

Seakeeping with the semi-Lagrangian particle finite element method

Prashanth Nadukandi¹  · Borja Servan-Camas¹  ·
Pablo Agustín Becker¹  · Julio Garcia-Espinosa¹ 

Received: 5 April 2016 / Revised: 16 July 2016 / Accepted: 20 July 2016 / Published online: 29 July 2016
© OWZ 2016

Abstract The application of the semi-Lagrangian particle finite element method (SL–PFEM) for the seakeeping simulation of the wave adaptive modular vehicle under spray generating conditions is presented. The time integration of the Lagrangian advection is done using the explicit integration of the velocity and acceleration along the streamlines (X-IVAS). Despite the suitability of the SL–PFEM for the considered seakeeping application, small time steps were needed in the X-IVAS scheme to control the solution accuracy. A preliminary proposal to overcome this limitation of the X-IVAS scheme for seakeeping simulations is presented.

Keywords Particle finite element method · Semi-Lagrangian formulations · Seakeeping

1 Introduction

The particle finite element method (PFEM) [18] is a versatile framework for the analysis of fluid–structure interaction problems. The PFEM combines Lagrangian particle-based techniques with the advantage of the integral formulation of the Finite Element Method (FEM). It has been shown [13–

15, 21, 27, 28] to successfully simulate a wide variety of complex engineering problems, e.g., free-surface/multi-fluid flows with violent interface motions, multi-fluid mixing and buoyancy-driven segregation problems etc. The latest development within the framework of the PFEM is the X-IVAS (eXplicit Integration along the Velocity and Acceleration Streamlines) scheme [16]. It is a semi-implicit scheme built over a semi-Lagrangian (SL) formulation of the PFEM.

In this paper we present the application of the SL–PFEM using the X-IVAS scheme¹ to the seakeeping simulation of the wave adaptive modular vehicle (WAM-V) [32]. The objective of the WAM-V is to be a lightweight watercraft capable of moving fast and efficiently on the surface of the sea. The motivation is that this technology could someday be used for quickly deploying research and reconnaissance equipment to far-flung locations and for search-and-rescue operations in the sea. To operate safely close to the shore, the WAM-V has an inflatable catamaran-style hull that displaces only a few feet below the waterline. To travel efficiently with low wave resistance in rough seas, it is designed to have a flexible wave-adaptive sub-structure. This feature allows it to surf on top of the waves rather than cut through them.

Objective The purpose of this paper is first to show the suitability of the SL–PFEM method to simulate the afore-said seakeeping problem under spray generating conditions and secondly, to identify some limitations of the SL–PFEM related to the trajectory approximation obtained using the X-IVAS scheme when the underlying physics imposes wave-

✉ Prashanth Nadukandi
npras@cimne.upc.edu; nadukandi@gmail.com

Borja Servan-Camas
bservan@cimne.upc.edu

Pablo Agustín Becker
pbecker@cimne.upc.edu

Julio Garcia-Espinosa
julio@cimne.upc.edu

¹ Centre Internacional de Mètodes Numèrics en Enginyeria (CIMNE), Edifici C1, Gran Capitan s/n, 08034 Barcelona, Spain

¹ The SL–PFEM method using the X-IVAS scheme has been called the particle finite element method second generation (PFEM-2) by the original authors [16]. However, we choose the former nomenclature (SL–PFEM + X-IVAS) to acknowledge its connection (see Sect. 2) with semi-Lagrangian schemes.

like motion. Finally, the work presents a preliminary proposal to overcome the identified limitation.

Seakeeping simulations usually involve water wave motion. Stokes waves propagating on the water surface is an interesting example where time integration schemes are coerced into using small time steps. The X-IVAS scheme is no exception; the hypothesis—the streamlines are a good approximation to the pathlines, fails for this example. Using second-order Stokes wave propagation as the model problem we illustrate the particle trajectory approximation in the X-IVAS scheme which explains why the underlying hypothesis fails. An alternative to the X-IVAS scheme is where the particles are driven by the acceleration in the latest configuration. The trajectory approximation driven by acceleration in the latest configuration is illustrated which shows a remarkable improvement compared to the one obtained by the X-IVAS scheme.

2 Semi-Lagrangian formulation

Semi-Lagrangian schemes can be classified into two groups based on the way the problem variables are stored in the formulation: the Eulerian storage (ES) group and the Lagrangian storage (LS) group. However, the keyword *semi-Lagrangian* almost always points to ES–SL schemes in the literature. The ES–SL schemes are extensions of the Courant–Isaacson–Rees [4] method for hyperbolic equations. Sawyer [29] and Robert [26] made seminal contributions to ES–SL schemes. ES–SL schemes are unconditionally stable and therefore allow for large time steps. The solution is defined on a mesh and the nodal values are stored. Lagrangian particles are used as an auxiliary tool to compute the advective time evolution of the variables. For each time step (say t^n to t^{n+1}) particles are initially placed at the mesh nodes and are transported backwards in time (for $t^{n+1} - t^n$ s) along the pathlines passing through these nodes at time t^{n+1} . The t^n solution at the terminal positions are then assigned to the t^{n+1} nodal values. The backward trajectories seldom end at existing nodes. The ES–SL schemes provide models for the a priori unknown pathlines and to determine the t^n solution at the terminal positions.

The simplest first-order ES–SL scheme traces back a straight line characteristic and use piecewise-linear interpolation of the t^n solution. Min and Gibou [22] proposed a second-order scheme by numerically tracing back curved characteristics and using higher order polynomial interpolation. Dupont and Liu [7] combined a first-order SL scheme with the back and forth error compensation and correction (BFEC) method [6] to obtain a second-order unconditionally stable scheme. Selle et al. used a similar approach to create a second-order scheme [30] using the MacCormack method [20] instead of the BFEC method. The initial con-

dition and the solution evolution in the ES–SL schemes are known up to the spatial resolution of a fixed mesh. Hence, the overall accuracy of the ES–SL schemes is subjected to numerical erosion.

On the contrary, a majority of the solution variables are stored with the particles in the LS–SL schemes and their advective evolution is computed in a Lagrangian manner. Necessarily implicit variables (NIV) like the pressure are defined on the mesh and are stored at the nodes. Implicit corrections (e.g., action of viscosity) and the evolution of NIV (e.g., pressure) are computed on the mesh. For each time step (say t^n to t^{n+1}) particles are transported forward in time (for $t^{n+1} - t^n$ s) along the pathlines passing through their respective positions at time t^n . The solution variables carried with the particles are updated by interpolating (usually piecewise-linear) implicit corrections (if any) at the updated locations.

