \tau_1 & \gamma & 0 & 0 & 0 \\ \tau_2 & a_{21} & \gamma & 0 & 0 \\ \tau_3 & a_{31} & a_{32} & \gamma & 0 \\ \tau_4 & a_{41} & a_{42} & a_{43} & \gamma \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} \qquad \begin{array}{c|cccc} \tau_1 & \gamma & 0 & 0 & 0 \\ \tau_2 & a_{21} & \gamma & 0 & 0 \\ \tau_3 & a_{31} & a_{32} & \gamma & 0 \\ 1 & a_{41} & a_{42} & a_{43} & \gamma \\ \hline & a_{41} & a_{42} & a_{43} & \gamma \end{array}$$

Table 1.2. Four-stage EDIRK, ESDIRK and ESDIRK_SA [5]

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & a_{22} & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & a_{33} & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & a_{44} \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
1 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & a_{41} & a_{42} & a_{43} & \gamma
\end{array}
$$

Singly diagonal, and the letter E denotes explicit first stage. Another constraint on the coefficients, which is both beneficial and straightforward is the stiffly-accurate (SA) assumption which imposes $a_{sj} = b_j$ and $\tau_s = 1$ [4]. The detail properties on the applied DIRK methods and its implementations are not included in the summary documents but described in detail in the Appendix 3.

The classical foamStar with Crank-Nicolson method and various DIRK-type methods are compared using several applications. Table 1.3 summarized the tested time integration schemes. Only Stiffly accurate (SA) DIRK methods are utilized, and maximum three implicit stages are considered. For the validation of each solver, periodic wave propagation test case is used. For the detail information of set-up, please refer to the appendix.

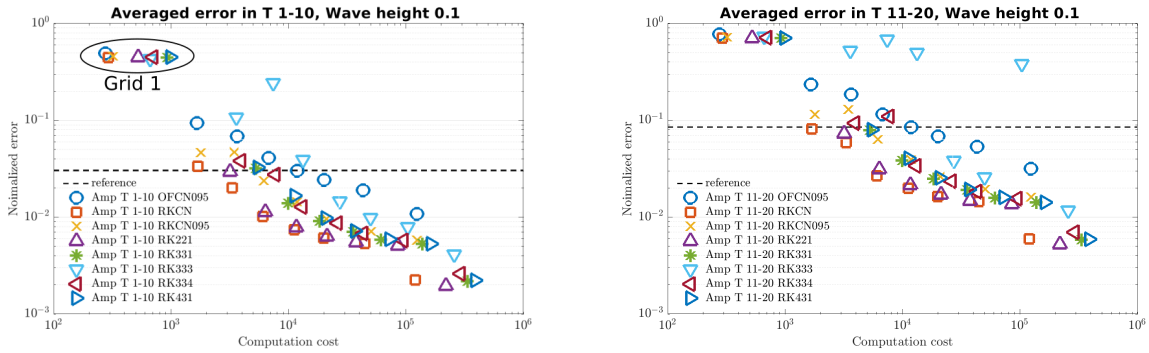
Table 1.3. Name and properties of solvers

Code name	Identification name	Order	Implicit stage	Notes
OFCN095	OpenFOAM C-N	2	1	off-centered 0.95
RK_CN	ESDIRK[2](2)_SA	2	1	-
RK_CN095	ESDIRK[2](2)_SA	2	1	off-centered 0.95
RK221	SDIRK[2](2)_S	2	2	-
RK331	SDIRK[3](3)_SA	3	3	-
RK333	EDIRK[3](3)_SA	3	2	-
RK334	ESDIRK[3](3)_SA	3	2	-
RK431	ESDIRK[3](4)_SA	4	3	-

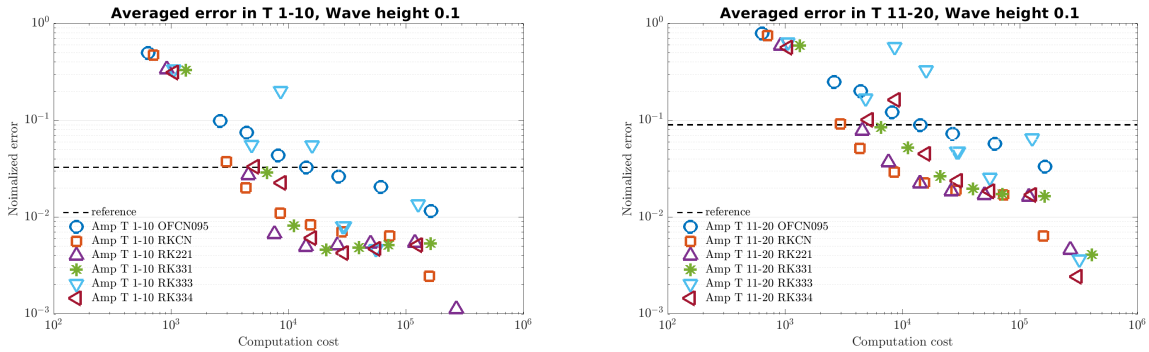
The efficiency of each temporal discretization scheme is measured using accuracy and CPU time. Fig 1.3 shows part of the wave propagation results. Subplot a and b shows the results with static mesh and dynamic mesh, respectively. The dynamic mesh motion is imposed by putting an heaving and pitching imaginary body at the free-surface. For the smooth mesh motion, the surrounding grids morph using smooth weight function. The heave and the pitch period is equal to the regular wave period, and the heave amplitude is $0.01m$ and the pitch amplitude is 1 degree. Y-axis is the normalized averaged

error in time and X-axis is the total computation time. Both axes are in log scale. For the multi-core computation, the computation time is linearly multiplied by number of core. The black doted line is the reference case. The results from solver RK333 shows unstable and unpredictable behavior, thus this is excluded for the discussion.

The acceptable range of error has determined to estimate the efficiency. Since the wave propagation is basis of all naval application, this study targeted to have around 1% of error. In this point of view, the mesh resolution of 'Grid 1' is not appropriate for the proper wave propagation. Except the condition 'Grid 1', RKCN and RK221 scheme show the best efficiency. Also, author must notice that there have large difference between OpenFOAM Crank-Nicolson and DIRK based Crank-Nicolson. This difference is not only due to the off-centering, but also due to the treatment of the mass flux. The higher-order schemes (RK331, RK334, RK431) did not showed the higher-order convergence, and these are less efficient than second order DIRK schemes. The result of subplot (b) indicate the capability of computation with moving mesh. Compare to the static mesh, the accuracy decreased and the variation of solutions are large. Considering that this periodic test case is very tough case due to the cumulative error, the resulting solutions from DIRK schemes are acceptable



(a) Static mesh



(b) Dynamic mesh

Figure 1.3. The comparison of the efficiency of the solvers, wave height H0.1

Bibliography

[1] Y. Choi, B. Bouscasse, S. Seng, G. Ducrozet, L. Gentaz, P. Ferrant, Generation of regular and irregular waves in navier-stokes cfd solvers by matching with the nonlinear potential wave solution at the boundaries, International Conference on Offshore Mechanics and Arctic Engineering Volume 2: CFD and FSI (2018) OMAE2018–78077.

[2] J. Roenby, H. Bredmose, H. Jasak, IsoAdvector: Geometric VOF on general meshes, Selected papers of the 11th Workshop, 2017.

[3] V. Vukčević, Numerical modelling of coupling potential and viscous flow for marine applications, Ph.D. thesis, University of Zagreb (2016).

[4] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, mathematics of computation 28 (125) (1974) 145–162.

[5] C. A. Kennedy, M. Carpenter, Diagonally implicit runge-kutta methods for ordinary differential equations. a review.

Appendix 01
Wave propagation & absorption

CSI(Comité de Suivi Individuel) report

Kim Young Jun

Supervisor: prof. David Le Touzé

Ecole Centrale de Nantes

Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA)

Nantes, 09/2019

1. Wave generation & absorption

The wave propagation in a viscous flow is one of the important part of naval computational fluid dynamic (CFD) applications. However, in the two-phase finite volume method (FVM) framework, it is still challenging to generate the precise desired wave and to limit the dissipation in order to maintain the wave amplitude for the whole simulation. And when this is achieved, the good energy conservation may lead to instabilities in the flow disturbing the correctness of the solution.

The generation and absorption of waves in `foamStar` is based on explicit relaxation schemes which blend the computed solution with target flow fields in the relaxation zones [1, 2, 3, 4, 5]. The explicit relaxation scheme is applied in this study which uses a weight function which varies between 0 and 1 in the relaxation zone. After solving the governing equations, the computed flows are relaxed with target flow values as

$$\chi = (1 - w)\chi + w\chi^{Target}, \tag{1.1}$$

where χ is the computed flow, w is a weight function and χ^{Target} is a target flow.

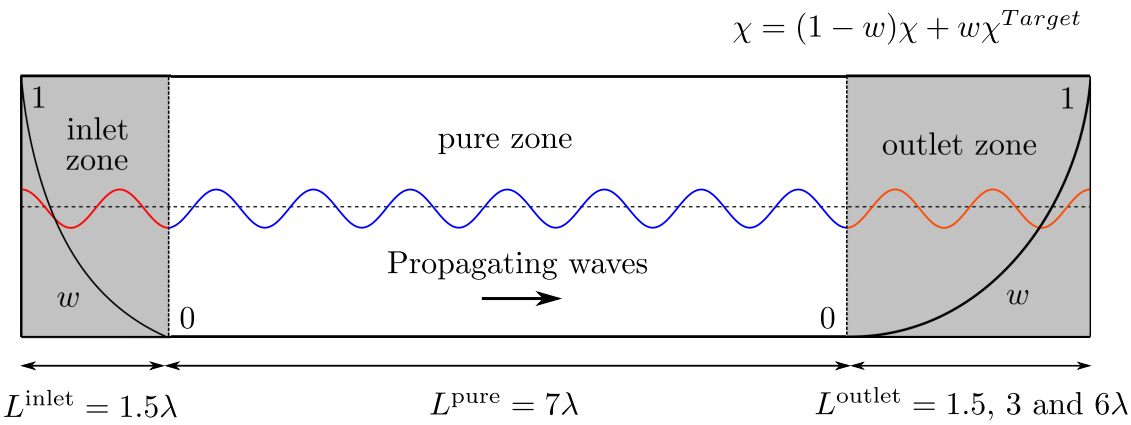


Figure 1.1. Schematic view of the NWT for a parametric study on the relaxation schemes.

1.1. Mesh stretching outlet

As the stretched mesh schemes differ as to the cases and to the researchers [6, 7], it is difficult to find a standard for wave propagation. Therefore, the cell length is stretched with a ratio (r_x) of an adjusted cell which is located at upstream as

$$\Delta x_n = r_x \Delta x_{n-1} \tag{1.2}$$

where Δx_n and r_x represents a longitudinal length of the n-th cell and the stretching ratio, respectively. To find a representative result of stretched mesh outlet, the different stretching ratios are considered as

$$r_x = \begin{cases} c, & \text{constant stretching; } c = 1.02,\ 1.05 \text{ and } 1.08 \\ \cosh(\gamma n), & \text{exponential stretching; } \gamma = 1.001 \end{cases} \tag{1.3}$$

The outlet meshes are stretched until the length of the last three cells is larger than one wavelength ($\Delta x \geq 1\lambda$).

1.2. Damping layer outlet

Damping layer method, which is also called as 'sponge layer method', is considered here [8]. The linear and quadratic damping source terms are commonly used to damp waves [9, 10]. Shen and Wan (2016) utilized a linear damping source to absorb waves [11]. The momentum equations of multiphase flow with the damping source terms are given as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}^T) - \nabla \cdot \left[\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] = -\nabla p_d - (\mathbf{g} \cdot \mathbf{x})\nabla \rho + \mathbf{S} \tag{1.4}$$

where μ is a dynamic viscosity, p_d is the dynamic pressure and $\mathbf{S}$ is the linear and quadratic damping source. The damping sources are only applied in vertical velocity components as

$$\mathbf{S} = [0, 0, \rho(C_1 + C_2 u_z) w u_z]^T \tag{1.5}$$

of which C_1 and C_2 are the linear and quadratic damping coefficients and w is weight function. The parametric study on the damping coefficients conducted by Perić and Abdel-Maksoud(2016) showed that the optimal choice of coefficients could minimize wave reflections [10]. Recently, Perić et al. (2018) proposed the method evaluating the wave reflection coefficients by linear damping coefficient and the outlet length. Furthermore, they proposed the method calculating the linear damping coefficient, which minimizes the reflection coefficient. Though, no waves is used for target flows in the relaxation zone, the linear damping coefficient, having the minimum reflection coefficient with outlet length 1.5λ, is around $C_1 \approx 7.5$. In the present study, the linear damping coefficients are considered due to the complexity of combination.

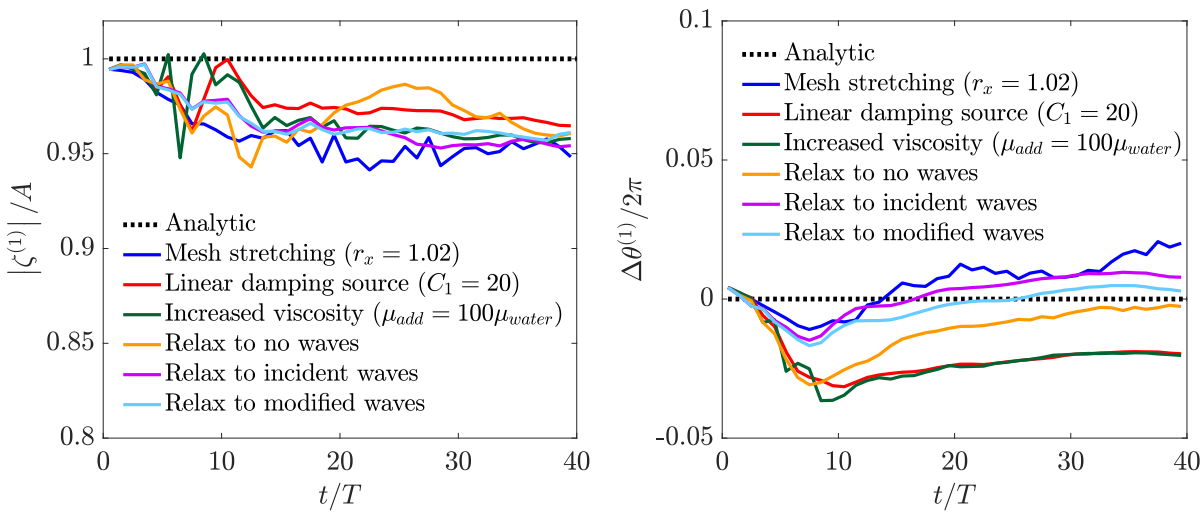
The viscosity can be increased artificially in the outlet zone to damp the waves. The momentum equation with increased viscosity in the outlet is given as

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}^T) - \nabla \cdot \left[(\mu + \mu_{add}) \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T\right)\right] = -\nabla p_d - (\mathbf{g} \cdot \mathbf{x})\nabla \rho \tag{1.6}$$

of which μ_{add} is the increased viscosity in the outlet zone. Also, the mesh stretched outlet, which utilizes the numerical damping by stretching meshes, is considered in the present study.

