Numerical Simulation of Nonlinear Water Waves based on Fully Nonlinear Potential Flow Theory in OpenFOAM®-Extend

Arshad Mehmood [a], *David I. Graham* [a], *Kurt Langfeld* [a], *Deborah M. Greaves* [b]
[a] School of Computing, Electronics and Mathematics, Plymouth University, UK
[b] School of Marine Science and Engineering, Plymouth University, UK

ABSTRACT

We develop a free surface tracking solver for numerical simulation of unsteady irrotational fully non-linear water waves in a freely available open-source computational fluid dynamics toolbox OpenFOAM®-Ext, which is community-driven release of OpenFOAM®. The solver is based on the solution of the Laplacian of the velocity potential with moving free surface. The free surface is tracked by solving the kinematic boundary condition based on the normal flux out of the surface. We also develop the necessary boundary conditions for the realistic wave generation at inlet and the absorption boundary condition at the outlet boundary. To avoid numerical instability, a 5-point smoothing technique is used to smooth the free surface elevation. Solution of Laplace's equation for the velocity potential, the non-linear free surface boundary conditions, the wave generation and the absorption boundary conditions are all not part of the standard OpenFOAM® distribution. The potential flow solver is able to simulate large amplitude standing and progressive waves. We validate the solver by comparing the numerical results with analytical results for second order standing waves, and progressive waves with experimental results and satisfactory agreement is found.

KEY WORDS: free surface flows; surface tracking; finite volume method; nonlinear potential flow theory; Laplace's equation.

INTRODUCTION

The numerical simulation of unsteady free surface waves has received considerable attention in computational fluid dynamics because of the challenge in determining the free surface. A variety of methods have been developed and can be classified namely as "surface capturing" and "surface tracking". In surface capturing methods, the solution is obtained in a computaional domain that covers at least two fluids. This method mostly solves the Navier-Stokes equations in the computational domain with an additional equation for a scalar field to determine the location of the free surface at each instant in time. This approach can easily capture complicated flow phenomena e.g waves overturning, wave breaking and bubble generating. However solving two fluid regions with an additional scalar transport equation results in high computational cost. Moreover, the scalar field also suffers from numerical diffusion. In the surface tracking approach, the flow is solved only for one fluid in a computational domain with the free surface treated as a moving upper boundary. Among this method, many reserachers (Mayer, Garapon and Sorensen, 1998; Zwart, Raithby and Raw, 1999; Muzaferija and Perič, 1997), solve the full Navier-Stokes equations without making any simplifying flow assumptions, whilst others (Greaves, Wu, Borthwick and Taylor, 1997; Ma, Wu and Taylor, 2001; Santos and Greaves, 2006) solve a Laplacian equation assuming the fluid to be inviscid and the flow to be irrotational and incompressible, as is the case of the method presented herein. Following the irrotational flow assumption, the solution has been obtained by various numerical methods including boundary element method (Wu and Taylor, 1995), finite element method (Greaves, Wu, Borthwick and Taylor, 1997; Ma, Wu and Taylor, 2001; Santos and Greaves, 2006) and by finite volume method (Mehmood, Graham, Langfeld and Greaves, 2015). The method is efficient and can follow the evolution of a simple free surface profile very accurately. This approach is accurate and less expensive, however, restricted by when the free surface breaks or overturns.

In the present paper, we have updated the moving boundary finite volume formulation (Mehmood, Graham, Langfeld and Greaves, 2015) for two-dimensional fully nonlinear time-dependent free surface waves. The current nonlinear full potential flow (NLPF) solver has been developed in OpenFOAM®-Extend environment. In recent years, a wide range of community is using the open source CFD library OpenFOAM®-Extend for various applications. Among them hydrodynamic group is also an active one which uses it for coastal related applications. Different toolboxes namely waves2Foam (Jacobsen, Fuhrman and Fredsoe, 2012) and ihFoam (Higuera, Lara and Losada, 2013) have been created for the community to generate and absorb free suface flows. In waves2Foam, waves are generated actively while abosrbed using wave relaxation zones, wheras in ihFoam, waves are gnerated and absorbed actively, thus reducing the computational cost. However, both these tools are using the existing interFoam as a solver while solving three-dimensional Navier-Stokes equations for two fluids. Both these tools are used to create very accurately the wave profiles and different sea conditions. However, in case of large domains and specially the