The LS–SL schemes have their origin in the Particle-in-Cell (PIC) method [8, 11]. However the PIC method is an ES–SL scheme (solution is stored on the mesh) and it is not suitable to simulate incompressible flows. The material-point method [31] (MPM) is an improvement of the PIC method wherein each particle is endowed with a fixed point mass, a position, a stress and specific material parameters. In the MPM, the pathlines are approximated by tracing forward a simple straight line characteristic (see Zhang et al. [33] and Gelet et al. [9]).

The X-IVAS scheme is a LS–SL scheme wherein the streamlines computed using the t^n solution are used to approximate the pathlines. Unlike in the MPM, here particles carry with them only the intrinsic material and flow properties. This permits the user to insert or remove particles without affecting the extrinsic flow properties (e.g., total mass). Several benchmark CFD and FSI examples were simulated [13] with the SL–PFEM using the X-IVAS scheme and taking very large time steps, e.g., 10–15 times the standard Courant–Friedrichs–Lewy (CFL) stability limit [3]. Therein the simulation results of multi-fluid incompressible flows were compared [13] with those obtained with OpenFOAM® (openfoam.org). For similar accuracy and using a desktop computer for the simulations, the SL–PFEM using the X-IVAS scheme was reported [13] to be twice faster than OpenFOAM®. The latter uses the Finite Volume Method (FVM) for discretisation and the Volume Of Fluid (VOF) method for capturing the multi-fluid interfaces. Gimenez et al. did [10] similar performance studies for incompressible flow simulations using the SL–PFEM on parallel computers. They reported strong scalability of the SL–PFEM at par with OpenFOAM® (using the solver `pimpleFoam`); however, a three-fold gain in speed (wall-time) was reported for the former for a chosen level of accuracy.

Celledoni et al. showed [2] that second-order ES–SL schemes are effectively second-order approximations to standard exponential integrators. A similar connection between

the X-IVAS scheme (a LS–SL scheme) and exponential integrators was pointed out [23] by the first author, albeit using heuristic arguments. Exponential integrators are designed to integrate stiff initial value problems with large time-steps. This connection explains why the SL schemes admit very large time-steps without deteriorating the solution accuracy.

3 Semi-Lagrangian particle finite element method

Notation Vectors are written using bold italic font and matrices are written using bold upright font. The independent variables in Lagrangian kinematics are $\{\lambda, t\}$, where λ represents a label to identify particles and t represents the time elapsed after labeling. The primary dependent variable is the fluid particle trajectory denoted as $X(\lambda, t)$. The independent variables in Eulerian kinematics are $\{\mathbf{x}, t\}$, where \mathbf{x} denotes the spatial coordinates. The primary dependent variable is the fluid velocity $\mathbf{u}(\mathbf{x}, t)$.

Consider the Eulerian description of the incompressible Navier-Stokes equations.

$$\frac{\partial}{\partial t} \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} - \nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

where ν is the kinematic viscosity and $p(\mathbf{x}, t)$, $\mathbf{f}(\mathbf{x}, t)$ are the pressure (scaled by the density) and the external acceleration fields, respectively. The effective acceleration field $\mathbf{a}(\mathbf{x}, t)$ in the fluid domain is obtained from the momentum balance equation of the flow.

$$\mathbf{a} = \left[\frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla \right] \mathbf{u} = \nu \Delta \mathbf{u} - \nabla p + \mathbf{f} \tag{3}$$

Note that the functional dependence on the independent variables is suppressed in Eqs. (1), (2) and (3) for brevity.

The fundamental principle of kinematics relates the Eulerian description of the flow with the Lagrangian description as follows.

$$\frac{d}{dt} X(\lambda, t) = \mathbf{u}(X(\lambda, t), t) \tag{4}$$

$$\frac{d^2}{dt^2} X(\lambda, t) = \mathbf{a}(X(\lambda, t), t) \tag{5}$$

X-IVAS scheme The basic idea is to update the fluid particle position and velocity within a time-step $t^n \leq t \leq t^{n+1}$ using

$$\frac{d}{dt} X^h(\lambda, t) = \mathbf{u}^h(X^h(\lambda, t), t^n) = \mathbf{A}^n X^h(\lambda, t) + \mathbf{b}^n \tag{6}$$

$$\frac{d}{dt} U^h(\lambda, t) = \mathbf{a}^h(X^h(\lambda, t), t^n) = \mathbf{C}^n X^h(\lambda, t) + \mathbf{d}^n \tag{7}$$

where $\mathbf{u}^h(\mathbf{x}, t)$ and $\mathbf{a}^h(\mathbf{x}, t)$ denote spatially continuous piecewise linear approximations of the velocity and accel-

eration defined on a background simplicial mesh. Note that \mathbf{u}^h and \mathbf{a}^h at time $t = t^n$ are used. It follows that $\mathbf{u}^h(\mathbf{x}, t^n)$ and $\mathbf{a}^h(\mathbf{x}, t^n)$ can be expressed locally as linear functions of \mathbf{x} , i.e. $\mathbf{u}^h(\mathbf{x}, t^n) = \mathbf{A}^n \mathbf{x} + \mathbf{b}^n$ and $\mathbf{a}^h(\mathbf{x}, t^n) = \mathbf{C}^n \mathbf{x} + \mathbf{d}^n$. The matrices \mathbf{A}^n , \mathbf{C}^n and the vectors \mathbf{b}^n , \mathbf{d}^n are spatially piecewise constant and depend on the time t^n . The particle trajectory and its velocity computed in this manner are denoted as $X^h(\lambda, t)$ and $U^h(\lambda, t)$, respectively.