2. Results

The mesh stretching, linear damping source, increased viscosity, relaxation to no waves, incident waves and modified waves are compared for different wave outlets. The limited outlet length of $L^{outlet} = 1.5\lambda$, except for mesh stretching, is only considered. The representative results from the parametric study on the each wave outlet are taken for comparison. The first order harmonic amplitudes and phase differences with respect to the different outlet conditions are compared in figure 2.1.



(a) Amplitude **(b)** Phase differences

Figure 2.1. The first-order harmonic amplitudes and phase differences with respect to different outlet.

The measured first harmonic amplitudes during 40 wave periods have an errors range of 7%. Amplitude modulations over simulation time are observed for all cases, and stable results are obtained for relaxation to incident waves and modified waves. Waves are propagating with a constant phase difference after a certain time of simulation for linear damping source and increased viscosity outlets. The phase differences of relaxation to the incident and modified waves are smaller than other boundary conditions. Relax to no waves slowly converges to the analytic solution. The reflection coefficients for the different outlets during simulation are plotted in figure 2.2.

At the initial simulation stage, the high reflection coefficients are measured for linear damping outlet, increased viscosity, and relax to no waves. It is thought that the different outlet configurations such as different viscosity quantities, artificial sources, and no flow, affect the flow fields immediately due to the limited length of the outlet. After an initial disturbance, the wave reflection with linear damping source and increased viscosity decreases, and has a similar reflection with relaxation to waves from $t = 22T$. Meanwhile, the relax to no waves shows a higher

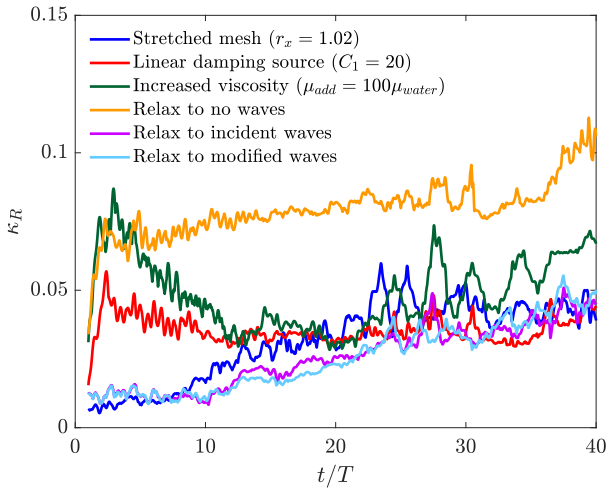


Figure 2.2. Evolution of reflection coefficients for different outlets.

wave reflection over simulation time after initial disturbance. It shows that blending to wrong target function in the relaxation zone do not promise a good result. The mesh stretching outlet and relaxation to the incident and the modified waves have gradually increasing wave reflection over simulation time. The time averages of wave reflection coefficients during the simulation time $t \in 15T, 20T$ for different outlets are tabulated in Table 2.1.

Table 2.1. Time averaged reflection coefficients with respect to different outlets for $t \in [15T, 20T]$.

Outlet	κ_R
Stretched mesh ($r_x = 1.02$)	0.0312
Linear damping source ($C_1 = 20$)	0.0334
Increased viscosity ($\mu_{add} = 100\mu_{water}$)	0.0357
Relax to no waves	0.0788
Relax to incident waves	0.0225
Relax to modified waves	0.0192

The smallest wave reflections are obtained for relaxation to modified waves, and the other outlet conditions also give good results. Proper tuning of linear damping source and increased viscosity would give probably better results. Nevertheless the parametric study of these outlets needs more work than relaxation schemes, and it is difficult to understand a physical meaning of tuned parameter. To conclude, the comparison of different outlets in the case of propagating waves confirms that the outlet conditions used in viscous flow solver give good results if they are properly tuned.

Bibliography

[1] S. Mayer, A. Garapon, L. S. Sørensen, A fractional step method for unsteady free-surface flow with applications to non-linear wave dynamics, Int. J. for Num. Methods in Fluids 28 (1998) 295–315.

[2] D. R. Fuhrman, Numerical solutions of Boussinesq equations for fully nonlinear and extremely dispersive water waves, Ph.D. thesis, Technical University of Denmark, Kgs. Lyngby (2004).

[3] A. P. Engsig-Karup, Unstructured nodal DG-FEM solution of high-order Boussinesq-type equations, Ph.D. thesis, Technical University of Denmark, Kgs. Lyngby (2006).

[4] N. Jacobsen, D. Fuhrman, J. Fredsøe, A wave generation toolbox for the open-source CFD library: OpenFoam, Int. J. for Num. Methods in Fluids 70 (9) (2012) 1073–1088.

[5] S. Seng, Slamming and whipping analysis of ships, Ph.D. thesis, Technical University of Denmark (2012).

[6] C. Monroy, G. Ducrozet, F. Bonnefoy, A. Babarit, L. Gentaz, P. Ferrant, Rans simulations of a calm buoy in regular and irregular seas using the swense method, International Journal of Offshore and Polar Engineering 21.

[7] R. Perić, M. Abdel-Maksoud, Assessment of uncertainty due to wave reflections in experiments via numerical flow simulations 2015 (2015) 530–537.

[8] M. Israeli, S. Orszag, Approximation of radiation boundary conditions, J. of Comp. Physics 41 (1) (1981) 115–135.

[9] J.-C. Park, M.-H. Kim, H. Miyata, Fully non-linear free-surface simulations by a 3D viscous numerical wave tank, Int. J. for Num. Methods in Fluids 29 (6) (1999) 685–703.

[10] R. Perić, M. Abdel-Maksoud, Reliable damping of free-surface waves in numerical simulations, Ship Tech. Research 63 (1) (2016) 1–13.

[11] Z. Shen, D. Wan, An irregular wave generating approach based on naoe-FOAM-SJTU solver, China Ocean Eng. 30 (2) (2016) 177–192.

Appendix 02
Interface treatment schemes

CSI(Comité de Suivi Individuel) report

Kim Young Jun

Supervisor: prof. David Le Touzé

Ecole Centrale de Nantes

Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA)

Nantes, 09/2019

1. Introduction

The wave propagation in a viscous flow is one of the important part of naval computational fluid dynamic (CFD) applications. However, in the two-phase finite volume method (FVM) framework, it is still challenging to generate the precise desired wave and to limit the dissipation in order to maintain the wave amplitude for the whole simulation. And when this is achieved, the good energy conservation may lead to instabilities in the flow disturbing the correctness of the solution. Furthermore, when computing the practical order of convergence in space and time in a wave propagation problem, even if the dissipation and stability of the solution is satisfactory, the order of convergence achieved is often less than the expected theoretical values. The main reason for all these difficulties is the necessity of free-surface modeling. As an interface treatment is required to reflect the discontinuous property of the free-surface (or the multiphase interface) in the numerical computation.

This paper aims to simulate two-dimensional wave propagation periodic problem using different interface treatment schemes. The theories on the interface treatment scheme is presented in section II. Section III and IV present respectively the simulation set-up of periodic wave propagation and its analysis. The base viscous solver is open source library package OpenFOAM (Version 5.0). Additional numerical implementations are from the library package 'foamStar' which is the software developed with Bureau Veritas (BV) with collaboration with Ecole Cnetrale de Nantes.

2. Mathematical model

2.1. Typical two-phase flow model

In this section few mathematical models for incompressible two-phase flow are presented. The typical incompressible two-phase flow model is derived based on conditional averaging of two fluid. Following momentum and continuity equations is:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u}) - \nabla\bullet(\mu(\nabla\mathbf{u})) - \nabla\mathbf{u}\bullet\nabla\mu = -\nabla p_d - (\mathbf{g}\bullet\mathbf{x})\nabla\rho + \sigma_\kappa\nabla\alpha\,, \tag{1}$$

$$\nabla\bullet\mathbf{u} = 0\,, \tag{2}$$

where, $\mathbf{u}$ is the continuous velocity field; ρ is continuous density field which has spatial variation due to two fluids ($\rho_{air} = 1.0$ Kg/m^2, $\rho_{water} = 1000$ Kg/m^2); μ is averaged dynamic viscosity ($\mu_{air} = 1.0 \times 10^{-5}$ Ns/m^2, $\mu_{air} = 1.0 \times 10^{-3}$ Ns/m^2); p_d is dynamic pressure; σ_κ is surface tension

coefficient and considered negligible ($\sigma_\kappa = 0$) in this paper. The finite volume (FV) representation of Eq.Eqn. (1) is:

$$\frac{\partial(V_P \rho \mathbf{u})}{\partial t} + \sum_f (\rho\phi_f \mathbf{u}) - \sum_f (\mu(\nabla \mathbf{u})) - \nabla \mathbf{u} \bullet \nabla \mu V_P = -\nabla p_d V_P - (\mathbf{g} \bullet \mathbf{x}) \nabla \rho V_P \,. \tag{3}$$

Where, P refers to owner cell's properties; V_P is the volume of cell P; $\rho\phi_f$ is the mass flux of a faces f constructing the cell P. The coupled equations Eq.Eqn. (3) and Eq.Eqn. (2) are solved with PISO algorithm ([1], [2]) and explained detail in [3] and [4]. Note that density ρ and mass flux $\rho\phi_f$ should be calculated before the PISO loop.

The VOF method define the phase indicator function α as a volume fraction of water in a cell. Thus, $\alpha = 1$ at fully submerged cell and $\alpha = 0$ in the air. From mass conservation equation, the VOF convection equation can be derived as:

$$\frac{\partial(\alpha)}{\partial t} + \nabla \bullet (\mathbf{u}\alpha) + \nabla \bullet (\mathbf{u}_r \alpha(1-\alpha)) = 0 \,. \tag{4}$$

Where, $\mathbf{u}_r$ is relative velocity normal to the free-surface and the third term of LHS is an artificial compression term ([5, 6]) which compress the VOF field to minimize the smearing.

2.2. Compressive VOF convection algorithm

The typical VOF convection algorithm which is also on the OpenFoam embedded solver interFoam (incompressible, Newtonian two-phase flow solver) is presented. The Eq.Eqn. (4) can be transformed into FV representation with spatial discretization and off-centered Crank Nicolson temporal discretization:

$$\begin{aligned} \frac{(V\alpha)^{t+h} - (V\alpha)^t}{h} + C_{CN} \sum_f [F_f + F_{rf}]^{t+h} + (1 - C_{CN}) \sum_f [F_f + F_{rf}]^t = 0 \,, \\ F_f = \phi_f \alpha_f \quad F_{rf} = \phi_{rf} \alpha_f (1 - \alpha_f) \,, \end{aligned} \tag{5}$$

The value C_{CN} is a Crank Nicolson coefficient where $C_{CN} = 0.5$ yields classical Crank Nicolson scheme and $C_{CN} = 1$ yields Backward Euler scheme.

To solve the bounded and non-smeared VOF field, Eq.Eqn. (5) is separated into two equations: 'Predictor step Eq.Eqn. (6)' and 'Corrector step Eq.Eqn. (7)'.

$$\frac{(V\alpha)^* - (V\alpha)^t}{h} + C_{CN} \sum_f [F_{f,upwind}]^{t+h} + (1 - C_{CN}) \sum_f [F_{f,upwind}]^t = 0 \,. \tag{6}$$

$$\frac{(V\alpha)^{t+h} - (V\alpha)^*}{h} + C_{CN} \sum_f [\lambda F_{Correct}]^{t+h} + (1 - C_{CN}) \sum_f [\lambda F_{Correct}]^t = 0 \,. \tag{7}$$

$$
\begin{aligned}
F_{Correct} &= (F_{f,HighOrder} + F_{rf,HighOrder} - F_{f,upwind}) \\
F_{Total} &= F_{f,upwind} + \lambda F_{Correct} \,.
\end{aligned}
\tag{8}
$$

Here, α^* is a intermediate VOF field and λ is a limiter evaluated from MULES(Multidimensional Universal Limiter for Explicit Solver, [3]) which limits the total flux and guarantees the boundedness of the VOF field. The continuous density field and the mass flux are evaluated for PISO algorithm.

$$
\rho = \alpha\rho_{water} + (1-\alpha)\rho_{air} \,, \qquad (\rho\phi)_f = \rho_{water}F_{Total} + \rho_{air}(\phi_f - F_{Total}) \,. \tag{9}
$$

2.3. VOF convection with IsoAdvector method

The algorithm of geometric VOF scheme 'IsoAdvector' method is presented in this subsection. Instead of solving Eq.Eqn. (5) in algebraic way, Roenby et al.([7]) developed a new VOF convection method which uses the isosurface of a cell. From the mass conservation equation, the following equation is satisfied for each cell i:

$$
\alpha_i(t+h) = \alpha_i(t) - \frac{1}{V_i}\sum_{faces=F_j}\int_t^{t+h}\int_{F_j} H(\mathbf{x},\tau)\mathbf{u}(\mathbf{x},\tau)d\mathbf{S}d\tau \,. \tag{10}
$$

Here, $H(\mathbf{x},t)$ is a phase indicator function at a point and satisfies:

$$
H(\mathbf{x},\tau) = \frac{\rho(\mathbf{x},\tau) - \rho_{air}}{\rho_{water} - \rho_{air}} \,, \qquad \alpha_i = \frac{1}{V_i}\int_{V_i} H(\mathbf{x},\tau)dV \,. \tag{11}
$$

The physical meaning of the integration on LHS of Eq.Eqn. (10) is that the total volume of water moves across the face j of cell i during the time interval $[t, t+h]$. To simplify the integral, constant face velocity during the time interval is assumed (Eq.Eqn. (13)).

$$
\Delta V_{i,j}(t,h) = \int_t^{t+\Delta t}\int_{F_j} H(\mathbf{x},\tau)\mathbf{u}(\mathbf{x},\tau)d\mathbf{S}d\tau \,, \tag{12}
$$

$$
\Delta V_{i,j}(t,h) \approx \frac{0.5(\phi_j(t) + \phi_j(t+h))}{|\mathbf{S}_j|}\int_t^{t+\Delta t} A_j(\tau)d\tau \,. \tag{13}
$$

$$
A_j(\tau) = \int_{F_j} H(\mathbf{x},\tau)d\mathbf{S} \,. \tag{14}
$$

The $A_j(\tau)$ is equal to the time series of submerged area of each face j of cell i. The main objective of IsoAdvector scheme is to calculate the time integration of $A_j(\tau)$. To calculate the time variation of submerged area, IsoAdvector scheme construct the initial isosurface in cell i. The constructed isosurface moves with constant velocity and evaluate the change of submerged area in mesh level.