domains where waves do not steepen and break, these tools will result in high computational cost. A key element for coastal engineering studies currently lacking within OpenFOAM®-Extend is the ability to generate and propagate waves using less computational resources. In this article, we describe a solver that will generate and propogate waves with less computational cost. The solver will make calculations upto the point where the wave begins to overturn. It has been coded from scratch using the OpenFOAM®-Extend functions to realistically generate and propogate waves.

Particular reason of developing this solver in OpenFOAM®-Extend is to couple this solver with the existing incompressible and compressible Navier-Stokes solvers. Since the method can not predict cases of wave breaking, bubble capturing which will be made possible by coupling it with the already built-in incompressible and compressible Navier-Stokes (NS) solvers through a proper boundary condition that will enable simulations of the full range of wave conditions. Additional advantages include utilization of their built-in utilities including parallelization, various time and spatial discretisation schemes, meshing types, mesh motion solvers, selection of different waves conditions. The current wave generating and absorption boundary conditions have been implemented following OpenFOAM®-Extend C++ class structure and hence can be updated easily. The adoption of object oriented programming imlpies that an extension of the wave theories merely requires knowledge of the variation of u_x as a function of space and time: hence additional wave theories can easily be added.

In this method, the Laplace's equation for the velocity potential is solved inside the flow domian with the Neumann and Dirichlet type boundary conditions on the boundaries. Kinematic and dynamic boundary conditions are imposed on the free surface. The moving free surface is tracked as a part of the solution along with the flow velocity. In the implemented scheme, the free surface is allowed to deform and the inside mesh moves to accommodate the free surface deformation at each time step. Modeling of the nonlinear free surface waves which satisfies the Laplaces of the velocity potential inside the fluid domain with the inclusion of kinematic, dynamic, waves generating and radiation boundary conditions were not part of the standard distribution. OpenFOAM®-Extend uses a finite volume discretisation approach on unstructured meshes consisting of arbitrary convex polyhedrals. In the current simulations, we used structured meshes, since the solver will make the calculations up to the point when the free surface breaks or overturns. For a full description of space and time integration schemes in OpenFOAM®-Extend, readers are referred to (e.g. Jasak, 1996).

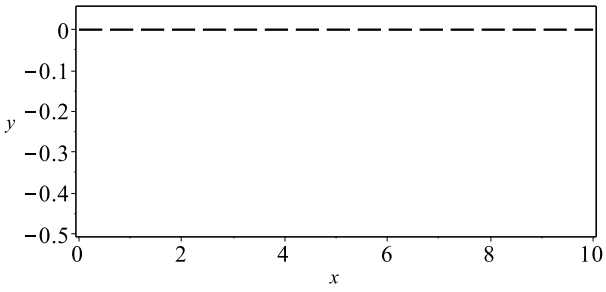

Fig. 1 Computational domain.

MATHEMATICAL FORMULATIONS

Assuming the fluid to be inviscid, incompressible, and the flow to be irrotational, the governing equation of the fluid flow is written in the form as:

$$\nabla^2\phi(\boldsymbol{x}, t) = 0, \tag{1}$$

where $\boldsymbol{x} = (x, y)$ and ϕ represents the velocity potential. We consider a right-hand Cartesian coordinate system O-xyz, with the origin placed at the left of the domain and on the undisturbed free surface represented by dotted line as shown in Fig. 1. The y-axis is negative downward.