Nielson and Jung presented [24] formulas in 2D and 3D to compute the closed-form analytical solution of tangent curves for piecewise linear vector fields defined over simplicial meshes. Thus, the Nielson–Jung formulas can be used to compute the analytical solution of (6). Idelsohn et al. presented [16] a procedure to compute the analytical solution of (6) and (7) in 2D. However the Nielson–Jung formulas and the calculation procedure described by Idelsohn et al. to compute the analytical solution are not numerically stable; loss of significance occurs due to subtractive cancellations near removable singularities [12,23]. Recently, the first-author presented [23] numerically stable formulas in 2D and 3D for the closed-form analytical solution of (6) and (7). However, due to ease of implementation numerical sub-stepping procedures based on simple finite difference schemes (e.g. forward Euler) are often used to integrate Eqs. (6) and (7).

We briefly describe the algorithm (see Table 1 for an outline) to implement the SL–PFEM using the X-IVAS scheme. First the Lagrangian advection (see [23, Sect. 3] for details) of the particles: $X^h(\lambda, t^n) \rightarrow X^h(\lambda, t^{n+1})$ and $U^h(\lambda, t^n) \rightarrow U^h(\lambda, t^{n+1})$ are done using Eqs. (6) and (7). Then we project the data advected with particles onto a background finite element (FE) mesh. The data include the particle velocities, identities² and other problem dependent intrinsic variables. Using the particle velocities projected onto the FE mesh as the solution at the start of the time interval $t^n \leq t \leq t^{n+1}$, we solve the Stokes equations on the background FE mesh. For instance using the backward Euler time integration the semi-discrete Stokes system to be solved is

$$\widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1}) = \mathcal{P}^h U^h(\lambda, t^{n+1}) \tag{8}$$

$$\frac{\mathbf{u}^{h,n+1} - \widehat{\mathbf{u}}^{h,n+1}}{\Delta t^n} - \nu \Delta \mathbf{u}^{h,n+1} + \nabla p^{h,n+1} = \mathbf{f}^{h,n+1} - \mathbf{a}^{h,n} \tag{9}$$

$$\nabla \cdot \mathbf{u}^{h,n+1} = 0 \tag{10}$$

where \mathcal{P}^h is a projection operator from the particles to the FE mesh, $\mathbf{u}^{h,n+1} = \mathbf{u}^h(\mathbf{x}, t^{n+1})$, $\widehat{\mathbf{u}}^{h,n+1} = \widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1})$, $p^{h,n+1} = p^h(\mathbf{x}, t^{n+1})$, $\mathbf{a}^{h,n} = \mathbf{a}^h(\mathbf{x}, t^n)$ and $\mathbf{f}^{h,n+1} = \mathbf{f}^h(\mathbf{x}, t^{n+1})$. We refer to earlier papers [16,17,19] on the

² In multi-fluid flows, the interface between multiple fluids is reconstructed [1] on the FE mesh using the advected particle identities. Appropriate enrichments are determined [1] for the pressure FE shape functions about the interface.

Table 1 Algorithm to implement the SL–PFEM using the X-IVAS scheme

1.	Compute $\mathbf{u}^h(\mathbf{x}, t^n), p^h(\mathbf{x}, t^n) \rightarrow \mathbf{a}^h(\mathbf{x}, t^n)$ using the weak form of Eq. (3).
2.	Integrate $\mathbf{X}^h(\lambda, t^n) \rightarrow \mathbf{X}^h(\lambda, t^{n+1})$ using Eq. (6).
3.	Integrate $\mathbf{U}^h(\lambda, t^n) \rightarrow \mathbf{U}^h(\lambda, t^{n+1})$ using Eq. (7).
4.	Transfer $\mathbf{U}^h(\lambda, t^{n+1}) \rightarrow \widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1})$ using Eqs. (11) and (12).
5.	Integrate $\widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1}), \mathbf{a}^h(\mathbf{x}, t^n) \rightarrow \mathbf{u}^h(\mathbf{x}, t^{n+1}), p^h(\mathbf{x}, t^{n+1})$ using Eqs. (9) and (10).
6.	Update $\mathbf{U}^h(\lambda, t^{n+1}), \widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1}), \mathbf{u}^h(\mathbf{x}, t^{n+1}) \rightarrow \mathbf{U}^h(\lambda, t^{n+1})$ using Eq. (13).
7.	End of time step. Execute step 1 for next time step.

SL–PFEM for several alternate time integration strategies for the Stokes system to be solved on the background mesh.

Usually projection is an implicit operation which incurs a notable cost of inverting a global matrix. To avoid this cost an explicit kernel smoothing operator is used instead of the projection operator in Eq. (8). Thus, $\widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1})$ is approximated as the kernel smoother of the set of discrete data $\mathbf{U}^h(\lambda, t^{n+1})$ stored with the particles.

$$\widehat{\mathbf{u}}^h(\mathbf{x}, t^{n+1}) = \sum_{\lambda} \mathcal{K}^h(\mathbf{x}, \lambda) \mathbf{U}^h(\lambda, t^{n+1}) \quad (11)$$

$$\mathcal{K}^h(\mathbf{x}, \lambda) = \sum_a N^a(\mathbf{x}) \frac{W^a(\mathbf{X}^h(\lambda, t^{n+1}))}{\sum_{\bar{\lambda}} W^a(\mathbf{X}^h(\bar{\lambda}, t^{n+1}))} \quad (12)$$

where $N^a(\mathbf{x})$ and $W^a(\mathbf{x})$ are the standard FEM shape function and an appropriate weighting function associated with the FEM node with index a . λ and $\bar{\lambda}$ denote particle identity indices. We have used $W^a(\mathbf{x}) = N^a(\mathbf{x})$ in the simulations.

Finally the particle velocities are updated by the increment $\mathbf{u}^{h,n+1} - \widehat{\mathbf{u}}^{h,n+1}$ evaluated at the particle positions.

$$\begin{aligned} \mathbf{U}^h(\lambda, t^{n+1}) &= \mathbf{U}^h(\lambda, t^{n+1}) \\ &+ \mathbf{u}^h(\mathbf{X}^h(\lambda, t^{n+1}), t^{n+1}) - \widehat{\mathbf{u}}^h(\mathbf{X}^h(\lambda, t^{n+1}), t^{n+1}) \end{aligned} \quad (13)$$

Remark The kernel smoothing technique introduces numerical diffusion when data is transferred from the particles to the mesh. However, it must be noted that the particle data is not completely overwritten. Note that in Eq. (13) only the increment $\mathbf{u}^{h,n+1} - \widehat{\mathbf{u}}^{h,n+1}$ is transferred back to the particle which serves as a hedge against numerical diffusion.