The detail algorithm on isosurface construction, isosurface advection and bounding procedures are explained in [7, 8, 9].

2.4. Two-phase flow model with ghost fluid method

Instead of using averaged two-phase momentum equation (Eq.Eqn. (1)), another two-phase flow model taking into account the discontinuous properties at the interface with Ghost Fluid Method (GFM) is considered ([10], [11]). Two-phase flow is modeled like a single phase flow (Eq.Eqn. (15)), and the discontinuous dynamic pressure and density at the free-surface are considered inside the gradient and Laplacian evaluation algorithm.

$$\frac{\partial(\mathbf{u})}{\partial t} + \nabla\bullet(\mathbf{uu}) - \nabla\bullet(\nu(\nabla\mathbf{u})) - \nabla\mathbf{u}\bullet\nabla\mu = -\frac{1}{\rho}\nabla p_d\,, \tag{15}$$

$$\nabla\bullet\mathbf{u} = 0\,, \tag{16}$$

The free-surface jump conditions for dynamic pressure and density are given as Eq.Eqn. (17) and Eq.Eqn. (18), where the operator $[-]$ stands for the jump of quantities at the interface.

$$[p] = 0\,, \qquad [p_d] = -[\rho](\mathbf{g}\bullet\mathbf{x})\,, \tag{17}$$

$$[\frac{1}{\rho}p_d] = 0\,. \tag{18}$$

Further details on derivations and implementations on the GFM can be found at [11] and [12].

3. Set-up

Four different combination of solvers based on the mathematical models presented in previous section are investigated. For convenience, each solver are named 'Type A1 and A2' and 'Type B1 and B2' and selected schemes are tabulated in Table 1.

3.1. Simulation set-up

To investigate the ability of each solver in naval applications, a nonlinear wave propagating to positive x direction in two-dimensional domain is considered. The initial wave velocities and the free-surface positions are evaluated with stream function wave theory [13]. The wave conditions used in previous study ([14]) are applied again and tabulated in Table 2.

Table 1. Definition of solvers

Solver	VOF solver	Momentum solver
Type A1	Compressive VOF	Averaged two-phase
Type A2	IsoAdvector	Averaged two-phase
Type B1	Compressive VOF	GFM
Type B2	IsoAdvector	GFM

The computational domain is exactly one wave length in x-direction. Figure 1 shows the computational domain, boundary conditions and initial VOF field. The periodic boundary condition is applied to 'Inlet' and 'Outlet' boundaries. For the 'bottom' patch, slip boundary condition is applied. For the 'Air' patch, open air boundary condition is applied. To measure the wave, 100 wave probes are installed uniformly from 'Inlet' to 'Outlet' boundary.

Second order Crank Nicolson temporal discretization scheme is used for all solver (Table 1). Due to the stability reason, solver Type A1 and Type A2 used Crank Nicolson discretization with off-centering parameter $C_{oc} = 0.95$ which yields Crank Nicolson coefficient $C_{CN} = 1/(1+C_{oc})$. In contrast, other solvers used classical Crank Nicolson scheme ($C_{oc} = 1.0$). The convection term of Compressive VOF equation is discretized with second order accurate vanLeer flux limiter [15]. The convection term in the momentum equation is discretized with second order upwind biased linear scheme, and diffusion terms are discretized with central differencing scheme. Also, all gradient terms are discretized with central difference scheme. To minimize the temporal uncertainty, 8 outer (SIMPLE) and 2 inner (PISO) correctors are used. For all variables the residuals were lower than 10^{-7}.

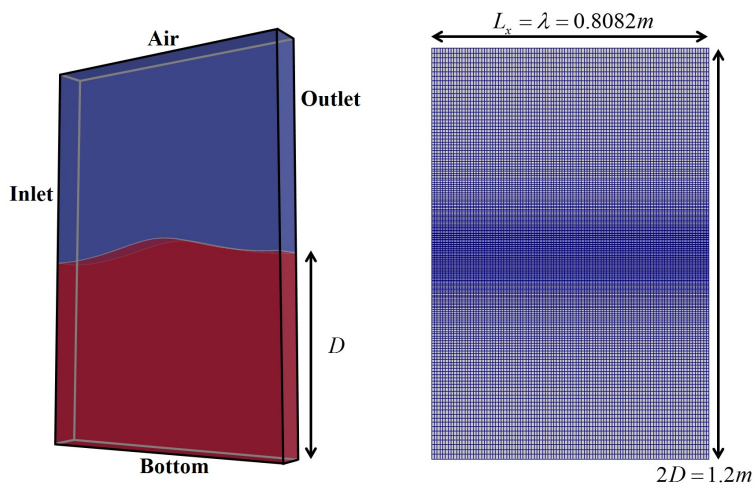


Figure 1. Computational domain and boundary conditions

Table 2. Wave condition

Item	Unit	Value
Depth (D)	[m]	0.6
Period (T)	[s]	0.7018
Height (H)	[m]	0.05753
Wave length (λ)	[m]	0.8082
Wave stiffness (kA)		0.24

3.2. Convergence study set-up

To check the convergence of each solver, a campaign of simulation with five different grids is carried out with similar Courant number. The information on the grid and time step (Δt) are tabulated in Table 3. The refinement factor is $r = 2$ for each level of grid except for the coarsest grid. The uncertainty assessment has performed with the open access tool developed by Eça and Hoekstra ([16]).

Table 3. Mesh for the convergence study (Courant number ≈ 0.17)

Case	$\lambda/\Delta x$	$H/\Delta z$	$T/\Delta t$	$\Delta x/\Delta z$
Grid 1	15	3	50	2.8
Grid 2	25	5	100	2.8
Grid 3	50	10	200	2.8
Grid 4	100	20	400	2.8
Grid 5	200	40	800	2.8

4. Results and Discussion

This section gives the result of convergence study on four different type of solvers (Table 1). Figure 2 shows the wave profile captured at time $t = 10T, t = 20T, t = 30T$ and $t = 40T$. Figure 3 and Figure 4 present the time averaged first harmonic wave amplitude and phase velocity respect to the refinement level of the mesh (Table 3). The order of convergence P and uncertainty U is evaluated for each type of solver and each time averaged amplitude. No uncertainty assessment

has performed for the phase velocity. The summarized order of convergence and uncertainties respect to type of solver and time are arranged on the Table 4.

The averaged amplitude from the solvers using averaged two-phase momentum equation (Type A1 and Type A2) show the order of convergence larger than 1 and during the first 10 periods second order convergence is achieved. The 'Type A2' (IsoAdvector) solver gives better conservation of amplitude than 'Type A1' (Compressive VOF) solver. However, the 'Type A2' solver makes wiggles on the interface for the simulation longer than 10 period. Also 'Type A2' solver shows that this solver require at least 100 cell for one wave length to obtain proper phase velocity. But it shows the increase of phase velocity after 10 period.

The solvers using GFM algorithm (Type B1 and Type B2) show lower order of convergence compare to the solver Type A1 and Type A2. The solver Type B1 and Type B2 both showed the convergence for phase velocity but the converged phase velocity is smaller than the analytic one. The solver Type B1 and Type B2 both show a similar large dissipation of wave amplitude in time and this requires more investigations.

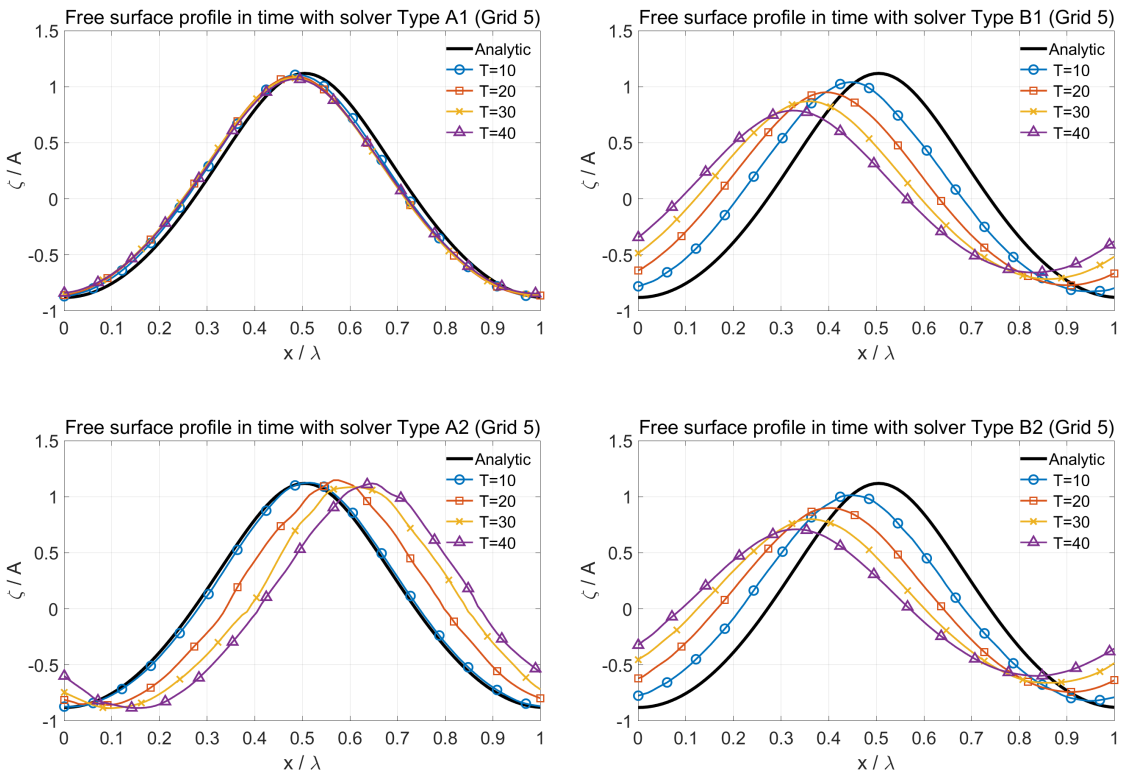


Figure 2. Free surface profile in time with Grid 5

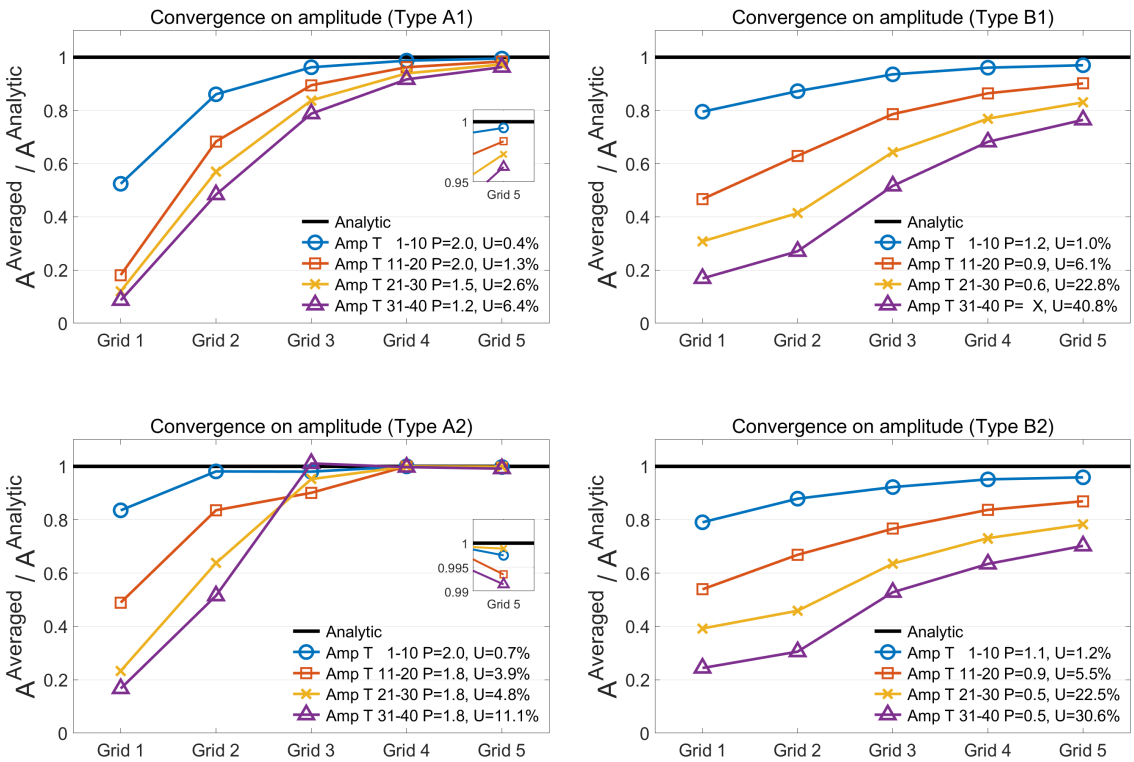


Figure 3. Convergence study on time averaged amplitude

Table 4. Summary of order of convergence and uncertainties of amplitude

Time	Type A1	Type A2	Type B1	Type B2
Periods 01-10	$P = 2.0$	$P = 2.0$	$P = 1.2$	$P = 1.1$
	$U = 0.4\%$	$U = 0.7\%$	$U = 1.0\%$	$U = 1.2\%$
Periods 11-20	$P = 2.0$	$P = 1.8$	$P = 0.9$	$P = 0.9$
	$U = 1.3\%$	$U = 3.9\%$	$U = 6.1\%$	$U = 5.5\%$
Periods 21-30	$P = 1.5$	$P = 1.8$	$P = 0.6$	$P = 0.5$
	$U = 2.6\%$	$U = 4.8\%$	$U = 22.8\%$	$U = 22.5\%$
Periods 31-40	$P = 1.2$	$P = 1.8$	$P = X$	$P = 0.5$
	$U = 6.4\%$	$U = 11.1\%$	$U = 40.8\%$	$U = 30.6\%$

ACKNOWLEDGMENT

This work has been performed in the framework of the Chaire Hydrodynamique et Structure Marines CENTRALE NANTES - BUREAU VERITAS.