Eq. 1 is solved by specifying on all rigid surfaces, the Neumann type boundary conditions and at the free surface, the dynamic and kinematic boundary conditions which are given as, respectively:

$$\frac{\partial\phi}{\partial t} = -g\eta - \frac{1}{2}\nabla\phi.\nabla\phi, \tag{2}$$

$$\frac{\partial\eta}{\partial t} = \frac{\partial\phi}{\partial y} - \frac{\partial\phi}{\partial x}\frac{\partial\eta}{\partial x}, \tag{3}$$

where η and g denotes the free surface elevation and the gravitational acceleration, respectively. ∇ is the gradient operator vector and t is time. We compute the fluid velocities by its gradient, $\mathbf{u} = \nabla\phi$. The kinematic boundary condition Eq. 3 based on the volume flux can be rewritten as Mayer, Garapon and Sorensen (1998):

$$\frac{\partial\eta}{\partial t} = \frac{\mathbf{u}.\mathbf{n}}{n_y}, \tag{4}$$

where $\mathbf{u}.\mathbf{n}$ is the normal volumetric flux out of the surface, and n_y is the vertical component of the unit normal pointing out of the fluid domain.

Eq. 4 expresses the temporal change of the free surface geometry based on the volume flux, resulting in a change in mesh. Note that in the OpenFOAM®-Extend implemented finite volume method, the flow variables are defined at cell centres. Moreover, the fluxes and the boundary conditions are computed at the face centre. However, in order to update the mesh, the change in the free surface geometry must be specified at cell vertices. Therefore, we interpolate the normal volumetric flux ($\mathbf{u}.\mathbf{n}$) from the face centre to the cell vertices and then integrate the values according to Eq. 4. Furthermore, the velocities are not defined at cell face centre, they are only defined at cells centres. We therefore extrapolate the velocities using first-order extrapolation to the face centre. The dynamic boundary condition Eq. 2 is then solved to give the velocity potential on the free surface for the next time step.

Knowing the free surface vertex points, the inner domian is solved using the OpenFOAM®-Extend standard solvers. In the developed method, the grid points at the free surface are moved explicitly and the interior grid points are moved based on "displacementLaplacian" solver, for which quadratic inverseDistance diffusivity was selected (most appropriate solver based on the author experiments). Since the flow variables are not computed at cell vertices, they are interpolated either from face centers or cell centers, causing saw-tooth free surface. To overcome this numerical instability, a 5-pt smoother has been used. The smoothed value at each vertex i on the free surface is calculated using the formula (Bai, Mingham, Causon and Qian, 2010): $f_i = (-f_{i-2} + 4f_{i-1} + 10f_i + 4f_{i+1} - f_{i+2})/16$, where subscripts $i-2$, $i-1$, i, $i+1$, $i+2$ denote the values at corresponding neighbouring vertices. It is also worth mentioning that the while using the smoothing, the 1st and last 2 points at the edges need special attention. For standing waves,

these points were used as a symmetry to the inner points whereas for progressive waves, these points were computed from inner domain using 3rd-order approximation as:

$$f_{i-1} = 3f_i - 3f_{i+1} + f_{i+2}, \tag{5}$$

$$f_{i-2} = 6f_i - 8f_{i+1} + 3f_{i+2}. \tag{6}$$

We also developed different necessary boundary conditions for the simulation of various test cases in the OpenFOAM®-Extend envioronment. The boundary conditions implemented in the present study are defined as:

1. inlet: The condition on this boundary can be expressed as

$$\frac{\partial \phi}{\partial x} = F(y,t), \tag{7}$$

where F(y,t) is zero for the standing wave cases (i.e zero gradient) and for the progressive waves, a horizontal component of the velocity of the known wave theory is imposed as:

$$\frac{\partial \phi}{\partial x} = u_x(y,t). \tag{8}$$

The applied velocity $u_x(y,t)$ may be either Stokes velocity profiles or it could be like physically piston wave makers. Moreover, the amplitude of the imposed boundary velocity is superposed by a ramp function. Currently sinusoidal and linear ramp functions are supported.

