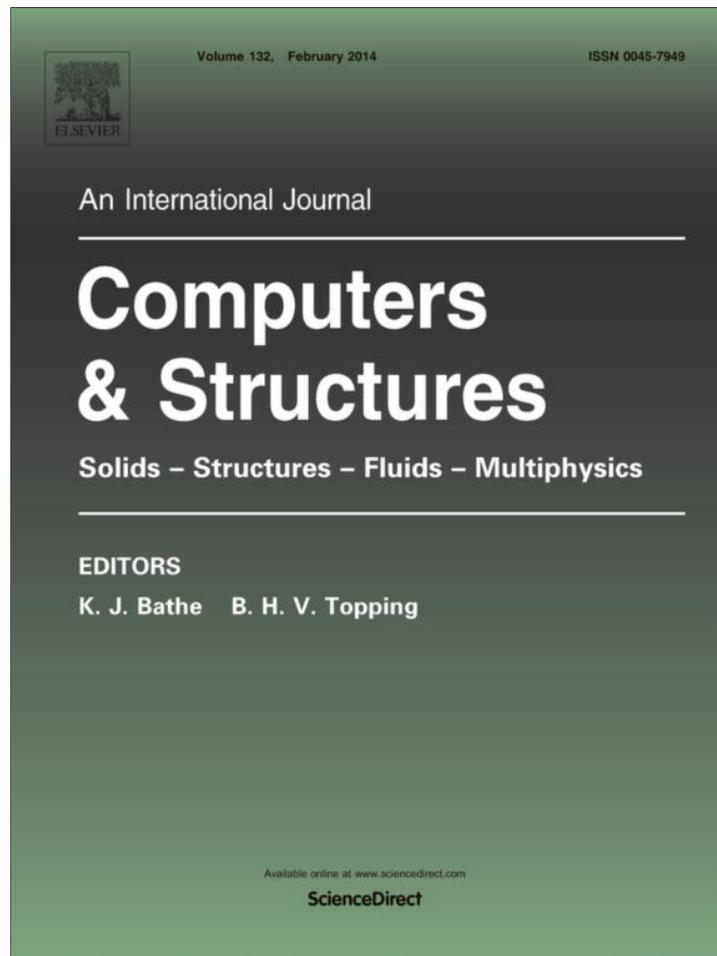


Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

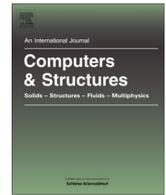
In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/authorsrights>



Contents lists available at ScienceDirect

Computers and Structures

journal homepage: [www.elsevier.com/locate/compstruc](http://www.elsevier.com/locate/compstruc)

# Modeling fluid–structure interaction by the particle finite element method in OpenSees



Minjie Zhu, Michael H. Scott\*

School of Civil and Construction Engineering, Oregon State University, Corvallis, OR 97331, USA

## ARTICLE INFO

### Article history:

Received 4 July 2013

Accepted 4 November 2013

Available online 5 December 2013

### Keywords:

Fluid–structure interaction

Finite element analysis

Wave loading

OpenSees

## ABSTRACT

The OpenSees finite element software framework is extended for simulating fluid–structure interaction (FSI) by the particle finite element method (PFEM). At high levels of the framework, new classes handle meshing and interface detection of the fluid and structure domains and implement the fractional step method in order to solve the governing equations of linear momentum and mass conservation. At lower levels of the framework, new finite element and pressure constraint classes assemble fluid contributions to the global system of equations. Verification and validation examples are presented along with a demonstrative example of wave loading on a coastal structure modeled using geometrically nonlinear frame elements with material nonlinear fiber sections. The extension of OpenSees for FSI allows analysts to simulate the complex phenomena of wave loading on structural models as well as the response of these models to sequential natural hazards such as earthquake induced tsunamis.

© 2013 Elsevier Ltd. All rights reserved.