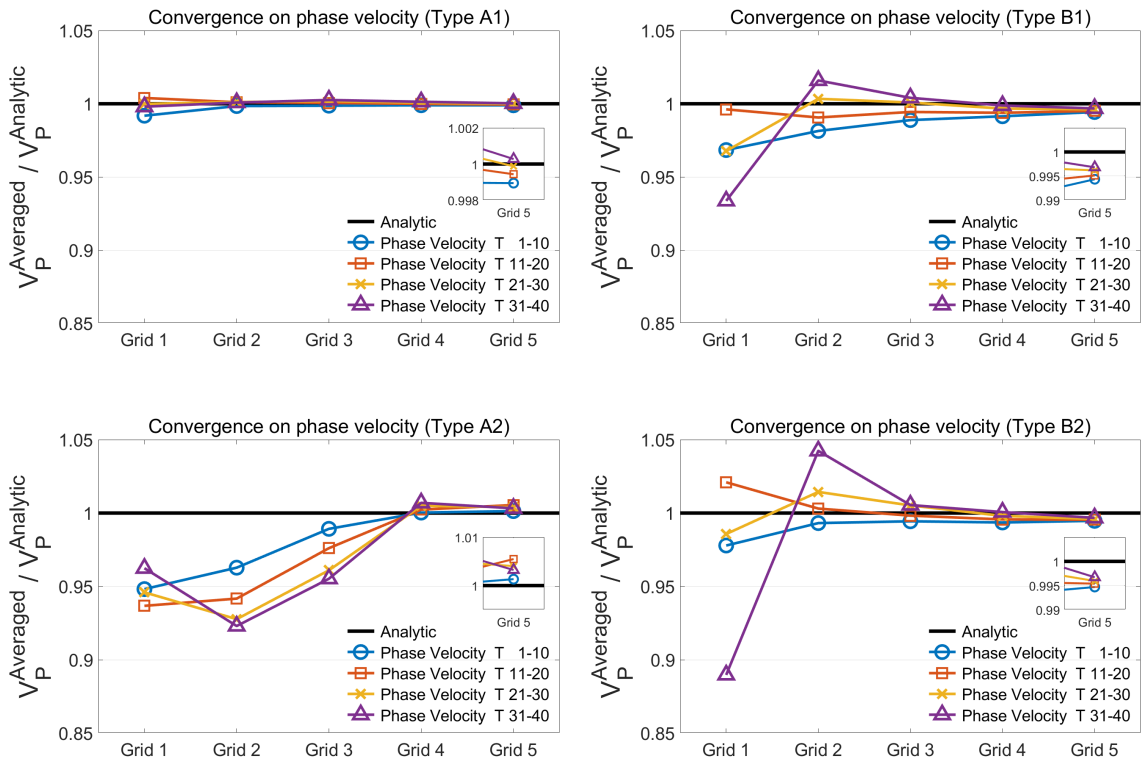


Figure 4. Convergence study on time averaged phase velocity

Bibliography

[1] R. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of Computational Physics 62 (1) (1986) 40 – 65. doi:https://doi.org/10.1016/0021-9991(86)90099-9.
URL http://www.sciencedirect.com/science/article/pii/0021999186900999

[2] S. V. Patankar, Numerical heat transfer and fluid flow, Series on Computational Methods in Mechanics and Thermal Science, Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group), 1980.
URL http://www.crcpress.com/product/isbn/9780891165224

[3] S. Márquez Damián, An extended mixture model for the simultaneous treatment of short and long scale interfaces, Ph.D. thesis, Universidad Nacional del Litoral, Santa Fe, Argentina (2013).

[4] S. Seng, Slamming and whipping analysis of ships, Ph.D. thesis, Technical University of Denmark (2012).

[5] R. Henric, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College of Science, Technology and Medicine (2002).

[6] E. Berberović, N. P. van Hinsberg, S. Jakirlić, I. V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, Phys. Rev. E 79 (2009) 036306. doi:10.1103/PhysRevE.79.036306.
URL https://link.aps.org/doi/10.1103/PhysRevE.79.036306

[7] J. Roenby, H. Bredmose, H. Jasak, A computational method for sharp interface advection, Royal Society Open Science 3 (2016) 160405. arXiv:1601.05392, doi:10.1098/rsos.160405.

[8] J. Roenby, H. Bredmose, H. Jasak, IsoAdvector: Geometric VOF on general meshes, Selected papers of the 11th Workshop, 2017.

[9] J. Roenby, IsoAdvector, https://github.com/isoadvector.

[10] J. Huang, P. M. Carrica, F. Stern, Coupled ghost fluid/two-phase level set method for curvilinear body-fitted grids, International Journal for Numerical Methods in Fluids 55 (9) 867–897. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.1499, doi:10.1002/fld.1499.
URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1499

[11] V. Vukčević, Numerical modelling of coupling potential and viscous flow for marine applications, Ph.D. thesis, University of Zagreb (2016).

[12] V. Vukčević, H. Jasak, I. Gatin, Implementation of the ghost fluid method for free surface flows in polyhedral finite volume framework, Computers & Fluids 153 (2017) 1 – 19. doi:https://doi.org/10.1016/j.compfluid.2017.05.003.
URL http://www.sciencedirect.com/science/article/pii/S0045793017301640

[13] M. M. Rienecker, J. D. Fenton, A fourier approximation method for steady water waves, Journal of Fluid Mechanics 104 (1981) 119–137. doi:10.1017/S0022112081002851.

[14] Y. Choi, B. Bouscasse, S. Seng, G. Ducrozet, L. Gentaz, P. Ferrant, Generation of regular and irregular waves in navier-stokes cfd solvers by matching with the nonlinear potential wave solution at the boundaries, International Conference on Offshore Mechanics and Arctic Engineering Volume 2: CFD and FSI (2018) OMAE2018–78077.

[15] B. V. Leer, Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection, Journal of Computational Physics 23 (3) (1977) 276 – 299. doi:https:

//doi.org/10.1016/0021-9991(77)90095-X.
URL http://www.sciencedirect.com/science/article/pii/002199917790095X

[16] L. Eça, M. Hoekstra, A procedure for the estimation of the numerical uncertainty of cfd calculations based on grid refinement studies, Journal of Computational Physics 262 (2014) 104 – 130. doi:https://doi.org/10.1016/j.jcp.2014.01.006.
URL http://www.sciencedirect.com/science/article/pii/S0021999114000278

Appendix 03
Temporal discretization schemes

CSI(Comité de Suivi Individuel) report

Kim Young Jun

Supervisor: prof. David Le Touzé

Ecole Centrale de Nantes

Laboratoire de recherche en Hydrodynamique, Énergétique et Environnement Atmosphérique (LHEEA)

Nantes, 09/2019

1 Introduction

During the past decade, the computational fluid dynamic (CFD) simulations on the naval applications based on finite volume method (FVM) are firmly established. [**?**] The naval applications include wave generation and propagation, fluid-structure interaction (FSI), and various turbulence phenomena. Most of the naval applications are an inherently unsteady problem and solve unsteady RANS (Reynolds Averaged Navier–Stokes). The incompressible two-phase unsteady Navier-Stokes equations solve a coupled system of ordinary differential equations (ODEs) in time. Thus, the simulation of unsteady phenomena often requires efficient time integration methods (temporal discretization or time-stepping method). The higher-order time integration schemes are believed to have the advantages of high order accuracy, low dissipation/dispersion, and capability of handling both complex physics and geometries. However, the efficiency of higher-order time integration schemes has not carefully studied yet for the two-phase naval application.

The study of numerical integration is a long developed topic, but classical methods are still very common in CFD. The OpenFOAM selectable time integration methods are Implicit Euler (Euler or BDF1), Crank Nicolson (CN), and second-order backward differential formula (backward, BDF2) method. The Implicit Euler time integration method is known to be stable, but the dissipations are very large, especially at the interface. The Crank-Nicolson scheme is a semi-implicit method, and it is the most commonly used time integration method in OpenFOAM. However, the Crank-Nicolson scheme has a stability issue. Therefore, off-centering is required, and this decreases the order of convergence. The BDF2 method gives second-order convergence with the Courant number, but it does not guarantee the boundedness of a color function [1].

The most frequently used higher-order temporal discretization scheme for the FVM simulation is third-order TVD (total variable diminishing) Runge-Kutta scheme [2, 3, 4, 5]. The semi-implicit Runge-Kutta schemes are continuously used temporal discretization scheme for the CFD simulation [10, 4, 11, 12]. It is also named Butcher's Diagonally implicit Runge-Kutta (DIRK) method [8]. The DIRK methods are already widely applied for the stiff ODEs. The DIRK method are expected to have more flexibility on the CFL limit. Bijl et al. investigated the accuracy and efficiency of the BDF and Explicit-first SDIRK (ESDIRK) family schemes for the unsteady compressible Navier–Stokes equations [9]. Author compared the error respect to computation time (work) and concluded that fourth- and fifth-order Runge–Kutta schemes are more efficient than the BDF2 scheme. D'Alessandro et al. compared second- and third-order DIRK schemes with Euler, Crank-Nicolson, and explicit Runge-Kutta scheme based on the OpenFOAM platform [12]. Author reported the reduction of numerical diffusion for the one-phase incompressible flow simulation with DIRK method.

The efficiency of higher-order time integration methods are not well investigated yet for two-phase flow model. This paper aims to investigate the higher-order time integration scheme for the two-phase flow model. To increase the order of convergence in time, DIRK type methods are applied inside the classical PIMPLE loop. The accuracy and efficiency of methods are compared for the simple naval applications.

2 Implicit Runge-Kutta method

This chapter gives the explanation on the various implicit Runge-Kutta schemes and its properties. With Implicit Runge-Kutta (IRK) temporal discretization, an arbitrary ordinary differential equation (ODE, Eqn. (2.1)) can be approximated into Eqn. (2.2) and Eqn. (2.3). Where, s is the number of Runge-Kutta iteration; h is the time-step; a_{ij} is the stage weight; b_i is the scheme weight; τ is the abscissa (sub-step) for Runge-Kutta iteration; $y(t^{(n,i)})$ is the sub-step output valid at $t^{(n)} + h\tau_{(i)}$.

$$\dot{y} = f(t, y(t)) \tag{2.1}$$

$$\begin{aligned} y(t^{(n,i)}) = y^{(i)} = y^{(n)} + ha_{ii} f(t^{(n,i)}, y(t^{(n,i)})) + h\sum_{j=1}^{i-1} a_{ij} f(t^{(n,j)}, y(t^{(n,j)})) \\ t^{(n,i)} = t^{(n)} + h\tau_{(i)} \end{aligned} \tag{2.2}$$

$$y^{(n+1)} = y^{(n)} + h\sum_{j=1}^{s} b_j f(t^{(n,j)}, y(t^{(n,j)})) \tag{2.3}$$

The coefficients of equations Eqn. (2.2) and Eqn. (2.3) define a Runge-Kutta method, and a set of the coefficients can be presented in a form of Butcher matrix (Eqn. (2.4)).

$$\begin{array}{c|cccc} \tau_1 & a_{11} & 0 & 0 & 0 \\ \tau_2 & a_{21} & a_{22} & 0 & 0 \\ \tau_3 & a_{31} & a_{32} & a_{33} & 0 \\ \tau_4 & a_{41} & a_{42} & a_{43} & a_{44} \\ \hline & b_1 & b_2 & b_3 & b_4 \end{array} \qquad \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array} \tag{2.4}$$

In this study, implicit Runge-Kutta (IRK) technique with a lower triangular Butcher matrix is considered. By using a lower triangular matrix, the equations may then be solved in a successive manner. This class of methods are named diagonally implicit Runge–Kutta (DIRK) method. DIRK class methods is most widely used IRK method in practical application. Table 2.1 and Table 2.2 show the families of four-stage DIRK method. These tables include sub-classes of DIRK, which are S, E and SA.

The letter S denotes Singly diagonal. Forcing each diagonal element $a_{ii} = \gamma$, greatly facilitates designing A- and L-Stable method by virtue of simplifying the stability function. For some ODE equation, this property could gives benefit on the matrix solver by using repeatedly the stored LU-factorization [13, 14].

The letter E denotes explicit first stage. The explicit first stage with $a_{11} = \tau_1 = 0$ are utilized to make the stage order two (Table 2.2). This use the solution from the previous time step as a first RK sub-step. The explicit first stage method is relatively very inexpensive compared to the computation of the other implicit stages. Therefore, when approximating the computation cost

of DIRK method, the number of the implicit stage (s_I) must be considered. The explicit first stage is also known to give benefit on the convergence rate of the solution to very stiff ODEs [?]. The simplest EDIRK (ESDIRK) possible to imagine is the Crank-Nicolson method. It is two-step ($s = 2$) explicit first stage ($s_I = s - 1$) DIRK method (Eqn. (2.6)).

Another constraint on the coefficients, which is both beneficial and straightforward is the stiffly-accurate (SA) assumption which imposes $a_{sj} = b_j$ and $\tau_s = 1$ [15]. By applying SA assumption to Eqn. (2.2), the final Runge-Kutta stage ($i = s$) of Eqn. (2.2) is now equal to Eqn. (2.3). This assumption benefits both the stability and rate of convergence of numerical solutions of the stiff equation.