2. freeSurface: At this boundary, the kinematic boundary condition Eq. 4 and a dynamic boundary condition Eq. 2 are applied.
3. bottom: a zero-gradient boundary condition is applied for all cases.
4. outlet: In addition to above conditons, a proper boundary condition needed to be specified at the downstream end in order to remove reflections as much as possible. We implemented the Sommerfeld condition

$$\frac{\partial \phi}{\partial t} + c\frac{\partial \phi}{\partial n} = 0, \tag{9}$$

where c is the phase velocity of the linear harmonic wave and n is the normal pointing out of the surface of the downstream boundary. We consider the parameter c as the velocity of linear harmonic wave $c = \sqrt{\frac{g * \tanh(kH)}{k}}$, where H is the water depth, k the wave number. Morover, a zero gradient boundary condition is applied for the standing waves test cases.

Sequence of the Solution Procedure

The sequence followed for integrating the system of fluid motion and free surface from time step t^n to t^{n+1} is as follows.

1. Generate the grid.
2. Apply the boundary conditions.
3. Solve Laplace's equation for the velocity potential.
4. Compute the required variables (i.e., velocities, fluxes).
5. Solve the kinematic boundary condition using Eq. 4, yielding new free surface geometry.
6. Update the grid based on the surface elevation computed in the previous step (step 5).
7. For the new grid, compute the velocity potential on the free surface using the dynamic boundary condition Eq. 2, thus providing the new velocity potential for the next time step and accordingly update the boundary conditions on the remaining 3-boundaries.
8. Advance the solution in time by repeating the procedure (step 4-8) in every time step.

RESULTS AND DISCUSSION

The developed method was tested using a number of linear standing wave test cases having analyical solutions in Mehmood, Graham, Langfeld and Greaves (2015). Here, we present test cases on more difficult problems. We consider test cases involving standing and progressive waves of high amplitudes.

Standing Waves

At t=0, the boundary condition at free surface was defined as

$$\phi(x,y) = 0, \qquad \eta(y) = F(x),$$

where $F(x)$ is a function describing the initial shape of the free surface based on the linear and 2nd-order approximation. On the remaining boundaries, a zero gradinet boundary condition is specified. Here, we consider test cases having relatively large wave amplitudes and small water heights where we can observe the nonlinear effects. We consider a standing sinusoidal wave with amplitudes $a = 0.01$ m, wavelength $\lambda = 1.0$ m and mean water depth $H = 0.1$ m as simulated by Santos and Greaves (2007). The initial profile of the free surface at $t = 0$ was set according to 2nd-order standing wave Eq.10 (Cozzi, 2010) as shown in Fig. 2 and the fluid is initially at rest. The time step was set to $\Delta t = 0.002$.

$$\eta(x,t) = a\cos(kx)\cos(\omega t) + \frac{\pi a^2}{\lambda}\left[\cos^2(\omega t) - \frac{1}{4\cosh^2(kH)} + \frac{3\cos(2\omega t)}{4\sinh^2(kH)}\right]\cos(2kx). \tag{10}$$

The time trace of the wave elevation measured at the centre of the tank is shown in Fig. 3. The linear and 2nd-order Airy solutions are also plotted for comparison. Note that the values have been non-dimensionalized as follows: η/a, where η is the calculated wave elevation and a is the initial wave amplitude. Similarly on the x-axis time is non-dimensionalized by the wave period which is calculated according to relation $T = \frac{2\pi}{\sqrt{gk\tanh(kH)}}$. This plot shows non-linear behaviour,

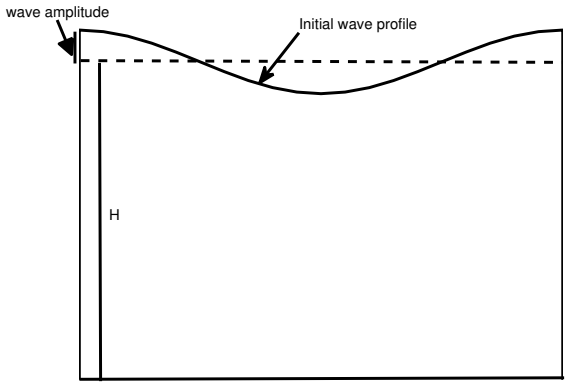

Fig. 2 Initial profile of the standing wave.