## 1. Introduction

Recent natural disasters such as the 2004 Indian Ocean and 2011 Tohoku earthquakes and tsunamis have increased the importance of simulating fluid–structure interaction (FSI) among structural engineering researchers and practitioners. Documented structural failures and subsequent analyses that are able to replicate the failure mechanisms experienced during these events are expected to have far-reaching implications on international design codes [1]. As a result, the design of new coastal infrastructure requires accurate simulation models in order to predict structural response to extreme wave loading, which is often precipitated by ground motion. Also of importance is predicting the response of existing coastal infrastructure to loads not anticipated in the original design, e.g., uplift forces on bridge decks and impact forces of water borne debris.

Various methods of simulating fluid–structure interaction have been developed for simulating wave impact on structures [2–6]. A common assumption in wave impact analysis is that the structure is a rigid object. Although it simplifies the analysis, this assumption can lead to over-estimates of the force that a structure can resist. To account for the deformation of structures impacted by wave loads, a finite element model of the structure is employed [7].

For the fluid domain, finite element approaches to simulating incompressible Newtonian fluids use either the Eulerian, Lagrangian or Arbitrary Lagrangian–Eulerian (ALE) formulations [8–12].

The Eulerian formulation tracks fluid motion through fixed locations in space while the Lagrangian formulation tracks the motion of individual fluid particles. For FSI simulations, a Lagrangian formulation is advantageous because, compared to an Eulerian formulation, it is easy to track the fluid free surface and it is free of the convective terms that pose numerical difficulties with Eulerian formulations. The disadvantage of the fully Lagrangian formulation is the element stretching which can be overcome by updating the mesh in every time step. This remeshing carries a high computational expense but can be mitigated by quality mesh generators. The ALE formulation attempts to combine the best features of both Lagrangian and Eulerian descriptions; however, it is challenging to implement and may not be suitable for breaking wave problems that are common in fluid–structure interaction [13].

An important advantage of the Lagrangian fluid formulation for FSI is that it conforms to Lagrangian formulations of structural mechanics. The particle finite element method (PFEM) has been shown to be an efficient approach to simulating fluid–structure interaction [14,15]. The efficiency of the PFEM arises from its tracking of the fluid surface in Lagrangian form using a computational procedure analogous to that of traditional solid finite element formulations. As a result, it is possible to use the PFEM to overcome the complexity of coupling separate fluid and structure analysis methods via staggered schemes [14], or monolithic approaches by assuming quasi-incompressible fluid [16,17]. The PFEM also is less sensitive to the exact location of the fluid–structure boundary, which can be a drawback to domain decomposition methods. Many applications of the PFEM have focused on highly flexible, elastic structures, such as gages and membranes [16] and flaps

\* Corresponding author.

E-mail address: [michael.scott@oregonstate.edu](mailto:michael.scott@oregonstate.edu) (M.H. Scott).

and valves [15]. While these components are important for certain application spaces, they are not representative of the bridges and buildings that comprise coastal infrastructure.

The open source finite element software framework OpenSees [18] has been developed to advance research in the simulation of structural response to earthquake hazards. A software framework defines the abstract classes from which developers can create concrete classes that implement specific functionality. The finite element formulations, constitutive models, and solution algorithms of the OpenSees framework have been designed in such a way that applications not foreseen in the initial framework development can be incorporated with a series of incremental improvements or additions. For example, the OpenSees framework has been extended to live load rating of bridge girders [19] and to simulate structural response to fire attack [20].

The objective of this paper is to show how the OpenSees framework is extended to accommodate the PFEM for FSI applications using a monolithic approach for fully incompressible fluid. First, the governing equations of the PFEM are presented along with their discrete approximations in space and time. Implementation details are presented for the new classes added to OpenSees for handling additional pressure and pressure gradient unknowns at the element level. Details at the structural level where the solution for the pressure unknowns is obtained alongside nodal displacements are also presented. Examples of fluid only and FSI are shown in order to verify, validate, and demonstrate the PFEM implementation within OpenSees.

## 2. Governing equations of the PFEM

A PFEM analysis satisfies conservation of linear momentum and conservation of mass for all points in the fluid and structural domains. Constitutive laws relate the displacements at points in the fluid and structural domains to pressure or stress.