Table 2.1. Four-stage DIRK, SDIRK and SDIRK_SA [16]

$$
\begin{array}{c|cccc}
\tau_1 & a_{11} & 0 & 0 & 0 \\
\tau_2 & a_{21} & a_{22} & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & a_{33} & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & a_{44} \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
\tau_1 & \gamma & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
\tau_1 & \gamma & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
1 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & a_{41} & a_{42} & a_{43} & \gamma
\end{array}
$$

Table 2.2. Four-stage EDIRK, ESDIRK and ESDIRK_SA [16]

$$
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & a_{22} & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & a_{33} & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & a_{44} \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
\tau_4 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & b_1 & b_2 & b_3 & b_4
\end{array}
\qquad
\begin{array}{c|cccc}
0 & 0 & 0 & 0 & 0 \\
\tau_2 & a_{21} & \gamma & 0 & 0 \\
\tau_3 & a_{31} & a_{32} & \gamma & 0 \\
1 & a_{41} & a_{42} & a_{43} & \gamma \\
\hline
 & a_{41} & a_{42} & a_{43} & \gamma
\end{array}
$$

The stability of a numerical method is measured by using the test equation $\dot{y} = ky$ ($\mathrm{Re}(k) < 0$) with initial condition $y(0) = 1$. If the numerical solution approaches zero as $t \to \infty$, then the method is A-Stable. The stability of Runge-Kutta method can be determined using stability function Eqn. (2.5), where $\mathbf{e}$ denotes the vector with ones and $z = kh$.

$$
R(z) = \frac{\det(\mathbf{I} - z\mathbf{A} + z\mathbf{e}\mathbf{b}^T)}{\det(\mathbf{I} - z\mathbf{A})} \tag{2.5}
$$

If $|R(z)| \leq 1$ for $\mathrm{Re}(k) < 0$, the stability function is A-acceptable and the method is A-stable. In addition to A-stability, if $R(z)_{z \to -\infty} = 0$, this stability function is L-stable and the method is L-stable. The L-stability provides damping of high-frequency modes which may be non-physical, and perfect damping of modes at infinity. A simple way to achieving L-stability is to use stiffly-accurate assumption. The stiffly-accurate assumption forces $y^{(n+1)}$ to be computed implicitly, while also increasing the convergence rate. Finally, for the easier naming of each method, the identification of Runge-Kutta methods are given (Table 2.3).

Table 2.3. Identification of implicit Runge-Kutta method (ESDIRK[p](s)_SA_i)

E	Explicit first stage	S	Singly
DI	Diagonally-implicit	RK	Runge-Kutta
p	Global order	s	Number of stage
SA	Stiffly accurate	i	identifier (when required)

2.1 Stiffly accurate SDIRK schemes

In this study, only stiffly-accurate schemes are considered for simpler implementation. Since the boundedness of volume of fraction (VOF) is an important criterion, the reduction of an equation is thought to be beneficial (Not valid yet). Second-, third- and forth- order stiffly-accurate (E)(S)DIRK schemes are introduced for each subsection. The derivation and the backgrounds on each DIRK method is not going to be treated in this paper. However, some of important characterizations of each method are discussed shortly. For more details on the DIRK-type methods, the interested reader is referred to review papers [**?**, 16]. Two kind of names are used. One is the identification name which is named using Table 2.3. Another is simplified name which is named 'RKspi', where 's', 'p' and 'i' is global order, number of stage and identifier, respectively.

2.1.1 Second-order scheme

ESDIRK[2](2)_SA ($s_I = 1$)**, RK_CN**

Excluding the first-order L-stable Implicit Euler method, Eqn. (2.6) is one of the simplest DIRK method. It is second-order accurate, two stage, explicit first, stiffly-accurate and A-stable method. It is also well known as Crank-Nicolson (or Lobatto IIIA or trapezoidal rule) method and it is already implemented in OpenFOAM in a different way.

$$\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0.5 & 0.5 \\ \hline & 0.5 & 0.5 \end{array} \tag{2.6}$$

SDIRK[2](2)_SA ($s_I = 2$)**, RK221 [17]**

Two stage, stiffly-accurate and L-stable SDIRK method is presented. Two possible solutions are exist : $\gamma = (2 \pm \sqrt{2})/2$. Both are L-stable, but $\gamma = (2 - \sqrt{2})/2$ is more accurate and stable.

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1 & 1-\gamma & \gamma \\ \hline & 1-\gamma & \gamma \end{array} \tag{2.7}$$

ESDIRK[2](3)_SA ($s_I = 2$)**, RK321**

Second order, three stage, stiffly-accurate, ESDIRK methods is presented. L-stable is achieved when $b_2 = (1-2\gamma)/(4\gamma)$ and $\gamma = (2-\sqrt{2})/2$.

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2\gamma & \gamma & \gamma & 0 \\ 1 & (1-b_2-\gamma) & b_2 & \gamma \\ \hline & (1-b_2-\gamma) & b_2 & \gamma \end{array} \tag{2.8}$$

2.1.2 Third-order scheme

SDIRK[3](3)_SA ($s_I = 3$)**, RK331 [17]**

Three stage, third order, stiffly-accurate, L-stable SDIRK methods. The L-stability achieved only with $\gamma = 0.435866521508458999416019 45$. $A^{(4)} = 0.02970$,

$$\begin{array}{c|ccc} \gamma & \gamma & 0 & 0 \\ \frac{(1+\gamma)}{2} & \frac{(1+\gamma)}{2} - \gamma & \gamma & 0 \\ 1 & -\frac{(6\gamma^2-16\gamma+1)}{4} & \frac{(6\gamma^2-20\gamma+5)}{4} & \gamma \\ \hline & -\frac{(6\gamma^2-16\gamma+1)}{4} & \frac{(6\gamma^2-20\gamma+5)}{4} & \gamma \end{array} \tag{2.9}$$

EDIRK[3](3)_SA ($s_I = 2$)**, RK333**

Third order, three stage, stiffly-accurate, EDIRK methods. (Not singly-diagonal)

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ \frac{7}{6} & \frac{7}{12} & \frac{7}{12} & 0 \\ 1 & \frac{5}{14} & -\frac{6}{7} & \frac{3}{2} \\ \hline & \frac{5}{14} & -\frac{6}{7} & \frac{3}{2} \end{array} \tag{2.10}$$

ESDIRK[3](3)_SA ($s_I = 2$)**, RK334**

Third order, three stage, stiffly-accurate, ESDIRK methods.

$b_2 = (1-2\gamma)/(4\gamma)$ and $\gamma = (3+\sqrt{3})/6$.

$$\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 2\gamma & \gamma & \gamma & 0 \\ 1 & (1-b_2-\gamma) & b_2 & \gamma \\ \hline & (1-b_2-\gamma) & b_2 & \gamma \end{array} \tag{2.11}$$

ESDIRK[3](4)_SA ($s_I = 2$)**, RK431**

Third order, four stage, stiffly-accurate, ESDIRK methods. Stage order two. L-stability.

$$a_{32} = \frac{c_3(c_3 - 2\gamma)}{4\gamma} \quad b_2 = \frac{-2 + 3c_3 + 6\gamma(1 - c_3)}{12\gamma(c_3 - 2\gamma)} \quad b_3 = \frac{1 - 6\gamma + 6\gamma^2}{3c_3(c_3 - 2\gamma)} \tag{2.12}$$

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 2\gamma & \gamma & \gamma & 0 & 0 \\ c_3 & (c_3 - a_{32} - \gamma) & a_{32} & \gamma & 0 \\ 1 & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & \gamma \\ \hline & (1 - b_2 - b_3 - \gamma) & b_2 & b_3 & \gamma \end{array} \tag{2.13}$$

Method can be optimized using c_3. minimum truncation error (A^4) can be achieve where

$$c_3 = \frac{3 - 20\gamma + 24\gamma^2}{4 - 24\gamma + 24\gamma^2} \tag{2.14}$$

3 Implementation to OpenFOAM

This chapter present the implementation of the DIRK time integration method to OpenFOAM incompressible two-phase flow solver (interDyMFoam). The volume of fluid (VOF) method is used to define the free-surface. The VOF method defines the phase indicator function α as a volume fraction of water in a cell. The governing equations give the VOF transport equation and two-phase incompressible Newtonian flow model equations.

$$\frac{\partial(\alpha)}{\partial t} + \nabla\bullet(\mathbf{u}\alpha) + \nabla\bullet(\mathbf{u}_r\alpha(1-\alpha)) = 0 \tag{3.1}$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\bullet(\rho\mathbf{u}\mathbf{u}) - \nabla\bullet(\mu(\nabla\mathbf{u})) - \nabla\mathbf{u}\bullet\nabla\mu = -\nabla p_d - (\mathbf{g}\bullet\mathbf{x})\nabla\rho \tag{3.2}$$

$$\nabla\bullet\mathbf{u} = 0 \tag{3.3}$$

Where, $\mathbf{u}$ is a continuous velocity field; ρ is a continuous density field which has spatial variation due to two fluids ($\rho_{air} = 1.0$ Kg/m^2, $\rho_{water} = 1000$ Kg/m^2); μ is a averaged dynamic viscosity ($\mu_{air} = 1.0\times10^{-5}$ Ns/m^2, $\mu_{air} = 1.0\times10^{-3}$ Ns/m^2); p_d is a dynamic pressure; $\mathbf{x}$ is the point field and this could change with mesh movement.; σ_κ is a surface tension coefficient and considered negligible ($\sigma_\kappa = 0$). The Eqn. (3.4) and Eqn. (3.5) show the finite volume (FV) representation of VOF convection equation and momentum equation.

$$\frac{\partial(V_P\alpha)}{\partial t} + \sum_f(\phi\alpha)_f + \sum_f(\phi_{r,f}\alpha_f(1-\alpha_f)) = 0 \tag{3.4}$$

$$\frac{\partial(V_P\rho\mathbf{u})}{\partial t} + \sum_f((\rho\phi)_f\mathbf{u}_f) - \sum_f(\mu_f(\nabla\mathbf{u}_f)) - \nabla\mathbf{u}\bullet\nabla\mu V_P = -\nabla p_d V_P - (\mathbf{g}\bullet\mathbf{x})\nabla\rho V_P\,. \tag{3.5}$$

$$(\rho\phi)_f = (\phi\alpha)_f(\rho_{water} - \rho_{air}) + \phi_f\rho_{air} \tag{3.6}$$

The subscript P and f refers to the owner cell's center properties and face center properties, respectively; $\mathbf{u}_f$ and μ_f is its cell-to-face interpolated value; V_P is the cell volume; Assuming that the geometry of the cell P dose not change in time, the Eqn. (3.4) and Eqn. (3.5) are coupled ODEs respect to time, and Eqn. (3.6) non-linearly couples two governing equations.

3.1 DIRK implementation to foamStar

The implementation algorithm of DIRK method to interDyMFoam is describe in this section. The superscript i is a current RK iteration index; s is a total RK step; n refers to the previous time; $n+1$ refers to the current time. The time-step is defined as $h = t^{(n+1)} - t^{(n)}$. In this study,

only stiffly-accurate DIRKs are considered. Therefore, the final stage output is equal to current time step result ($y(t^{(n,s)}) = y^{(n+1)}$).

$$y(t^{(n,i)}) = y^{(n)} + ha_{ii}f(t^{(n,i)}, y(t^{(n,i)})) + h\sum_{j=1}^{i-1} a_{ij}f(t^{(n,j)}, y(t^{(n,j)})) \qquad t^{(n,i)} = t^{(n)} + h\tau_{(i)} \tag{3.7}$$

Fig. 3.1 illustrate the full algorithm of targeted solver. The Runge-Kutta scheme is applied upon the PIMPLE iteration. Therefore, this algorithm solve PIMPLE loop for each RK sub-time step ($t^{(n,i)}$), and it use previous RK sub-steps ($1 \sim i-1$) results to enhance the order of scheme. The explicit first stage algorithm saves previous time step data as first RK sub-step ($i = 1$) output, and this is treated inside the 'Runge-Kutta controls'. The mesh and relaxation zone weight are update only for the final RK sub-step ($i = s$) for the robustness and efficiency of the solver. The PIMPLE iteration includes the modified VOF(α) convection equation and the PISO loop. Following section will describe the application of RK algorithm to VOF convection equation (Eqn. (3.4)) and momentum equation (Eqn. (3.5)).

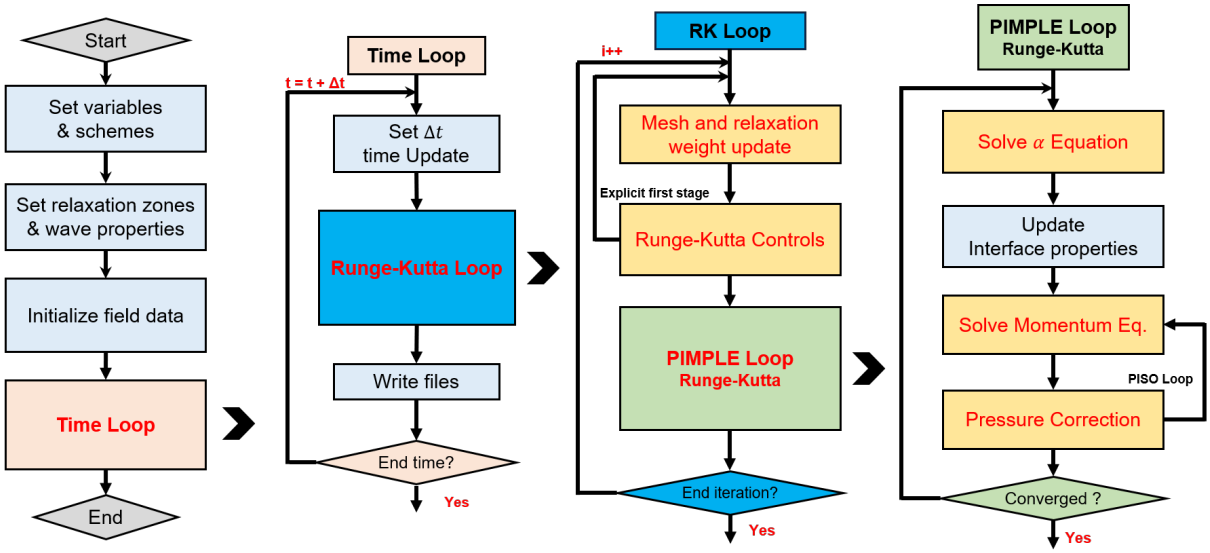


Figure 3.1. foamStarRK algorithm

3.2 The VOF convection algorithm

This section gives the detail implementation and flow chart of VOF (α) convection algorithm. Some arguments are defined to facilitate the formulation. The subscript $_{BD}$ refers to bounded spatial discretization scheme (upwind) applied to the VOF flux ($\phi\alpha$); $_{UB}$ refers to unbounded spatial discretization scheme (second-order). The subtraction from unbounded VOF flux to bounded VOF flux is named correction VOF flux and use the subscript $_{Corr}$.