showing higher crests, the phenomenon also observed by Santos and Greaves (2007) (Fig. 12). A second cases was also simulated with higher wave amplitudes $a = 0.015$ m while keeping all other parameters the same. The corresponding non-dimensionalized time history of the wave elevation calculated at the middle of the tank is shown in Fig. 4, again illustrating non-linear behaviour. From these figures, we can see that the overall appearance of the wave profile is very similar between the theoretical and the calculated ones, but a difference exists at the peak levels of the wave. The source of this difference seems to come from the wave nonlinearity for the considered wave parameters, which can not be captured using 1st and 2nd-order approximations. Qualitatively, the same non-linear behaviour was observed by Santos and Greaves (2007) (Fig. 13).

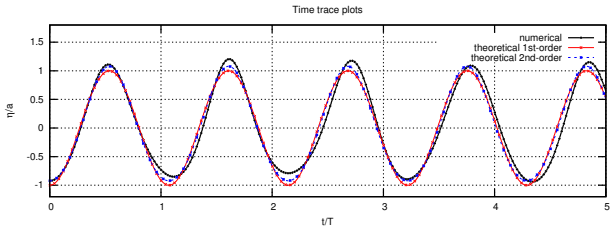

Fig. 3 Time history of free surface elevation at the center of the domain for $a = 0.02$ m, $H = 0.1$ m.

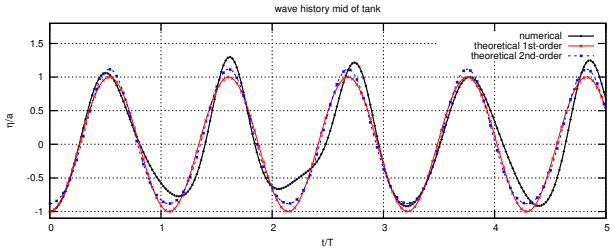

Fig. 4 Time history of free surface elevation at the center of the domain for $a = 0.015$ m, $H = 0.1$ m.

Progressive Waves

For progressive waves, we impose a non-zero gradient boundary condition both at inlet and outlet. At inlet, waves are generated by specifying the expression of the horizontal component of the velocity Eq. 8 of the known wave theories. The expression could have the effect of water depth, where the wave maker boundary condition acts as a sinusoidal wave paddle or it could be of the form where the wave paddle moves backward and forward with a constant speed all along the water height, having no effect of water depth. At outlet, we impose a radiation boundary condition Eq. 9 to absorb the incoming waves.

Firstly, we performed numerical simulations for progressive waves by specifying at inlet boundary, the horizontal component of the velocity from 1st order Stokes theory. The length of the tank was considered as $L = 10$ m, long enough that will enable us to simulate waves over a long period of simulated time and to see the effect of any reflection coming back from the outlet boundary. We consider regular incident wave of amplitude $a = 0.01$ m, wave period $T = 1.5$s. The depth of water was set as $H = 1.5$ m; and wave number was calculated from the the linear dispersion relation $\omega^2 = gk\tanh(kh)$, with k being the wave number and g the acceleration due to gravity. Fig. 5 show the time

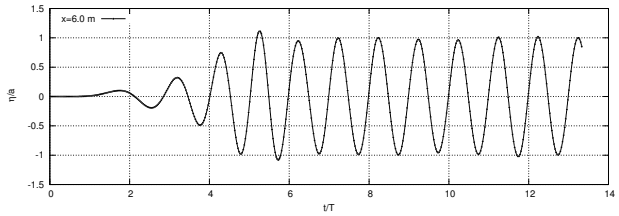

Fig. 5 Time history of free surface elevation at location $x = 6.0$ m (from inlet boundary) for $a = 0.01$ m, $H = 1.5$ m.