### 2.1. Conservation of linear momentum

Conservation of linear momentum is enforced in the volume,  $V$ , of both the fluid and structural domains

$$\rho \ddot{u}_i = \frac{\partial \sigma_{ij}}{\partial x_j} + \rho b_i \quad (1)$$

where  $\ddot{u}_i$  is the acceleration vector,  $\sigma_{ij}$  is the Cauchy stress tensor,  $x_j$  is the current position vector,  $b_i$  is the body force vector, and  $\rho$  is the density. Boundary conditions for both domains are enforced for prescribed tractions on the surface,  $\Gamma_t$ ,

$$\sigma_{ij} n_j = t_i \quad (2)$$

where  $t_i$  is the surface traction and  $n_j$  is the unit normal vector to the boundary surface. Boundary and initial conditions are imposed on displacements

$$u_i = u_i^p, \quad u_i = u_i^0 \quad (3)$$

where  $u_i^p$  is the fixed displacement on the boundary and  $u_i^0$  is the initial displacement. Initial conditions on velocity,  $\dot{u}_i^0$ , may also be prescribed. As the simulation proceeds, the relationship between position and displacement is

$$x_i = x_i^0 + u_i \quad (4)$$

where  $x_i^0$  is the initial position.

### 2.2. Conservation of mass

Mass conservation, or the continuity equation, must be satisfied in the fluid domain. Assuming incompressible fluid flow, continuity requires the divergence of velocity to be zero

$$\frac{\partial \dot{u}_i}{\partial x_i} = 0 \quad (5)$$

Conservation of mass is satisfied in the structural domain by construction.

### 2.3. Constitutive equations

In the structural domain, the constitutive equations are written as a general stress–strain relationship

$$\sigma_{ij} \equiv \sigma_{ij}(\varepsilon_{kl}) \quad (6)$$

where  $\varepsilon_{kl}$  is the strain tensor computed from derivatives of the displacement field

$$\varepsilon_{kl} = \frac{1}{2} \left( \frac{\partial u_k}{\partial x_l} + \frac{\partial u_l}{\partial x_k} \right) \quad (7)$$

The general stress–strain relationship allows for material nonlinear structural response to external loading.

For an incompressible fluid, the Newtonian constitutive equations are expressed as

$$\sigma_{ij} = S_{ij} - p \delta_{ij} \quad (8)$$

where  $\delta_{ij}$  is the Kronecker delta and  $p$  is the pressure. The deviatoric stress tensor,  $S_{ij}$ , is defined for linear fluid response as

$$S_{ij} = 2\mu \dot{\varepsilon}_{ij} \quad (9)$$

where  $\mu$  is the fluid viscosity and  $\dot{\varepsilon}_{ij}$  is the strain rate tensor, which is the time derivative of Eq. (7).

## 3. Finite element discretization

Discretization of the continuous governing equations of linear momentum and mass conservation leads to a system of algebraic equations for the displacements and pressures at discrete locations known as particles. An additional governing equation is introduced in order to stabilize the conservation of mass equation. Then, the response of both the fluid and structural domains is determined from a monolithic system of equations.

### 3.1. Stabilization of mass equation

For fluid elements that do not satisfy the LBB condition [21,22], spurious pressure modes can be eliminated by augmenting the mass conservation equation (Eq. (5)) with stabilizing terms [23–25]

$$\frac{\partial \dot{u}_i}{\partial x_i} - \sum_{i=1}^{n_d} \tau \frac{\partial}{\partial x_i} \left( \frac{\partial p}{\partial x_i} + \pi_i \right) = 0 \quad (10)$$

where  $n_d$  is the number of spatial dimensions,  $\pi_i$  is the pressure gradient projection, and  $\tau$  is the stabilization parameter [26]

$$\tau = \left( \frac{\rho}{\Delta t} + \frac{8\mu}{3l^2} \right)^{-1} \quad (11)$$

The variables  $\Delta t$  and  $l$  are the simulation time step and characteristic element length, respectively.

The pressure gradient projection ensures that the stabilizing terms in Eq. (10) vanish for exact solution of the continuity equation. This gives an additional equation that governs the response of fluid particles

$$\frac{\partial p}{\partial x_i} + \pi_i = 0 \quad (12)$$

The unknown pressure gradient,  $\pi_i$ , must be found at the global level along with the particle displacements and pressures, as described next.