4 Seakeeping simulation of the WAM-V using the SL–PFEM

Experimental testing of the WAM-V prototype revealed an operability problem—at high speeds the hull generates an intense water spray (see Fig. 1). Beyond a critical speed the spray leads to reduced visibility and the operator (or the



Fig. 1 Spray generated by the WAM-V hull at 21 knots [25]. Courtesy of Prof. Mehdi Ahmadian, VirginiaTech, USA

reconnaissance equipment) will get drenched. Both represent operational and also safety hazards, especially in cold climates where there is a heightened risk of marine ice formation.

Computer simulations are highly valuable to evaluate a wide range of ideas prior to construction and prevent the cost of purchasing components. The results of such simulations assist engineers in the decision making process towards better hull design that control and mitigate the spray formation.

The inflatable hull of the WAM-V is designed such that it displaces just a few feet below the waterline. It is reasonable to assume that the waterline never reaches the sub-structure connecting the hulls during the motion of the WAM-V. Therefore we just simulate the response of the free surface of the water due to the action of the WAM-V hull. A qualitative understanding of the physical conditions leading to spray generation can be obtained by simulating the action of just one hull.

The domain is a 3D box with straight walls (see Fig. 2) with dimensions: 9, 3 and 2 m along the x , y and z axes, respectively. The coordinates $(0, 0, -1)$ and $(9, 3, 1)$ represent two diagonally opposite corners of the domain. The face with coordinates $(0, 0, -1)$, $(9, 0, -1)$, $(9, 0, 1)$ and $(0, 0, 1)$ represents the plane of symmetry of the WAM-V. The geometry of the hull considered in the simulations corresponds to those of the 12 feet WAM-V. The 3D space of the fluid domain is obtained by subtracting the volume occupied by the catamaran-type hull of the WAM-V from the volume of

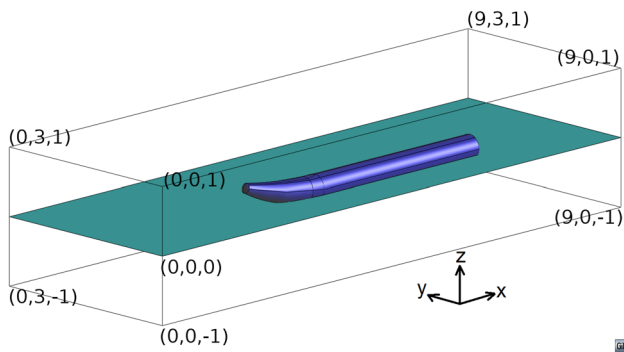


Fig. 2 The domain dimensions, location of the hull and the waterline

the containing 3D box. The draft of the hull, i.e. the displacement below the steady waterline is taken as 4 inches.

The problem domain is discretized by a mesh of 490 778 nodes and 3 080 211 four-node tetrahedral elements. On an average twenty particles (material points that transport intrinsic properties of the fluid) per element were used in the simulations.

A pre-defined rigid body oscillatory motion is imposed as a transient boundary condition to model the pitch of the WAM-V hull. The pitch axis is parallel to the y -axis and passes through the point $(6, 0, 0)$. The angle of rotation (in radians) about the pitch axis is taken as

$$\theta = -0.06 \sin^2(5t) \tag{14}$$

The background mesh is deformed every time step such that its internal boundary always conforms to that of the WAM-V hull. The mesh deformation scheme used is based on a Laplacian solver which is commonly used in the implementation of the Arbitrary Lagrangian–Eulerian formulations. This enables us to impose the no-slip velocity boundary conditions at the internal boundary in a straight-forward manner. The computer code used in this work was developed in the C++ programming environment KRATOS® [5]. KRATOS® is an open-source multi-physics software framework which provide a collection of tools useful to perform tasks that are common to all FEM codes.

Water wave motion is generated by imposing the solution of a deep-water second-order Stokes wave in a narrow strip of water at the inlet and on the walls of the outlet.

$$a = 0.05 \text{ m}, \quad k = \frac{4\pi}{3}, \quad g = 9.8 \text{ m/s}^2, \quad \omega = \sqrt{gk} \tag{15}$$

$$u_x(x, z, t) = a\omega \exp(kz) \cos[k(x - Ut) - \omega t] + U \tag{16}$$

$$u_z(x, z, t) = a\omega \exp(kz) \sin[k(x - Ut) - \omega t] \tag{17}$$

where a is the amplitude of the wave, k is the angular wavenumber, g is the acceleration due to gravity and ω is the angular frequency. Further U is the velocity with which the WAM-V moves relative to water and u_x, u_z represent the

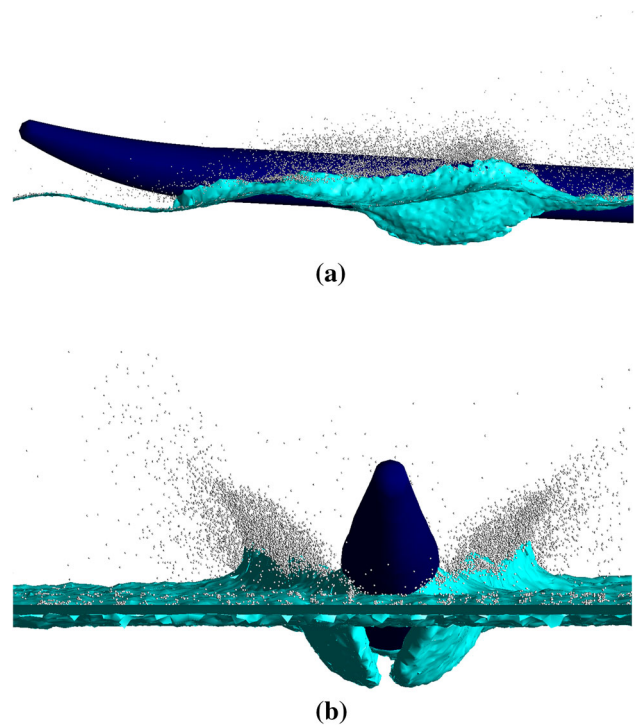


Fig. 3 Detail of the spray generated by the 12 ft WAM-V hull at 25 knots and time = 3.57 s. The isocontour corresponding to the air–water interface and the water particles in the air-elements are displayed. **a** XZ plane view **b** YZ plane view

spatial inlet velocity components of the water in an inertial reference frame that moves with the WAM-V. The narrow strip at the inlet where the wave velocity is imposed has a width of 0.2 m. This periodic velocity condition causes a disturbance which is propagated in the rest of the domain and whose motion is governed by the Navier–Stokes equations.