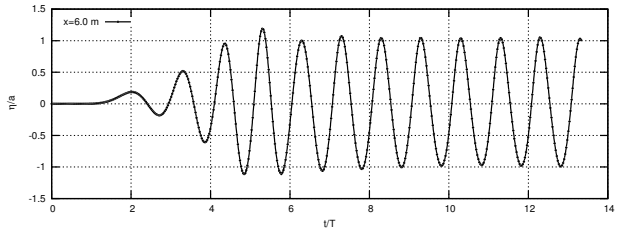

Fig. 6 Time history of free surface elevation at location $x = 6.0$ m (from inlet boundary) for $a = 0.06$ m, $H = 1.0$ m.

history of the nondimensionalized wave elevation at location $x = 6.0$ m. The wave elevation and time are nondimensionalized by the incident wave amplitude and wave period of the imposed wave, respectively. In the second case, we increase the wave amplitude of the incident wave to $a = 0.06$ m, and reduced the water depth to $H = 1.0$ m, and kept $T = 1.5$s. The non-dimensionalized time history of the wave elevation calculated at location $x = 6.0$ m is shown in Fig. 6. Moreover, in both test cases, the incident wave velocity was "ramped" using a sinusoidal function ($\sin(2\pi/(4.0 * T))$, where T is the wave period of the incident wave. The ramping is done from $t = 0$ to $t = T$, so that the wave maker imposes only a little disturbance at first and then gradually builds to the full disturbance level. From both these plots, we can see that the solver accurately capture the wave period and the amplitude of the wave elevation of the unsteady progressive waves. Moreover, the simulations were run for 20s long enough where the waves will be reflected back and forth from the outlet and inlet boundaries and passing through locations x=6.0 m. From the periodic response of wave elevations as shown in Figs. 5 and 6, we can see that there is no significant reflection coming back from the outlet boundary or re-reflection from the inlet boundary.

Next, we compare our simulation results with the experiment conducted by Gao (2003). The numerical simulations for this test case have also been carried out by Qian, Causon, Mingham and Ingram (2006) and Bai, Mingham, Causon and Qian (2010) using full Navier-Stokes(NS) computations. In the current simulations, regular waves were generated in a wave tank long $L = 8.85$ m, water depth $H = 0.28$ m, wave period as $T = 1.0$ s and wave amplitude $a = 0.025$ m. The waves were generated by prescribing a time varying velocity according to the expression:

$$u_x = a\omega \sin(kx - \omega t). \tag{11}$$

For this test case, a linear ramped function is superposed from $t = 0$ to $t = T$. A total of 12 s was simulated to compare with the available experimental data. In the experiment, the wave elevation was recorded at various locations along the wave tank. We obtained the solution for two different mesh discretisation (refined mesh 708 x 33 x 1) and (coarse mesh 354 x 17 x 1). The current numerical results (fine-mesh) are shown by black solid line and that of coarse mesh by blue dotted line and the data from the experiment is shown by red line. Fig. 7 show the comparison for free surface elevation with the experimental data at the same

locations (x = 0.55 m,x = 3.55 m,x = 5.45 m-from the inlet of the tank). The agreement with the experimental data for locations at $x = 0.55$ m as shown in Fig. 7a, is excellent, even better than the NS simulations by Qian, Causon, Mingham and Ingram (2006) (Fig. 7(a)) and Bai, Mingham, Causon and Qian (2010) (Fig. 13(a)). The wave extremes are well captured even with the coarsest discretisation used. The results at locations $x = 3.55$ m and at $x = 5.45$ m are also in good agreement with the experimental results.