### 3.2. Shape functions

Choosing the particle displacement,  $u_i$ , pressure,  $p$ , and pressure gradient,  $\pi_i$ , as primary unknowns and applying the standard Galerkin weighted residual method to the momentum equation (1), stabilized mass equation (10), pressure gradient projection equation (12), and boundary conditions gives

$$\int_V \delta u_i \left( \rho \ddot{u}_i - \frac{\partial \sigma_{ij}}{\partial x_j} - \rho b_i \right) dV - \int_{\Gamma_t} \delta u_i t_i d\Gamma_t = 0 \quad (13)$$

$$\int_V q \frac{\partial \dot{u}_i}{\partial x_i} dV + \int_V \sum_{i=1}^{n_d} \tau \frac{\partial q}{\partial x_i} \left( \frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \quad (14)$$

$$\int_V \delta \pi_i \tau \left( \frac{\partial p}{\partial x_i} + \pi_i \right) dV = 0 \quad (15)$$

where  $\delta u_i$ ,  $q$ , and  $\delta \pi_i$  are weighting functions that satisfy the essential boundary conditions. The stabilization parameter,  $\tau$ , is introduced in Eq. (15) for symmetry with Eq. (14). Boundary terms from the integration by parts in Eq. (14) are neglected.

Using standard finite element techniques [7], the displacement, pressure, and pressure gradient are approximated over each element using equal order linear interpolation

$$u_i = \sum_{j=1}^n N_j u_i^j, \quad p = \sum_{j=1}^n N_j p^j, \quad \pi_i = \sum_{j=1}^n N_j \pi_i^j \quad (16)$$

where  $n$  is the number of element nodes and  $N_j$  are the shape functions. For planar analysis with triangle elements, the shape functions are equal to the area coordinate of node  $j$ . In three dimensions, volume coordinates are used for the shape functions of tetrahedral elements.

### 3.3. System of algebraic equations

When assembled over all elements in the fluid domain, the discretized equations for the fluid particle response are expressed in matrix–vector form as

$$\mathbf{M}_f \ddot{\mathbf{u}}_f + \mathbf{K}_f \dot{\mathbf{u}}_f - \mathbf{G}_f \mathbf{p} = \mathbf{F}_f \quad (17)$$

$$\mathbf{G}_f^T \dot{\mathbf{u}}_f + \mathbf{L} \mathbf{p} + \mathbf{Q} \pi = \mathbf{0} \quad (18)$$

$$\mathbf{Q}^T \mathbf{p} + \hat{\mathbf{M}} \pi = \mathbf{0} \quad (19)$$

where  $\mathbf{u}_f$ ,  $\mathbf{p}$ , and  $\pi$  are vectors that collect the displacement, pressure, and pressure gradient of all fluid particles and  $\mathbf{F}_f$  is the vector of external forces. The objects  $\mathbf{M}_f$  and  $\mathbf{K}_f$  are the mass and stiffness matrices, respectively, of the fluid;  $\mathbf{G}_f$  is the gradient operator;  $\mathbf{L}$  is the Laplacian operator; and  $\mathbf{Q}$  and  $\hat{\mathbf{M}}$  are stabilization matrices. Further information on the discretized fluid equations is found in [24].

The discretized system of equations for dynamic response of the structural domain is

$$\mathbf{M}_s \ddot{\mathbf{u}}_s + \mathbf{C}_s \dot{\mathbf{u}}_s + \mathbf{F}_s^{int}(\mathbf{u}_s) = \mathbf{F}_s \quad (20)$$

where  $\mathbf{u}_s$  is the displacement vector of the structural particles (or nodes) and  $\mathbf{F}_s$  is the external load vector. The static resisting force vector,  $\mathbf{F}_s^{int}$ , is a nonlinear function of the nodal displacements. Like the mass and damping matrices,  $\mathbf{M}_s$  and  $\mathbf{C}_s$ , respectively, the resisting force vector is assembled from element contributions.