Three cases were studied which correspond to three different speeds of the WAM-V, viz. $U = 15$ knots, $U = 20$ knots and $U = 25$ knots, respectively. The total physical time of simulation was chosen to be 4 s using 800 time steps of 0.005 s each. All the three cases took nearly 24 h each to perform the computations using a workstation with an Intel® Core™ i7 – 3820 CPU and 32 GB RAM. The computations were performed in parallel using four (available) cores and a single thread per core. Due to the very large mesh size (3 million elements) and particle collection (60 million particles), the memory requirements of these simulations are high; nearly 17 GB of RAM.

Figure 3 illustrates the details of the spray generated by the 12 ft WAM-V hull moving at 25 knots. Eight snapshots are shown in Fig. 4 which correspond to a time interval (3.47 s–3.82 s) near the end of the simulation. In this sequence of snapshots we can see the initiation of the spray generation, its gradual development and a skewed separation from the axis of the WAM-V hull (cf. Fig. 1). Near the end of this snapshot

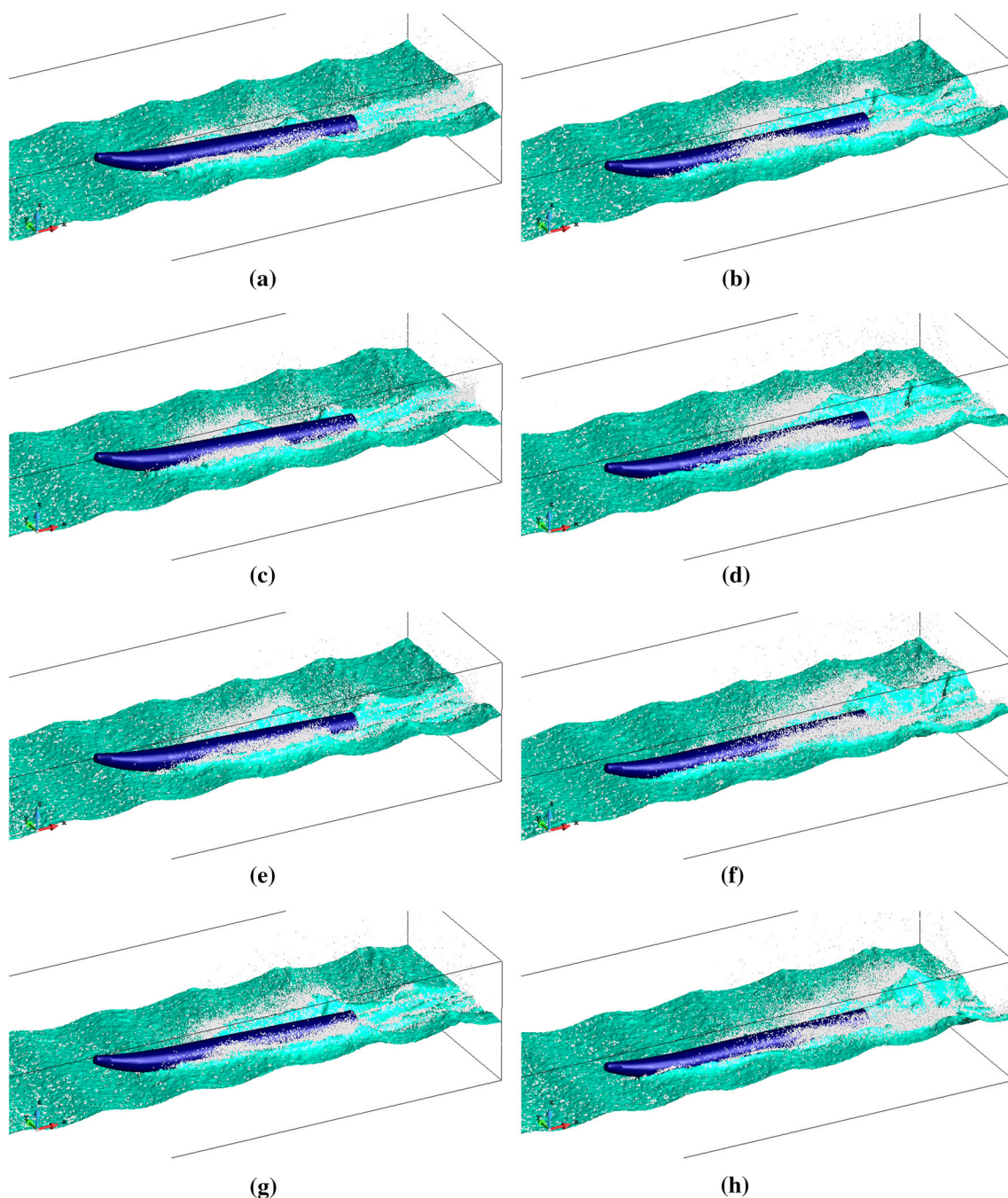


Fig. 4 Simulation of the spray generated by the 12 ft WAM-V hull moving at 25 knots using the SL–PFEM. The isocontour corresponding to the air–water interface and the water particles in the air-elements are

displayed. **a** $t = 3.47$ s, **b** $t = 3.67$ s, **c** $t = 3.52$ s, **d** $t = 3.72$ s, **e** $t = 3.57$ s, **f** $t = 3.77$ s, **g** $t = 3.62$ s, **h** $t = 3.82$ s

sequence we can see the process start over again. The spray generation (with respect to the location) at regular intervals is caused due to the presence of waves and the imposed pitch of the hull. The differences in the results obtained using $U = 15$ knots and $U = 20$ knots (not shown here) are in the height and intensity of the spray, which as expected increased gradually with the velocity.

We see that the SL–PFEM is suitable for the seakeeping simulation of the WAM-V under spray generating conditions. Conventional CFD software tools are not suitable to reproduce spray generating conditions for this challenging problem. This is an example where particle-based incompressible flow simulation tools (e.g. SL–PFEM) stand out and deliver superior results.