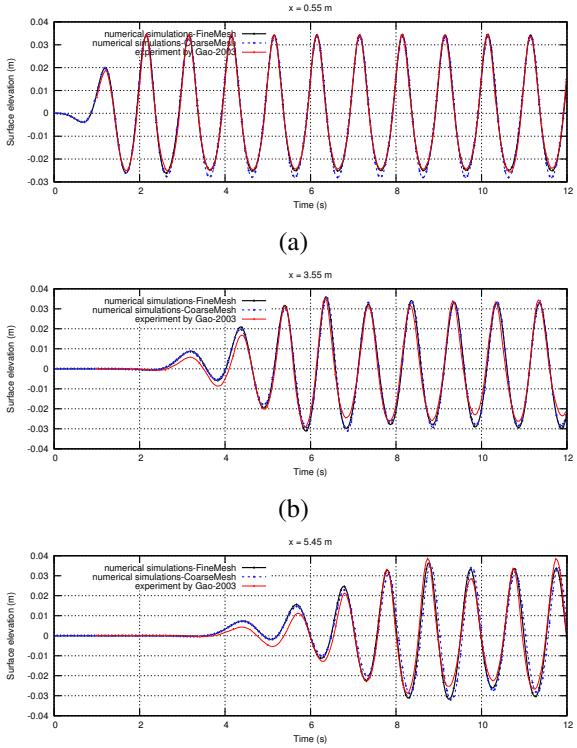

Fig. 7 Comparison between experimental data and current numerical simulations for free surface elevation at locations. (a) $x = 0.55$ m, (b) $x = 3.55$ m, (c) $x = 5.45$ m.

Parallel Computations

The main reason to create the existing solver in OpenFOAM environment was to use the OpenFOAM built-in capabilities and also to couple with the existing incompressible and compressible Navier Stokes solver for the full range of wave conditions. To show that the new generated solver and all of its required boundary conditions work in parallel, we run some simulations here. All computations were carried out on a Intel® Core i7-4790 CPU workstation with 16 Gb memory and a 3.6 GHz Power 8 processor. In the current simulations, we used 'simple" method of decomposition in the OpenFOAM®-Extend available decomposition environments. Fig.8 shows the speed-up curves for two different grid sizes (Grid-I-708 x 33 x1) and (Grid-II- 354 x 17 x 1) along with the linear (ideal) speed-up curve. The speed-up is defined as:

$$\text{Speed-up} = \frac{\text{Simulation time on 1 processor}}{\text{Simulation time on } N_p \text{ processors}}, \tag{12}$$

where N_p is the number of processors. It is observed that the Grid-I shows better performance than the Grid-II because the ratio of computation to communication (per processor) becomes larger for Grid-I. The plot show that solver runs in parallel, although does not show a good parallel speed-up compare to the ideal one. However, it is also worth to mention that current computations are performed on small domains where the grid sizes are small. It has been known that in order to get good performance in OpenFOAM, at least 20,000 cells per processor are needed.

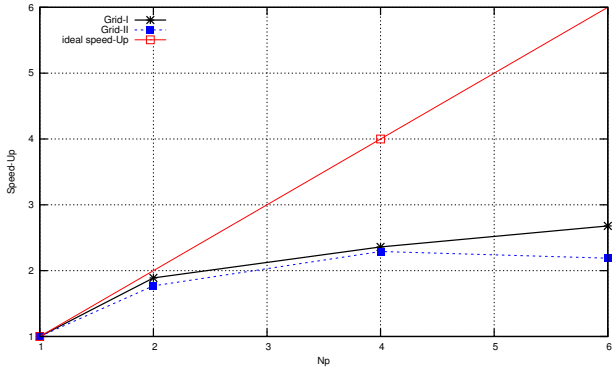

Fig. 8 Speed-up trends of the current solver for different grid sizes.