Particles connected to elements from both the fluid and structural domains are identified as interface particles whose contributions appear in the system of equations for both domains. From the structural domain, equations governing the interface response are extracted from Eq. (20) and assigned additional  $i$  and  $s$  subscripts

$$\mathbf{M}_{ss} \ddot{\mathbf{u}}_s + \mathbf{M}_{si} \ddot{\mathbf{u}}_i + \mathbf{C}_{ss} \dot{\mathbf{u}}_s + \mathbf{C}_{si} \dot{\mathbf{u}}_i + \mathbf{F}_s^{int}(\mathbf{u}_s, \mathbf{u}_i) = \mathbf{F}_s \quad (21)$$

$$\mathbf{M}_{is} \ddot{\mathbf{u}}_s + \mathbf{M}_{ii}^s \ddot{\mathbf{u}}_i + \mathbf{C}_{is} \dot{\mathbf{u}}_s + \mathbf{C}_{ii} \dot{\mathbf{u}}_i + \mathbf{F}_i^{int}(\mathbf{u}_s, \mathbf{u}_i) = \mathbf{F}_i^s \quad (22)$$

where  $\mathbf{u}_i$  is the displacement vector of the interface nodes. Similarly, the interface equations are extracted from Eqs. (17) and (18) for the fluid domain and given additional  $i$  and  $f$  subscripts

$$\mathbf{M}_{ff} \ddot{\mathbf{u}}_f + \mathbf{K}_{ff} \dot{\mathbf{u}}_f - \mathbf{G}_f \mathbf{p} = \mathbf{F}_f \quad (23)$$

$$\mathbf{M}_{ii}^f \dot{\mathbf{u}}_i + \mathbf{K}_{ii} \dot{\mathbf{u}}_i - \mathbf{G}_i \mathbf{p} = \mathbf{F}_i^f \quad (24)$$

$$\mathbf{G}_f^T \dot{\mathbf{u}}_f + \mathbf{G}_i^T \dot{\mathbf{u}}_i + \mathbf{L} \mathbf{p} + \mathbf{Q} \pi = \mathbf{0} \quad (25)$$

Eqs. (22) and (24) are combined in order to solve for the particle response on the fluid–structure interface.

## 4. Solution of discretized equations

At each simulation time step,  $t_{n+1}$ , the discretized momentum, pressure, and pressure gradient equations must be solved considering the change in state from the previous time step,  $t_n$ . The governing equations are posed in residual form then solved via the fractional step method.

### 4.1. Nonlinear solution algorithm

To utilize a wide range of root finding algorithms, Eqs. (21) and (23) for the structure and fluid response, respectively; Eqs. (22) and (24) for the interface response; and Eqs. (19) and (25) for the pressure and pressure gradient response are combined to give the following system of residual equations

$$\begin{aligned} \mathbf{r}_s &= \mathbf{F}_s - \mathbf{M}_{ss} \ddot{\mathbf{u}}_s - \mathbf{M}_{si} \ddot{\mathbf{u}}_i - \mathbf{C}_{ss} \dot{\mathbf{u}}_s - \mathbf{C}_{si} \dot{\mathbf{u}}_i - \mathbf{F}_s^{int}(\mathbf{u}_s, \mathbf{u}_i) \\ \mathbf{r}_i &= \mathbf{F}_i^s + \mathbf{F}_i^f - (\mathbf{M}_{ii}^f + \mathbf{M}_{ii}^s) \ddot{\mathbf{u}}_i - (\mathbf{C}_{ii} + \mathbf{K}_{ii}) \dot{\mathbf{u}}_i - \mathbf{C}_{is} \dot{\mathbf{u}}_s - \mathbf{M}_{is} \ddot{\mathbf{u}}_s \\ &\quad - \mathbf{F}_i^{int}(\mathbf{u}_s, \mathbf{u}_i) + \mathbf{G}_i \mathbf{p} \\ \mathbf{r}_f &= \mathbf{F}_f - \mathbf{M}_{ff} \ddot{\mathbf{u}}_f - \mathbf{K}_{ff} \dot{\mathbf{u}}_f + \mathbf{G}_f \mathbf{p} \\ \mathbf{r}_p &= -\mathbf{G}_f^T \dot{\mathbf{u}}_f - \mathbf{G}_i^T \dot{\mathbf{u}}_i - \mathbf{L} \mathbf{p} - \mathbf{Q} \pi \\ \mathbf{r}_\pi &= -\mathbf{Q}^T \mathbf{p} - \hat{\mathbf{M}} \pi \end{aligned} \quad (26)$$