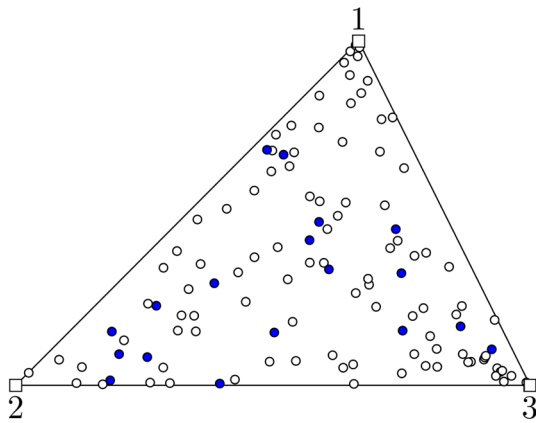


Fig. 5 Particles that compose the spray in an air-element. The water particles are shown as blue circles; the air particles are seen as white circles

In these simulations the water–air interface evolves in a very complicated manner. At any particular instant of time, all elements of the background mesh are labelled as water-element, air-element or interface-element. Each particle has an identity which is either air (label: +1) or water (label: −1). Within each element and at every time-step the particles transfer their identities to the element nodes. These contributions are weighted based on their location within the element and are assembled at the nodes. After the assembly, each node has an identity between −1 and +1. Following this line, a continuous piecewise linear approximation of the otherwise discrete identity data (on the particles) is obtained on the background mesh. The water–air interface is defined as the piecewise linear (planar) surface where the continuous piecewise linear approximate identity takes a value 0. The boundaries of the interface are determined at the resolution of the background mesh.

In due course, situations may arise where there are regions within an air-element that are filled with water particles. There will also be interface-elements wherein we will find water particles on the air-side of the interface. These particles compose the spray generated in the simulations (see Fig. 5). Should the pockets of water particles be large enough then it creates a situation where there are one or more water-elements surrounded by air-elements. These islands of water-elements are seen as water splash in the simulations which represent violent separation and/or merger of the interface. Naturally, such representation of water spray and its intensity depends on the number of particles chosen in the simulation. Nevertheless, the number of particles that compose the spray is not a representation of the mass of water in the spray. Recall that particles represent material points that carry with them only the intrinsic properties of the flow. So a smaller number of particles just mean that the spray representation is sampled at a coarser level of detail.

Other possible physical conditions which may generate spray are not modelled in the simulations. For instance, the viscous action of air motion may separate water particles from the interface or decompose existing water splash into water spray. Additionally, this phenomenon can happen at multiple scales wherein entities which can be classified as water spray at a coarse scale may be classified as water splash at a fine scale (which in turn can be decomposed into spray). This cascade will continue until surface tension forces comes to prominence and protect the integrity of the water droplets. Reproducing such physical conditions is out of the scope of the current implementation of the SL–PFEM.

Remark For a wide class of engineering applications the X-IVAS scheme usually admits very large time steps in the Lagrangian advection stage, e.g. 10–15 times the one given by the CFL condition. However in the simulations presented here we had to use small time steps (of the same order as the CFL limit) to control the accuracy of the solution. The reason for this limitation is discussed in the next section.

5 Trajectory approximation for Stokes waves

Consider a second-order Stokes wave in deep sea and position the 2D coordinate system at the height of the mean sea level. The solution of the free surface $\eta(x, t)$ is

$$\eta(x, t) = a \cos(kx - \omega t) + \frac{1}{2} a^2 k \cos[2(kx - \omega t)] \quad (18)$$

Substituting $U = 0$ in the Eqs. (16) and (17) we obtain the solution of the velocity components.

Consider the initial time step in the X-IVAS scheme and let the initial velocity and pressure be given by substituting $t = 0$ in the exact solution. Figure 6 illustrates the streamlines of the initial velocity field originating at three distinct positions on the surface of the Stokes wave. Two of these initial positions are chosen as the crest and trough of the wave and the third point is chosen midway. In the X-IVAS scheme the computed trajectories are identical to the streamlines. The exact trajectories (pathlines) of particles initially located at these three points are also shown.

For the Stokes wave we see that the streamlines are not a good approximation to the pathlines. Recall that in the X-IVAS scheme we first compute tangent curves (streamlines) of the velocity vector field and choose them as the trajectories of the particles.

$$\mathbf{X}^h(\lambda, t) = \mathbf{X}^h(\lambda, t^n) + \int_{t^n}^t \mathbf{u}^h(\mathbf{X}^h(\lambda, \xi), t^n) d\xi \quad (19)$$

Then we increment particle velocities by an amount equal to the line integral of the acceleration vector field along the streamlines.

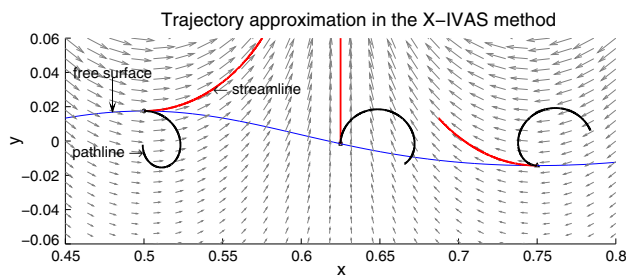


Fig. 6 Trajectory approximation in the X-IVAS scheme. The velocity field in the initial configuration is shown in the background as a vector plot

$$\mathbf{U}^h(\lambda, t) = \mathbf{U}^h(\lambda, t^n) + \int_{t^n}^t \mathbf{a}^h(\mathbf{X}^h(\lambda, \tau), t^n) d\tau \quad (20)$$

Therefore the updates in the particle position and its velocity within a time-step are decoupled in the X-IVAS scheme. When the velocity field has a significant temporal variation within a time step, the trajectory computed using Eq. (19) will have a significant error. In the Stokes wave propagation problem, transporting particles along the streamlines (see Fig. 6) with large time steps will lead to significant errors in the configuration of the fluid domain. Now consider an alternative scheme for the Lagrangian advection: update the fluid particle position and velocity within a time-step $t^n < t < t^{n+1}$ using solutions to the following explicit second-order system of equations.