For a static mesh, the application of Eqn. (3.7) to Eqn. (3.4) yields Eqn. (3.8).

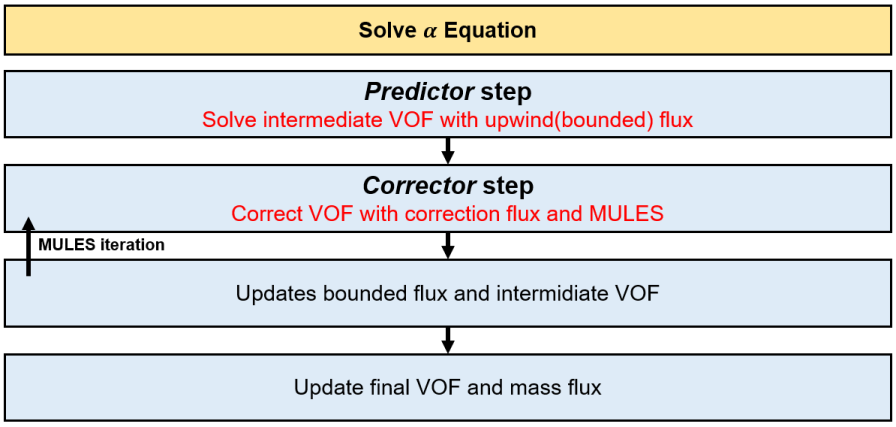


Figure 3.2. VOF convection algorithm

$$\begin{aligned}\frac{V_P^{(i)}\alpha_P^{(i)} - V_P^{(n)}\alpha_P^{(n)}}{h} + a_{ii}\sum_f (\phi_f\alpha_f)^{(i)} + a_{ii}\sum_f (\phi_{r,f}\alpha_f(1-\alpha_f))^{(i)} \\ + \sum_{j=1}^{i-1} a_{ij}\sum_f (\phi_f\alpha_f)^{(j)} + \sum_{j=1}^{i-1} a_{ij}\sum_f (\phi_{r,f}\alpha_f(1-\alpha_f))^{(j)} = 0\end{aligned} \tag{3.8}$$

For the boundedness of the VOF convection, the *Predictor* & *Corrector* steps are applied [**?**]. The Eqn. (3.8) is separated into *Predictor* equation Eqn. (3.9) and *Corrector* equation Eqn. (3.10).

$$\frac{V_P^{(i)}\alpha_P^{*(i)} - V_P^{(n)}\alpha_P^{(n)}}{h} + a_{ii}\sum_f (\phi_f\alpha_f^*)_{BD}^{(i)} + \sum_{j=1}^{i-1} a_{ij}\sum_f (\phi_f\alpha_f^*)_{BD}^{(j)} = 0 \tag{3.9}$$

$$\begin{aligned}\frac{V_P^{(i)}\alpha_P^{(i)} - V_P^{(i)}\alpha_P^{*(i)}}{h} + a_{ii}\sum_f [(\phi_f\alpha_f)^{(i)} - (\phi_f\alpha_f^*)_{BD}^{(i)}] + a_{ii}\sum_f (\phi_{r,f}\alpha_f(1-\alpha_f))^{(i)} \\ + \sum_{j=1}^{i-1} a_{ij}\sum_f [(\phi_f\alpha_f)^{(j)} - (\phi_f\alpha_f^*)_{BD}^{(j)}] + \sum_{j=1}^{i-1} a_{ij}\sum_f (\phi_{r,f}\alpha_f(1-\alpha_f))^{(j)} = 0\end{aligned} \tag{3.10}$$

The α_P^* is an intermediate VOF, and it is the solution of Eqn. (3.9). The bounded (upwind) flux is used to evaluate the intermediate VOF, and the artificial compression is not included. It is important to notice that, solver has to store the previous RK sub-step flux and VOF ($\phi^{(j)}$ and $\alpha^{(j)}$) to solve Eqn. (3.9). The bounded VOF flux at sub-step i is named $F_{BD}^{(i)}$.

$$F_{BD}^{(i)} = a_{ii}\sum_f [(\phi_f\alpha_f^*)_{BD}^{(i)}] \qquad F_{BD}^{(j)} = a_{ij}\sum_f [(\phi_f\alpha_f)_{BD}^{(j)}] \tag{3.11}$$

For the accurate but bounded correction, the MULES algorithm is applied. MULES limiter limits the VOF flux and assure the boundedness and continuity of VOF flux. With MULES limiter λ, the corrector equation transform into Eqn. (3.12) and Eqn. (3.13).

$$\frac{V_P^{(i)}\alpha_P^{**(i)} - V_P^{(i)}\alpha_P^{*(i)}}{h} + \lambda\left[F(\phi^{(i)}, \alpha^{*(i)})_{Corr} + \sum_{j=1}^{i-1} F(\phi^{(j)}, \alpha^{(j)})_{Corr}\right] = 0 \tag{3.12}$$

$$\begin{aligned} F(\phi^{(i)}, \alpha^{*(i)})_{Corr} &= a_{ii}\sum_f \left[(\phi_f\alpha_f^*)_{UB}^{(i)} + (\phi_{f,rel}\alpha_f^*(1-\alpha_f^*))_{UB}^{(i)}\right] - F_{BD}^{(i)} \\ F(\phi^{(j)}, \alpha^{(j)})_{Corr} &= a_{ij}\sum_f \left[(\phi_f\alpha_f)_{UB}^{(j)} + (\phi_{f,rel}\alpha_f(1-\alpha_f))_{UB}^{(j)}\right] - F_{BD}^{(j)} \end{aligned} \tag{3.13}$$

The α_P^{**} is a corrected VOF field, and λ is a explicit limiter field evaluated from MULES algorithm. After the correction of the VOF, to do the next correction based on current results, Eqn. (3.14) and Eqn. (3.15) updates the bounded flux and indeterminate VOF, respectively. Next, the intermediate VOF will be corrected again with corrected VOF and VOF flux. This loop is named MULES iteration.

$$\begin{aligned} F_{BD}^{(i)} &= F_{BD}^{(i)} + \lambda\sum_f F(\phi^{(i)}, \alpha^{*(i)})_{Corr} \\ F_{BD}^{(j)} &= F_{BD}^{(j)} + \lambda\sum_f F(\phi^{(i)}, \alpha^{(j)})_{Corr} \end{aligned} \tag{3.14}$$

$$\alpha_P^{*(i)} = \alpha_P^{**(i)} \tag{3.15}$$

After the MULES iteration, final updated the VOF flux and VOF field at RK sub-step i can be derived (Eqn. (3.16)). Finally, the Eqn. (3.17) shows the density field and the mass flux for RK sub-step i.

$$(\phi\alpha)_f^{(i)} = F_{BD}^{(i)}/a_{ii} \qquad \alpha_P^{(i)} = \alpha_P^{**(i)} \tag{3.16}$$

$$\rho_P^{(i)} = \alpha_P^{(i)}\rho_{water} + (1-\alpha_P^{(i)})\rho_{air} \qquad (\rho\phi)_f^{(i)} = (\phi\alpha)_f^{(i)}(\rho_{water} - \rho_{air}) + \phi^{(i)}\rho_{air} \tag{3.17}$$

3.3 The PISO algorithm

This section presents the algorithm of PISO iteration inside the Runge-Kutta loop. The design of RK PISO loop is very similar to the origianl PISO loop. Only difference would be the additional source terms and coefficients in the equation. Assuming that mesh does not move in time ($\mathbf{x}^{(i)} = \mathbf{x}^{(n)}, V_P^{(i)} = V_P^{(n)}$), the substitution of Eqn. (3.7) to momentum equation gives Eqn. (3.18). The mass flux ($(\rho\phi)_f^{(i)}$) is a constant inside the PISO loop, and it is calculated from VOF convection equation.

$$\frac{(V\rho\mathbf{u})_P^{(i)} - (V\rho\mathbf{u})_P^{(n)}}{h} + a_{ii}\mathbf{R}^{(i)} + \sum_{j=1}^{i-1} a_{ij}\mathbf{R}^{(j)} \tag{3.18}$$

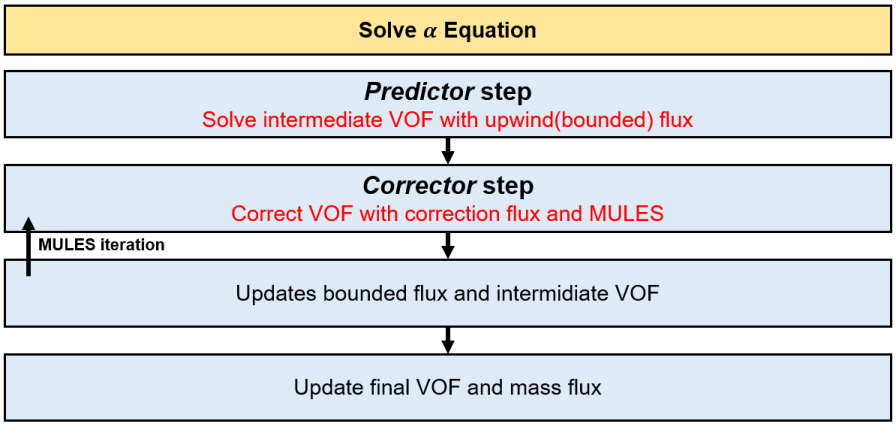


Figure 3.3. VOF convection algorithm

$$\mathbf{R}^{(i)} = \sum_f \left[(\rho\phi)_f^{(i)}\mathbf{u}_f^{(i)}\right] - \sum_f \left[\mu_f^{(i)}(\nabla\mathbf{u}_f^{(i)})\right] - \nabla\mathbf{u}_P^{(i)}\bullet\nabla\mu V_P^{(i)} + \nabla p_d^{(i)} V_P^{(i)} + (\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)} V_P^{(i)} \tag{3.19}$$

The transposition of dynamic pressure term and hydro-static term results Eqn. (3.20). The variable $\mathbf{a}_P$ and $\mathbf{H}$ are commonly used OpenFOAM notation for the explanation of PISO algorithm [18]. $\mathbf{a}_P^{(i)}$ indicates the diagonal components of UEqn (fvVectorMatrix) at RK sub-step i, and $\mathbf{H}^{(i)}$ includes the off-diagonal terms and all the source terms.

$$\begin{aligned}\frac{(V\rho\mathbf{u})_P^{(i)} - (V\rho\mathbf{u})_P^{(n)}}{h} + a_{ii}\Big[\sum_f \left[(\rho\phi)_f^{(i)}\mathbf{u}_f^{(i)}\right] - \sum_f \left[\mu_f^{(i)}(\nabla\mathbf{u}_f^{(i)})\right] - \nabla\mathbf{u}_P^{(i)}\bullet\nabla\mu V_P^{(i)}\Big] \\ + \sum_{j=1}^{i-1} a_{ij}\mathbf{R}^{(j)} = -a_{ii}\left[\nabla p_d^{(i)} V_P^{(i)} + (\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)} V_P^{(i)}\right] \\ = \mathrm{UEqn}^{(i)} = \mathbf{a}_P^{(i)}\mathbf{u}_P^{(i)} - \mathbf{H}^{(i)}\end{aligned} \tag{3.20}$$

The resulting velocity of Eqn. (3.20) must satisfy the continuity constraint. The application of continuity equation (Eqn. (3.3)) provides Eqn. (3.22).

$$\nabla\mathbf{u}_P^{(i)} = 0 = \nabla\Big[\frac{\mathbf{H}^{(i)}}{\mathbf{a}_P^{(i)}} - \frac{a_{ii}}{\mathbf{a}_P^{(i)}}\left[\nabla p_d^{(i)} V_P^{(i)} + (\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)} V_P^{(i)}\right]\Big] \tag{3.21}$$

$$\nabla\Big[\frac{a_{ii}}{\mathbf{a}_P^{(i)}}\nabla p_d^{(i)} V_P^{(i)}\Big] = \sum_f \Big[\Big(\frac{a_{ii}}{\mathbf{a}_P^{(i)}}\nabla p_d^{(i)}\Big)_f\Big] = \sum_f \Big[\frac{\mathbf{H}^{(i)}}{\mathbf{a}_P^{(i)}} + \frac{a_{ii}}{\mathbf{a}_P^{(i)}}(\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)} V_P^{(i)}\Big]_f \tag{3.22}$$

After the dynamic pressure is obtained, the flux and velocity are corrected using dynamic pressure.

$$\begin{aligned}\phi^{(i)} &= \left[\frac{\mathbf{H}^{(i)}}{\mathbf{a}_P^{(i)}} + \frac{a_{ii}}{\mathbf{a}_P^{(i)}}(\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)}V_P^{(i)}\right]_f - \left[\frac{a_{ii}}{\mathbf{a}_P^{(i)}}\nabla p_d^{(i)}\right]_f \\ \mathbf{u}^{(i)} &= \left[\frac{\mathbf{H}^{(i)}}{\mathbf{a}_P^{(i)}} + \frac{a_{ii}}{\mathbf{a}_P^{(i)}}(\mathbf{g}\bullet\mathbf{x}^{(i)})\nabla\rho_P^{(i)}V_P^{(i)}\right] - \left[\frac{a_{ii}}{\mathbf{a}_P^{(i)}}\nabla p_d^{(i)}\right]\end{aligned} \tag{3.23}$$

3.4 Consideration of moving mesh

If the meshes move during the simulation, the change of quantities due to the mesh movement must be considered. This section provide the a explanation of moving mesh treatment and application to implicit Runge-Kutta method. The Space Conservation Law (SCL) can be used to consider the movement of mesh [19]. The integral form for the SCL can be derived directly from the mass conservation for a moving boundary.

$$\frac{d}{dt}\int_V \rho dV + \oint_f \rho(\mathbf{u}-\mathbf{u}_m)\bullet\mathbf{n}dS = 0 \tag{3.24}$$

The fluid velocity is $\mathbf{u}$ and the velocity of the boundary of the control volume is $\mathbf{u}_m$ For simpler expression, time invariant uniform field $\rho = 1$ and $\mathbf{u} = 0$ is applied. The finite volume (FV) expression is

$$\frac{d}{dt}V = \oint_f \mathbf{u}_m\bullet\mathbf{n}dS. \tag{3.25}$$

The time integration from t to $t+h$ is

$$\int_t^{t+h}\frac{d}{dt}V = \int_t^{t+h}\sum_f \mathbf{u}_{m,f}\bullet\mathbf{S}. \tag{3.26}$$

$$\frac{1}{h}(V_{p^{t+h}}^{t+h} - V_{p^t}^t) = \Big(\sum_f \mathbf{u}_{m,f}\bullet\mathbf{S}\Big)^{t+h} \tag{3.27}$$

If the exact location of grid points are known, the integral r.h.s can be computed. OpenFOAM measure the signed swept-volume (δV) on each face using only the old points and the new (moved) points (Fig.3.4).

$$\mathbf{u}_{m,f}\bullet\mathbf{S} = \frac{\delta V}{h} = \phi_m^{t+h} \tag{3.28}$$

Another derivation can be derived from Implicit Euler scheme [20]. In the view of ODE, all variables must be independent in time. If $\rho = 1$ and $\mathbf{u} = 0$, in the new mesh, the volume should not change in time.

$$\frac{1}{h}(V_{p^{t+h}}^{t+h} - V_{p^{t+h}}^t) = 0 \tag{3.29}$$

Since $V_{p^{t+h}}^t$ can not be measured, the interpolated value is used.