For simultaneous solution of the preceding equations, all unknowns are collected in a single vector,  $\mathbf{v}$ , along with the vector,  $\mathbf{r}$ , of residual equations

$$\mathbf{v} = \begin{bmatrix} \dot{\mathbf{u}}_s \\ \dot{\mathbf{u}}_i \\ \dot{\mathbf{u}}_f \\ \mathbf{p} \\ \pi \end{bmatrix}, \quad \mathbf{r} = \begin{bmatrix} \mathbf{r}_s \\ \mathbf{r}_i \\ \mathbf{r}_f \\ \mathbf{r}_p \\ \mathbf{r}_\pi \end{bmatrix} \quad (27)$$

With an initial guess for the unknowns at the start of the current time step, typically the converged state at the previous time step,  $\mathbf{v}_{n+1}^0 = \mathbf{v}_n$ , the incremental update within a simulation time step is

$$\mathbf{v}_{n+1}^{j+1} = \mathbf{v}_{n+1}^j + \Delta \mathbf{v}^{j+1} \quad (28)$$

The update is computed according to a Newton algorithm

$$\Delta \mathbf{v}^{j+1} = (\mathbf{K}_r^j)^{-1} \mathbf{r}^j \quad (29)$$

where  $\mathbf{K}_r^j = -\partial \mathbf{r} / \partial \mathbf{v}$  is the Jacobian of the residual evaluated at the current value of the unknowns,  $\mathbf{v}^j$

$$\mathbf{K}_T = \begin{bmatrix} \tilde{\mathbf{K}}_{Tss} & \tilde{\mathbf{K}}_{Tsi} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \tilde{\mathbf{K}}_{Tis} & \tilde{\mathbf{K}}_{Tii} & \mathbf{0} & -\mathbf{G}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{K}}_{Tff} & -\mathbf{G}_f & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_i^T & \mathbf{G}_f^T & \mathbf{L} & \mathbf{Q} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{Q}^T & \tilde{\mathbf{M}} \end{bmatrix} \quad (30)$$

Matrices with an over-tilde, e.g.,  $\tilde{\mathbf{K}}_{Tss}$ , are the algorithmic matrices that depend on the simulation time step and the chosen time integration method.

Backward Euler time integration is employed here using the particle velocities as primary unknowns. The displacement and acceleration are expressed in terms of the velocity at the current time step and the displacement and velocity at the previous time step

$$\begin{aligned} \mathbf{u}_{n+1} &= \mathbf{u}_n + \Delta t \dot{\mathbf{u}}_{n+1} \\ \ddot{\mathbf{u}}_{n+1} &= \frac{\dot{\mathbf{u}}_{n+1} - \dot{\mathbf{u}}_n}{\Delta t} \end{aligned} \quad (31)$$

Inserting these approximations into the system of algebraic equations leads to the algorithmic matrix for the structural contribution

$$\tilde{\mathbf{K}}_{Tss} = \frac{1}{\Delta t} \mathbf{M}_{ss} + \mathbf{C}_{ss} + \Delta t \mathbf{K}_{Tss} \quad (32)$$

where  $\mathbf{K}_{Tss} = \partial \mathbf{F}_s^{int}(\mathbf{u}_s, \mathbf{u}_i) / \partial \mathbf{u}_s$  is the tangent stiffness matrix of the structure. Analogous definitions for  $\tilde{\mathbf{K}}_{Tsi}$  and  $\tilde{\mathbf{K}}_{Tis}$  are straightforward to obtain.