$$\frac{d^2}{dt^2} \mathbf{X}^h(\lambda, t) = \mathbf{a}^h(\mathbf{X}^h(\lambda, t), t^n) = \mathbf{C}^n \mathbf{X}^h(\lambda, t) + \mathbf{d}^n \quad (21)$$

Unlike in the X-IVAS scheme the solution of the above equation admits correction to the trajectories caused due to the updates in the particle velocities within a time-step.

$$\mathbf{U}^h(\lambda, t) = \mathbf{U}^h(\lambda, t^n) + \int_{t^n}^t \mathbf{a}^h(\mathbf{X}^h(\lambda, \xi), t^n) d\xi \quad (22)$$

$$\begin{aligned} \mathbf{X}^h(\lambda, t) &= \mathbf{X}^h(\lambda, t^n) + (t - t^n) \mathbf{U}^h(\lambda, t^n) \\ &+ \int_{t^n}^t \int_{t^n}^{\tau} \mathbf{a}^h(\mathbf{X}^h(\lambda, \xi), t^n) d\xi d\tau \end{aligned} \quad (23)$$

Figure 7 illustrates the trajectories driven by the initial acceleration field and originating at the considered three initial positions. Clearly the approximation in Fig. 7 stands out as a higher fidelity model to the pathlines. Numerically stable formulas to compute the closed-form analytical solution of Eq. (21) will be done during the implementation of the Marie Skłodowska-Curie action—FastFlowSim³.

³ http://cordis.europa.eu/project/rcn/200435_en.html

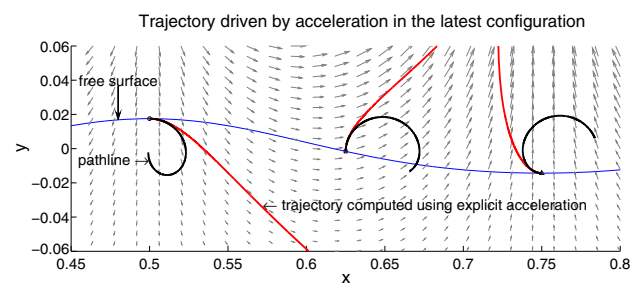


Fig. 7 Trajectory driven by the acceleration field (shown in the background as a vector plot) in the initial configuration

6 Conclusions

In the seakeeping simulation of the WAM-V, we have seen that the semi-Lagrangian particle finite element method is a suitable CFD tool for the analysis of incompressible flows subjected to challenging physical conditions, e.g. violent interface motions, spray generating conditions etc. Due to the Lagrangian treatment of the advective processes and the Lagrangian data storage strategy in the SL-PFEM, the interfaces are accurately tracked. Further, the computational task associated to advective transport is mutually exclusive and hence scalable on parallel computers.

However, seakeeping simulations often involve water wave motion. Examining a second-order Stokes wave propagating on the surface of a deep sea, we have seen that the streamlines are not a good approximation to the pathlines. In such situations the use of large time steps in the X-IVAS scheme will result in significant errors in the fluid configuration. The particle trajectories driven by the acceleration in the initial configuration of the second-order Stokes wave are shown to be a better approximation to the pathlines. For seakeeping simulations with the SL-PFEM, this latter time-integration scheme seems to be a promising alternative to mitigate the small time-step limitation of the X-IVAS scheme.

Acknowledgements Permission to use the image shown in Fig. 1 has been granted by Prof. Mehdi Ahmadian, VirginiaTech, USA. This image has appeared earlier in Andrew William Peterson's Ph.D. thesis [25, Fig. 3.12, p. 55].

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Funding This study was partially supported by the WAM-V project funded under the Navy Grant N62909-12-1-7101 issued by Office of Naval Research Global, the SAFECON project (ref. 267521, FP7-IDEAS-ERC) and the FORECAST project (ref. 664910, H2020-ERC-2014-PoC) of the European Research Council (European Commission). The United States Government has a royalty-free license throughout the world in all copyrightable material contained herein.