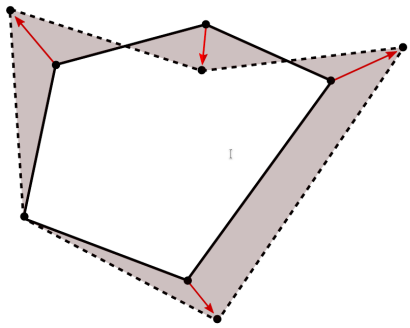


Figure 3.4. The total changes of the control volume and (signed) swept-volume of each face

$$
\begin{aligned}
\frac{V^t_{P^{t+h}}}{h} &= \frac{V^t_{P^t}}{h} + \frac{\nabla V^t_{P^{(t,t+h)}} \bullet (\mathbf{C}^{t+h} - \mathbf{C}^t)}{h} \\
&= \frac{V^t_{P^t}}{h} + \sum_{f^{(t,t+h)}} \frac{\mathbf{S}^t_f \bullet (\mathbf{f}^{t+h}_c - \mathbf{f}^t_c)}{h} \\
&= \frac{V^t_{P^t}}{h} + \sum_{f^{(t,t+h)}} \phi^{t+h}_m
\end{aligned}
\tag{3.30}
$$

Thus, the result is equal to the SCL formulation.

$$
\frac{1}{h}(V^{t+h}_{p^{t+h}} - V^t_{p^t}) = \sum_{f^{t+h}} \phi^{t+h}_m \tag{3.31}
$$

3.4.1 Application of ALE formulation

The application of SCL formulation to VOF convection equation write: The additional term, mesh flux (ϕ_{mesh}), is also adapted to *Predictor* & *Corrector* algorithm.

$$
\begin{aligned}
\frac{V^{(i)}_{P^{(n+1)}} \alpha^{(i)}_{P^{(n+1)}} - V^{(n)}_{P^{(n+1)}} \alpha^{(n)}_{P^{(n+1)}}}{h} &+ \sum_f (\phi_{mesh} \alpha_f)^{(i)} + a_{ii} \sum_f (\phi_f \alpha_f)^{(i)} \\
&+ \sum_{j=1}^{i-1} a_{ij} \sum_f (\phi_f \alpha_f)^{(j)} = 0.
\end{aligned}
\tag{3.32}
$$

The application of SCL formulation to momentum equation write:

$$
\frac{(V\rho\mathbf{u})^{(i)}_P - (V\rho\mathbf{u})^{(n)}_P}{h} + \sum_f \left[(\rho\phi_{mesh})^{(i)} \mathbf{u}^{(i)}_f \right] + a_{ii}\mathbf{R}^{(i)} + \sum_{j=1}^{i-1} a_{ij}\mathbf{R}^{(j)} \tag{3.33}
$$

$$
\mathbf{R}^{(i)} = \sum_f \left[(\rho\phi)^{(i)}_f \mathbf{u}^{(i)}_f \right] - \sum_f \left[\mu^{(i)}_f (\nabla \mathbf{u}^{(i)}_f) \right] - \nabla \mathbf{u}^{(i)}_P \bullet \nabla \mu V^{(i)}_P + \nabla p^{(i)}_d V^{(i)}_P + (\mathbf{g} \bullet \mathbf{x}^{(i)}) \nabla \rho^{(i)}_P V^{(i)}_P \tag{3.34}
$$

4 Applications

The classical foamStar with Crank-Nicolson method and various DIRK-type methods are compared using several applications. Table 4.1 summarized the tested time integration schemes. Only Stiffly accurate (SA) DIRK methods are utilized, and maximum three implicit stages are considered. For the validation of each solver, several two-phase flow test models are applied. These are periodic wave propagation, dam-break, CALM buoy diffraction, and towed KCS (KRISO Container Ship) in wave. Each of the test case is described in detail at the following subsections.

Table 4.1. Name and properties of solvers

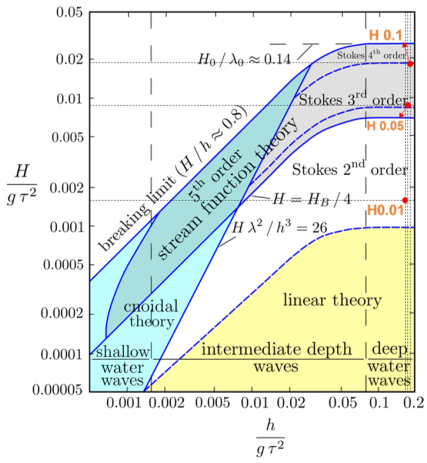
Code name	Identification name	Order	Implicit stage	Notes
OFCN095	OpenFOAM C-N	2	1	off-centered 0.95
RKCN	ESDIRK[2](2)_SA	2	1	-
RKCN095	ESDIRK[2](2)_SA	2	1	off-centered 0.95
RK221	SDIRK[2](2)_S	2	2	-
RK331	SDIRK[3](3)_SA	3	3	-
RK333	EDIRK[3](3)_SA	3	2	-
RK334	ESDIRK[3](3)_SA	3	2	-
RK431	ESDIRK[3](4)_SA	4	3	-

4.1 Periodic wave propagation

4.1.1 Simulation set-up

The periodic wave propagation simulation campaign is constructed with three wave conditions, four mesh resolutions and four temporal resolutions. Fig. 4.1 describe the three different regular wave conditions. The wave length is fixed to unity, and this is equal to the horizontal length of domain . Three wave steepness (H/λ) are used, and wave periods are defined based on the stream function wave theory. Fig. 4.1 shows the information on the domain, mesh and temporal resolution. The reference case of this campaign is 'Grid3 dt400', and this is listed twice for the easier visualization. This grid and temporal resolution is recognized conditions for the accurate wave propagation simulation [**?**]. The expected Courant number for the reference case 'Grid3 dt400' is 0.2. The domain size is uniform regardless to the wave condition. However, since the vertical mesh size is defined based on wave height (H), mesh size and mesh aspect ratio differ respect to the wave condition.

To check the order of convergence of solver, four different resolution of girds and five time-steps are used. In order to minimized the non-linear iteration uncertainly, 8 PIMPLE (outer iteration) and 2 PISO (inner iteration) loops are imposed. With this iterations, the initial residual of all velocity components and dynamic pressure reduced to 10^{-6}. The spatial discretization schemes are also important options for the accurate and stable simulation. Since the meshes used for the



Item	Unit	H0.1	H0.05	H0.01
Depth (D)	[m]	1.0	1.0	1.0
Wave length	[m]	1.0	1.0	1.0
Wave period (T)	[sec]	0.76179	0.79049	0.79991
Wave height (H)	[m]	0.1	0.05	0.01
Wave steepness	H/L	10%	5%	1%
1st order Amp	[m]	0.047413	0.024751	0.004998
2nd order Amp	[m]	0.008294	0.001995	0.000079
3rd order Amp	[m]	0.002263	0.000243	0.000002
4th order Amp	[m]	0.000744	0.000035	-

Figure 4.1. Three wave condition and its wave steepness

Case	$\lambda/\Delta x$	$H/\Delta z$	$T/\Delta t$
Grid 1	25	5	100
Grid 2	50	10	200
Grid 3	100	20	400
Grid 4	200	40	800
Grid 3 dt100	100	20	100
Grid 3 dt200	100	20	200
Grid 3 dt400	100	20	400
Grid 3 dt800	100	20	800
Grid 3 dt1600	100	20	1600

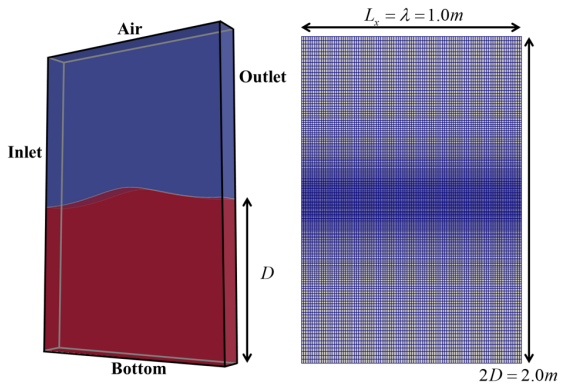


Figure 4.2. Four mesh resolutions and four temporal resolutions

wave propagation simulation has no skewness and fully orthogonal, most accurate (second-order) spatial schemes are utilized. The summary of the applied spatial discretization schemes is listed at Table 4.2.

In additional to typical wave propagation test case, mesh motion is imposed to check the capability of time integration scheme in moving mesh. An imaginary body do the heave and the pitch motion at the free-surface and the surrounding grids morph using smooth weight function. The heave and the pitch period is equal to the regular wave period, and the heave amplitude is $0.01m$ and the pitch amplitude is 1 degree. Since the periodic boundary condition is applied, this test condition is tough, and the wave quality will degenerate in time.

4.1.2 Wave propagation results

The efficiency of each temporal discretization scheme is measured using accuracy and CPU time. Fig 4.3, Fig 4.4 and Fig 4.5 show the regular wave propagation results. Y-axis is the normalized averaged error in time and X-axis is the total computation time. Both axes are in log scale. For the multi-core computation, the computation time is linearly multiplied by number of core.

Table 4.2. Spatial discretization schemes

Terms	Equation	Solvers
Gradient	∇a	Gauss linear
S.N. gradient	$\mathbf{n}\bullet(\nabla a)_f$	Corrected 1.0
VOF convection	$\nabla\bullet(\mathbf{u}\alpha)$	Gauss vanLeer
VOF relative	$\nabla\bullet(\mathbf{u}_r\alpha(1-\alpha))$	Gauss linear
Mom. convection	$\nabla\bullet(\rho\mathbf{uu})$	Gauss linearUpwindV
Mom. Laplacian	$\nabla\bullet(\mu\nabla(\mathbf{u}))$	Gauss linear corrected
Pressure Laplacian	$\nabla\bullet(\frac{1}{a_p}\nabla(p))$	Gauss linear corrected
Under-relaxation	$U, UFianl$	1.0, 1.0

The figure on the left only plot the Grid1 to Grid4 result, and right figure plot includes all test cases. Two figures are illustrated to clearly separate the error from mesh resolution and temporal resolution. The black doted line is the reference case which used OpenFOAM CN scheme with computation condition 'Grid3 dt400' in Table 4.2. Solver RK333 shows unstable and unpredictable results for all wave conditions, thus this is excluded for the discussion.

To estimate the efficiency, the acceptable error should be determined. Since the wave propagation is basis of all naval application, this study targeted to have around 1% of error. In this point of view, the mesh 'Grid1' is not appropriate for the proper wave propagation. Except the results of 'Grid1' RKCN and RK221 scheme shows the best efficiency for all wave conditions. Also, author must notice that there have large difference between OpenFOAM Crank-Nicolson and DIRK based Crank-Nicolson. This difference is not only due to the off-centering, but also due to the treatment of the mass flux Eqn. (3.16).

Fig 4.6 indicates the capability of solver with moving mesh. Compare to the static mesh, the accuracy has decreased and the variation of the solutions are large. Considering the cumulative error in the periodic test case, the resulting solutions from DIRK schemes are acceptable.