CONCLUSIONS

In this paper, a freely available open source code OpenFOAM®-Extend has been extended to actively generate and absorb single phase free surface flows using the nonlinear full potential flow theory. The method has been used to simulate various kinds of standing and progressive waves in a rectangular tank. The solver has been validated by application to a number of test cases, ranging from shallow water standing waves to different wave amplitudes progressive waves. The numerical simulation results agreed well with the available analytical solutions, and excellent agreement with the experimental data shows successful implementation of the solver and implemented boundary conditions in OpenFOAM®-Extend. The results show that the present scheme could be used as a solver to simulate the waves in large domains with low computational cost and can be easily extended to three-dimensional nonlinear free surface waves. The developed solver and the associated boundary conditions will be released as an open-source for the marine and offshore community.

ACKNOWLEDGMENTS

The author gratefully acknowledges the EPSRC grant reference number EP/K038303/1 for the financial support for the current work.

REFERENCES

Bai, W, Mingham, CG, Causon, DM, and Qian, L (2010). "Finite volume simulation of viscous free surface waves using the Cartesian cut cell approach," *Intl J Numerical Methods in Fluids*, 63, 69–95.

Cozzi, O (2010). "Free surface flow simulation:correcting and benchmarking the ALE method in Code_Saturne,"MS Thesis, The University of Manchester.

Gao, F, (2003). "An efficient finite element technique for free surface flow,"Ph.D. thesis, Brighton University, UK.

Greaves, DM, Wu, G, Borthwick, A, and Taylor, ER (1997). "A moving boundary finite element method for fully nonlinear wave simulations," *Journal of Ship Research*, 41(3), 181–194.

Higuera, P, Lara, JL, and Losada, IJ (2013). "Realastic wave generation and active wave absorption for Navier-Stokes models applications to OpenFOAM®," *Intl J Numerical Methods in Fluids*, 70(9), 1073–1088.

Jacobsen, NG, Fuhrman, DR, and Fredsoe, J (2012). "A wave generation toolbox for the open-source CFD library: OpenFoam®," *Intl J Numerical Methods in Fluids*, 70(9), 1073–1088.

Jasak, H (1996). "Error analysis and estimation for the finite volume method with applications to fluid flows,"Ph.D. Thesis, Imperial College of Science, Technology and Medicine

Mayer, S, Garapon, A and Sorensen, LS (1998). "A fractional step method for unsteady free surface flow with applications to non-linear wave dynamics," *Intl J Numerical Methods in Fluids*, 28(2), 293–315.

Muzaferija, S, and Perič M (1997) "Computation of free-surface flows using the finite-volume method and moving grids," *Numerical Heat Transfer, Part B: Fundamentals: An Intl J of Computation and Methodology*, 32(4), 369–384.

Ma, QW, Wu, GX, and Taylor, ER (2001). "Finite element simulation of fully non-linear interaction between vertical cylinders and steep waves. Part 1: Methodology and numerical procedure," *Intl J Numerical Methods in Fluids*, 36(3), 265–285.

Mehmood, A, Graham, DI, Langfeld, K, Greaves, DM, (2015). "OpenFoam® Finite Volume Method Implementation of a Fully Nonlinear Potential Flow Model for Simulating Wave-Structure Interactions," *Proc 25th Intl Ocean and Polar Engineering Conference*, Kona, USA, 663–667.

Qian, L, Causon DM, Mingham, CG and Ingram, DM (2006). "A free-surface capturing method for two fluid flows with moving bodies," *Proc. R. Soc.*, 462, 21–42.

Santos, CM, and Greaves, DM (2006). "A mixed Lagrangian-Eulerian method for non-linear free surface flows using multigrid on hierarchical Cartesian grids," *Computers & Fluids*, 36(5), 914–923.

Wu, GX, and Taylor, ER (1995). "Time stepping solutions of the two-dimensional nonlinear wave radiation problem," *Ocean Engng*, 22(8), 785–798.

Wu, GX, and Taylor, ER (1994). "Finite element analysis of twodimensional non-linear transient water waves," *Applied Ocean Research*, 16(6), 363–372.

Zwart, PJ, Raithby, GD, and Raw, MJ (1999). "The integrated space-time finite volume method and its application to moving boundary problems," *J of Computational Physics*, 154, 497–519.