References

1. Becker P, Idelsohn SR, Oñate E (2014) A unified monolithic approach for multi-fluid flows and fluid–structure interaction using the particle finite element method with fixed mesh. *Comput Mech* 55:1091. doi:[10.1007/s00466-014-1107-0](https://doi.org/10.1007/s00466-014-1107-0)
2. Celledoni E, Kometa BK, Verdier O (2015) High order semi-Lagrangian methods for the incompressible Navier–Stokes equations. *J Sci Comput* 66:91. doi:[10.1007/s10915-015-0015-6](https://doi.org/10.1007/s10915-015-0015-6)
3. Courant R, Friedrichs K, Lewy H (1967) On the partial difference equations of mathematical physics. *IBM J Res Dev* 11:215. doi:[10.1147/rd.112.0215](https://doi.org/10.1147/rd.112.0215)
4. Courant R, Isaacson E, Rees M (1952) On the solution of nonlinear hyperbolic differential equations by finite differences. *Commun Pure Appl Math* 5(3):243–255. doi:[10.1002/cpa.3160050303](https://doi.org/10.1002/cpa.3160050303)
5. Dadvand P, Rossi R, Oñate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Arch Comput Methods Eng* 17(3):253–297. doi:[10.1007/s11831-010-9045-2](https://doi.org/10.1007/s11831-010-9045-2)
6. Dupont TF, Liu Y (2003) Back and forth error compensation and correction methods for removing errors induced by uneven gradients of the level set function. *J Comput Phys* 190(1):311–324. doi:[10.1016/S0021-9991\(03\)00276-6](https://doi.org/10.1016/S0021-9991(03)00276-6)
7. Dupont TF, Liu Y (2007) Back and forth error compensation and correction methods for semi-lagrangian schemes with application to level set interface computations. *Math Comput* 76(258):647–669. doi:[10.1090/S0025-5718-06-01898-9](https://doi.org/10.1090/S0025-5718-06-01898-9)
8. Evans MW, Harlow FH (1957) The particle-in-cell method for hydrodynamic calculations. Technical report, Los Alamos National Laboratory, Los Alamos. <http://www.fas.org/sgp/othergov/doe/lanl/dtic/ADA384618.html>
9. Gelet RM, Nguyen G, Rognon P (2015) Modelling interaction of incompressible fluids and deformable particles with the Material Point Method. In: The 6th International conference on computational methods (ICCM2015)
10. Gimenez JM, Nigro NM, Idelsohn SR (2014) Evaluating the performance of the particle finite element method in parallel architectures. *Comput Part Mech* 1(1):103–116. doi:[10.1007/s40571-014-0009-4](https://doi.org/10.1007/s40571-014-0009-4)
11. Harlow FH (1957) Hydrodynamic problems involving large fluid distortions. *J ACM* 4(2):137–142. doi:[10.1145/320868.320871](https://doi.org/10.1145/320868.320871)
12. Higham NJ (2002) Accuracy and stability of numerical algorithms. *Soc Ind Appl Math*. doi:[10.1137/1.9780898718027](https://doi.org/10.1137/1.9780898718027)
13. Idelsohn SR, Marti J, Becker P, Oñate E (2014) Analysis of multifluid flows with large time steps using the particle finite element method. *Int J Numer Methods Fluids* 75(9):621–644. doi:[10.1002/fld.3908](https://doi.org/10.1002/fld.3908)
14. Idelsohn SR, Mier-Torrecilla M, Nigro N, Oñate E (2010) On the analysis of heterogeneous fluids with jumps in the viscosity using a discontinuous pressure field. *Comput Mech* 46(1):115–124. doi:[10.1007/s00466-009-0448-6](https://doi.org/10.1007/s00466-009-0448-6)
15. Idelsohn SR, de Mier-Torrecilla M, Oñate E (2009) Multi-fluid flows with the particle finite element method. *Comput Methods Appl Mech Eng* 198(33–36):2750–2767. doi:[10.1016/j.cma.2009.04.002](https://doi.org/10.1016/j.cma.2009.04.002)
16. Idelsohn SR, Nigro N, Limache A, Oñate E (2012) Large time-step explicit integration method for solving problems with dominant convection. *Comput Methods Appl Mech Eng* 217–220:168–185. doi:[10.1016/j.cma.2011.12.008](https://doi.org/10.1016/j.cma.2011.12.008)
17. Idelsohn SR, Nigro NM, Gimenez JM, Rossi R, Marti JM (2013) A fast and accurate method to solve the incompressible Navier–Stokes equations. *Eng Comput* 30(2):197–222. doi:[10.1108/02644401311304854](https://doi.org/10.1108/02644401311304854)
18. Idelsohn SR, Oñate E, Del Pin F (2004) The particle finite element method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *Int J Numer Methods Eng* 61(7):964–989. doi:[10.1002/nme.1096](https://doi.org/10.1002/nme.1096)
19. Idelsohn SR, Oñate E, Nigro N, Becker P, Gimenez J (2015) Lagrangian versus Eulerian integration errors. *Comput Methods Appl Mech Eng* 293:191–206. doi:[10.1016/j.cma.2015.04.003](https://doi.org/10.1016/j.cma.2015.04.003)
20. MacCormack RW (2003) The effect of viscosity in hypervelocity impact cratering. *J Spacecr rockets* 40(5):757–763. doi:[10.2514/2.6901](https://doi.org/10.2514/2.6901)
21. Marti J, Ryzhakov P, Idelsohn SR, Oñate E (2012) Combined Eulerian-PFEM approach for analysis of polymers in fire situations. *Int J Numer Methods Eng* 92:782. doi:[10.1002/nme.4357](https://doi.org/10.1002/nme.4357)
22. Min C, Gibou F (2006) A second order accurate projection method for the incompressible Navier–Stokes equations on non-graded adaptive grids. *J Comput Phys* 219(2):912–929. doi:[10.1016/j.jcp.2006.07.019](https://doi.org/10.1016/j.jcp.2006.07.019)
23. Nadukandi P (2015) Numerically stable formulas for a particle-based explicit exponential integrator. *Comput Mech* 55(5):903–920. doi:[10.1007/s00466-015-1142-5](https://doi.org/10.1007/s00466-015-1142-5)
24. Nielson GM, Jung IH (1999) Tools for computing tangent curves for linearly varying vector fields over tetrahedral domains. *IEEE Trans Vis Comput Graph* 5(4):360–372. doi:[10.1109/2945.817352](https://doi.org/10.1109/2945.817352)
25. Peterson AW (2014) Simulation and Testing of wave-adaptive modular vessels. Ph.D. thesis, Virginia Polytechnic Institute and State University. <http://hdl.handle.net/10919/54555>
26. Robert A (1981) A stable numerical integration scheme for the primitive meteorological equations. *Atmosphere-Ocean* 19(1):35–46. doi:[10.1080/07055900.1981.9649098](https://doi.org/10.1080/07055900.1981.9649098)
27. Ryzhakov P (2016) A modified fractional step method for fluid–structure interaction problems. *Revista Internacional de Métodos Numéricos para Cálculo y Diseño en Ingeniería*. doi:[10.1016/j.rimni.2015.09.002](https://doi.org/10.1016/j.rimni.2015.09.002)
28. Ryzhakov P, Rossi R, Oñate E (2012) An algorithm for the simulation of thermally coupled low speed flow problems. *Int J Numer Methods Fluids* 70(1):1–19. doi:[10.1002/fld.2674](https://doi.org/10.1002/fld.2674)
29. Sawyer JS (1963) A semi-Lagrangian method of solving the vorticity advection equation. *Tellus* 15(4):336–342. doi:[10.1111/j.2153-3490.1963.tb01396.x](https://doi.org/10.1111/j.2153-3490.1963.tb01396.x)
30. Selle A, Fedkiw R, Kim B, Liu Y, Rossignac J (2008) An unconditionally stable MacCormack method. *J Sci Comput* 35(2–3):350–371. doi:[10.1007/s10915-007-9166-4](https://doi.org/10.1007/s10915-007-9166-4)
31. Sulsky D, Zhou SJ, Schreyer HL (1995) Application of a particle-in-cell method to solid mechanics. *Comput Phys Commun* 87(1–2):236–252. doi:[10.1016/0010-4655\(94\)00170-7](https://doi.org/10.1016/0010-4655(94)00170-7)
32. WAM-V: the wave adaptive modular vessel. <http://www.wam-v.com>
33. Zhang DZ, Zou Q, VanderHeyden WB, Ma X (2008) Material point method applied to multiphase flows. *J Comput Phys* 227(6):3159–3173. doi:[10.1016/j.jcp.2007.11.021](https://doi.org/10.1016/j.jcp.2007.11.021)