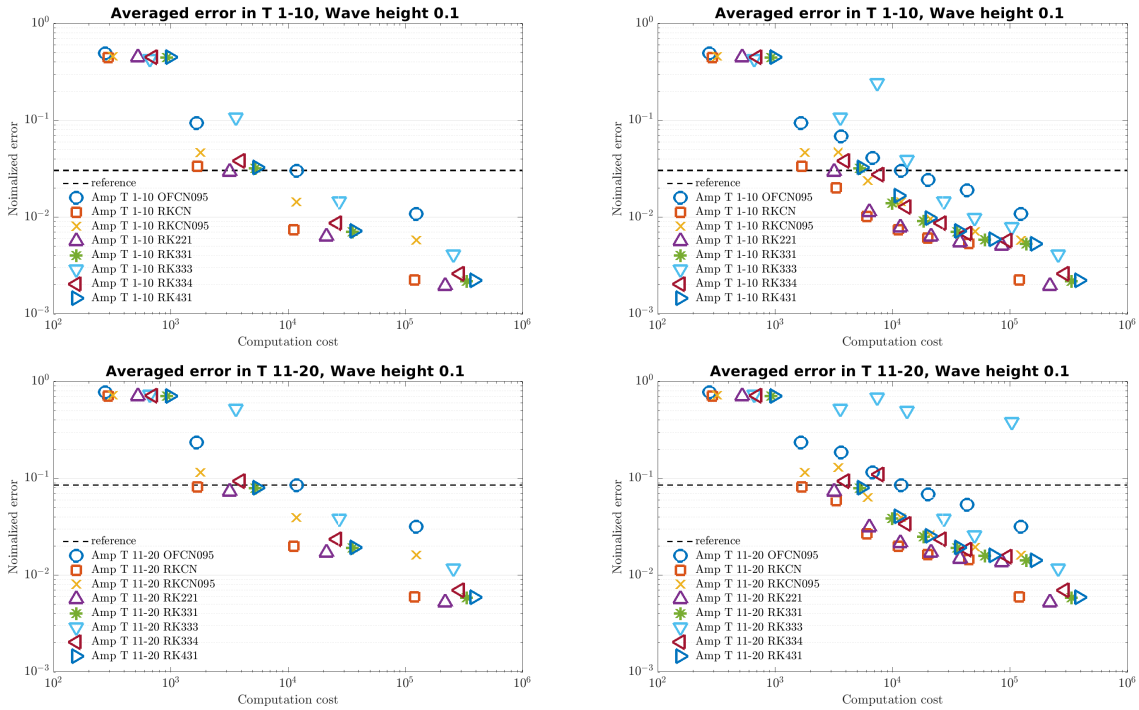


Figure 4.3. The comparison of the efficiency of the solvers, wave height H0.1

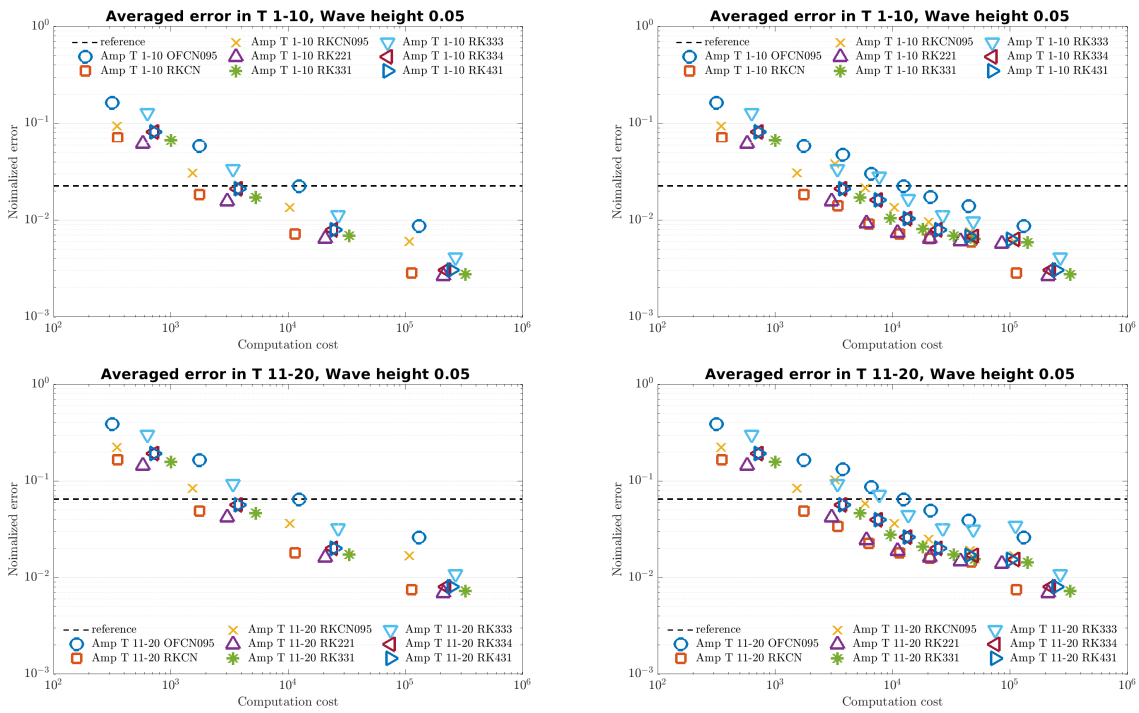


Figure 4.4. The comparison of the efficiency of the solvers, wave height H0.05

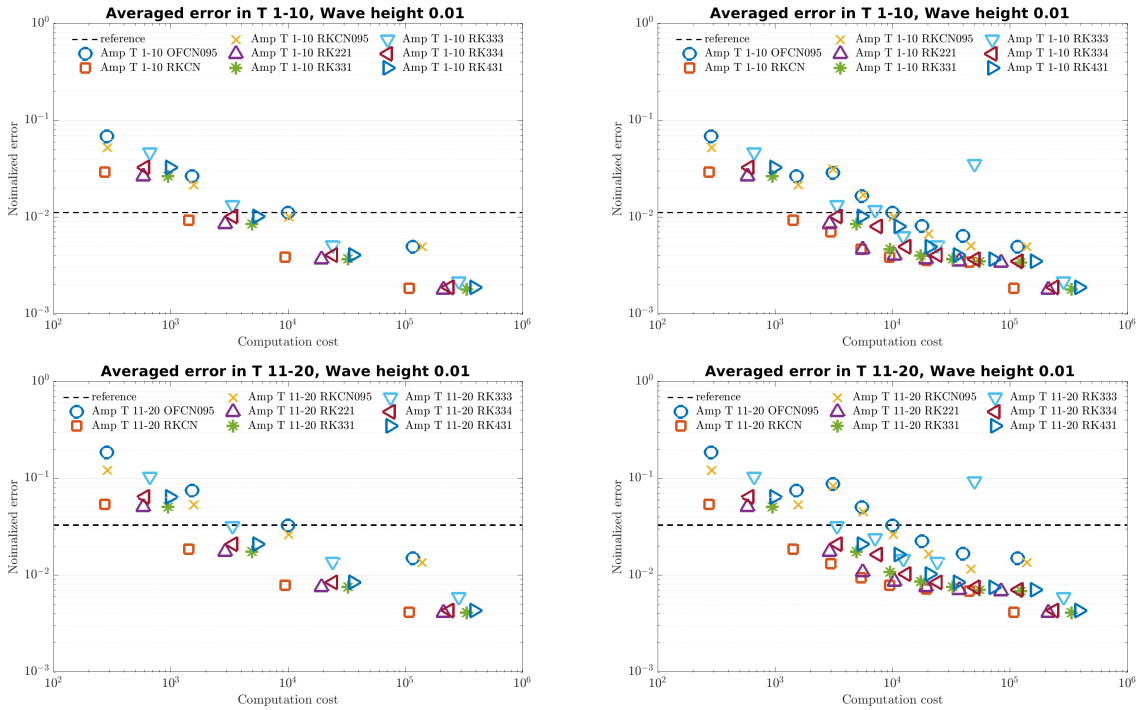


Figure 4.5. The comparison of the efficiency of the solvers, wave height H0.01

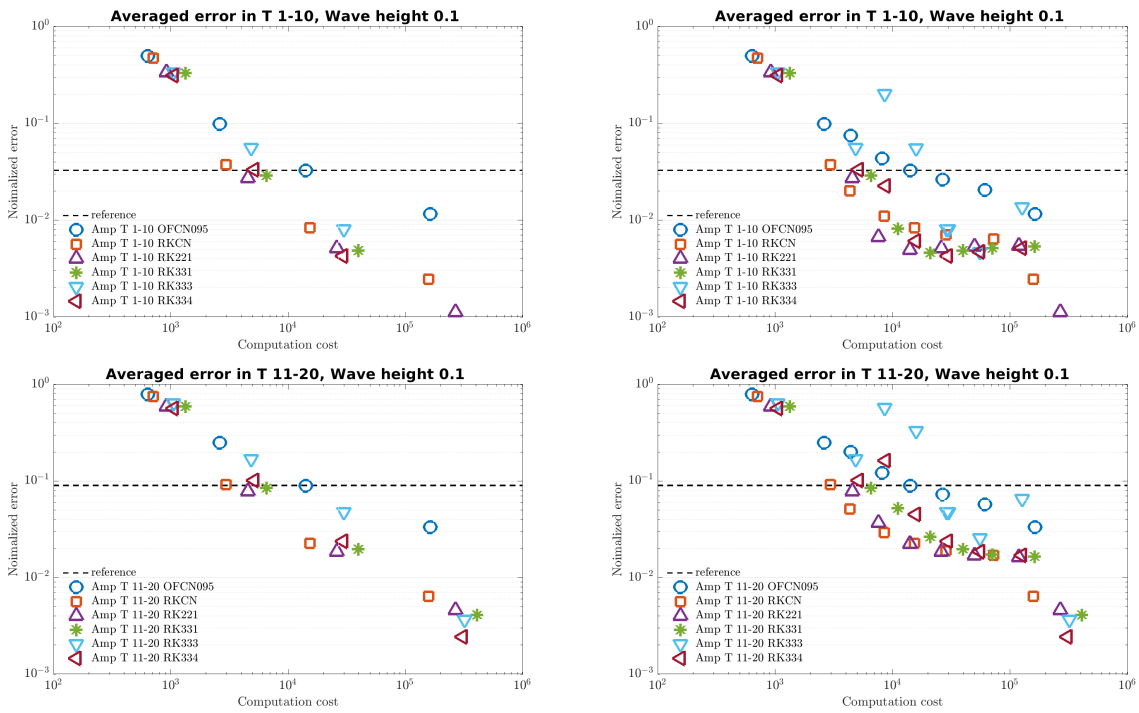


Figure 4.6. The comparison of the efficiency of the solvers, wave height H0.1, Dynamic mesh

References

[1] Y. J. Kim, Y.-M. Choi, B. Bouscasse, S. Seng, D. L. Touzé, Influence of interface treatment in wave propagation in cfd, 16èmes Journées de l'Hydrodynamique.

[2] W. Li, Efficient Implementation of High-Order Accurate Numerical Methods on Unstructured Grids, Series on Computational Methods in Mechanics and Thermal Science, Berlin, Heidelberg: Springe, 2014.

[3] H. Bihs, A. Kamath, M. A. Chella, A. Aggarwal, Øivind A. Arntsen, A new level set numerical wave tank with improved density interpolation for complex wave hydrodynamics, Computers & Fluids 140 (2016) 191 – 208. doi:https://doi.org/10.1016/j.compfluid.2016.09.012.
URL http://www.sciencedirect.com/science/article/pii/S0045793016302729

[4] Q. Wang, Y.-X. Ren, J. Pan, W. Li, Compact high order finite volume method on unstructured grids iii: Variational reconstruction, Journal of Computational Physics 337 (2017) 1 – 26. doi:https://doi.org/10.1016/j.jcp.2017.02.031.
URL http://www.sciencedirect.com/science/article/pii/S0021999117301249

[5] T. Martin, I. Shevchuk, Implementation and validation of semi-implicit weno schemes using openfoam, Computation 6 (1).

[6] V. Titarev, E. Toro, Ader schemes for three-dimensional non-linear hyperbolic systems, Journal of Computational Physics 204 (2) (2005) 715 – 736. doi:https://doi.org/10.1016/j.jcp.2004.10.028.
URL http://www.sciencedirect.com/science/article/pii/S0021999104004358

[7] M. Dumbser, T. Schwartzkopff, C.-D. Munz, Arbitrary high order finite volume schemes for linear wave propagation, in: E. Krause, Y. Shokin, M. Resch, N. Shokina (Eds.), Computational Science and High Performance Computing II, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 129–144.

[8] J. Butcher, Diagonally-implicit multi-stage integration methods, Applied Numerical Mathematics 11 (5) (1993) 347 – 363. doi:https://doi.org/10.1016/0168-9274(93)90059-Z.
URL http://www.sciencedirect.com/science/article/pii/016892749390059Z

[9] H. Bijl, M. H. Carpenter, V. N. Vatsa, C. A. Kennedy, Implicit time integration schemes for the unsteady compressible navier–stokes equations: Laminar flow, Journal of Computational Physics 179 (1) (2002) 313 – 329. doi:https://doi.org/10.1006/jcph.2002.7059.
URL http://www.sciencedirect.com/science/article/pii/S0021999102970592

[10] Y. P. Marx, Time integration schemes for the unsteady incompressible navier-stokes equations, Journal of Computational Physics 112 (1) (1994) 182 – 209. doi:https://doi.org/10.1006/jcph.1994.1091.
URL http://www.sciencedirect.com/science/article/pii/S0021999184710916

[11] V. Vuorinen, J.-P. Keskinen, C. Duwig, B.-J. Boersma, On the implementation of low-dissipative runge-kutta projection methods for time dependent flows using openfoam (r), Computers & Fluids 93 (2014) 153–163. doi:10.1016/j.compfluid.2014.01.026.

[12] V. D'Alessandro, L. Binci, S. Montelpare, R. Ricci, On the development of openfoam solvers based on explicit and implicit high-order runge–kutta schemes for incompressible flows with

heat transfer, Computer Physics Communications 222 (2018) 14 – 30. doi:https://doi.org/10.1016/j.cpc.2017.09.009.
URL http://www.sciencedirect.com/science/article/pii/S0010465517302965

[13] E. Hairer, G. Wanner, Solving Ordinary Differential Equations II. Stiff and Differential-Algebraic Problems, Vol. 14, 1996. doi:10.1007/978-3-662-09947-6.

[14] P. D. Boom, D. W. Zingg, Optimization of high-order diagonally-implicit runge–kutta methods, Journal of Computational Physics 371 (2018) 168 – 191. doi:https://doi.org/10.1016/j.jcp.2018.05.020.
URL http://www.sciencedirect.com/science/article/pii/S0021999118303206

[15] A. Prothero, A. Robinson, On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations, mathematics of computation 28 (125) (1974) 145–162.

[16] C. A. Kennedy, M. Carpenter, Diagonally implicit runge-kutta methods for ordinary differential equations. a review.

[17] R. Alexander, Diagonally implicit runge–kutta methods for stiff o.d.e.'s, SIAM Journal on Numerical Analysis 14 (6) (1977) 1006–1021. doi:10.1137/0714068.
URL https://doi.org/10.1137%2F0714068

[18] J. Hrvoje, Error analysis and estimation for the finite volume method with applications to fluid flows, Ph.D. thesis, Imperial College of Science, Technology and Medicine (1996).

[19] I. Demirdžić, M. Perić, Space conservation law in finite volume calculations of fluid flow, International Journal for Numerical Methods in Fluids 8.

[20] J. H. FERZIGER, M. Peric, Computational Methods for Fluid Dynamics / J.H. Ferziger, M. Peric., Vol. 3, 2002. doi:10.1007/978-3-642-56026-2.