The algorithmic matrix for the fluid contribution is

$$\tilde{\mathbf{K}}_{Tff} = \frac{1}{\Delta t} \mathbf{M}_{ff} + \mathbf{K}_{ff} \quad (33)$$

where  $\mathbf{K}_{ff}$  can be ignored for small  $\Delta t$  and low fluid viscosity. The contribution of the interface nodes to the Jacobian is

$$\tilde{\mathbf{K}}_{Tii} = \frac{1}{\Delta t} (\mathbf{M}_{ii}^f + \mathbf{M}_{ii}^s) + (\mathbf{C}_{ii} + \mathbf{K}_{ii}) + \Delta t \mathbf{K}_{Tii} \quad (34)$$

where  $\mathbf{K}_{Tii} = \partial \mathbf{F}_i^{int}(\mathbf{u}_s, \mathbf{u}_i) / \partial \mathbf{u}_i$  is the contribution of the tangent stiffness matrix of the structure to the interface particles. Similar to the fluid contribution in Eq. (33),  $\mathbf{K}_{ii}$  can be ignored for small  $\Delta t$  and low viscosity.

#### 4.2. Fractional step method

The monolithic matrix in Eq. (30) is ill-conditioned due to coupling of the velocity and pressure fields, making it difficult to obtain a stable numerical solution for the incremental velocities, pressures, and pressure gradients. To solve for these quantities efficiently, the fractional step method (FSM) is utilized [27,28,15]. The FSM segregates the unknown pressures and velocities into smaller systems of equations that generally are not ill-conditioned. The FSM can be summarized in three steps:

1. Compute predictor velocities by ignoring pressure contributions arising from  $\mathbf{G}_i$  and  $\mathbf{G}_f$  in the first three rows of Eq. (30).
2. Solve for the pressures from the fourth row of Eq. (30) using added mass and stiffness from the predicted velocities.
3. Correct the velocities and update the pressure gradients using the pressures found in step 2.

Implementation of the FSM requires important changes at high levels of the OpenSees framework, as described in the following section.

## 5. PFEM implementation in OpenSees

The software design of OpenSees favors object composition over class inheritance as the mechanism that enables flexibility and extensibility of the framework. At the highest level of the OpenSees framework, the *Domain* class contains components (nodes, elements, loads, constraints, etc.) that are created and added to the domain by a *ModelBuilder* object through an input script. The state of each domain component is computed by an **Analysis** object, which is composed of an equation solver, solution algorithm, time integrator, constraint handler, and element and nodal assembly objects. Complete details of the design of OpenSees for nonlinear finite element analysis are given in [29]. Only the details of the PFEM implementation in OpenSees are presented herein.

Since it was primarily designed to solve structural dynamics problems, there are two major challenges to the implementation of the PFEM in OpenSees. The first challenge arises from the PFEM's necessity to update the finite element mesh at every time step due to large domain changes and changes in the fluid–structure interface. While this re-meshing and interface detection can be handled at the script level, it is necessary for efficiency to implement these modules within the OpenSees core. The second challenge involves solving the linear system of equations via the FSM, which requires multiple equation solutions using the submatrices in Eq. (30). Although OpenSees contains a flexible set of equation solvers, there is an implicit assumption that only one equation solution takes place during each iteration within a time step.

### 5.1. Meshing and interface detection

To create particles and elements of the fluid domain, the *PFEMMesher* class is introduced to the OpenSees framework, as shown in Fig. 1. An instance of the *ModelBuilder* class calls the *PFEMMesher*, which for two-dimensional analysis, discretizes a planar straight line graph [30] into particles at the start of the analysis. In every time step, a *PFEMMesher* object generates a new mesh of *PFEMElement* objects for the fluid domain and interface using Delaunay triangulation and the alpha shape method based on the current particle positions. The alpha shape method eliminates unnecessary elements and detects changing boundaries of the fluid domain [31]. Meshing for three-dimensional analysis is more complex, but does not add complexity to the software implementation.

Similar to other elements in OpenSees, each instance of a *PFEMElement* is responsible for computing and returning to a calling object its mass, stiffness, and damping matrices and resisting force vector based on the state of its connected nodes and any internal history variables. When it is added to the domain, a *PFEMElement* creates a *Pressure\_Constraint* object for each of its connected particles (Fig. 1). The *Pressure\_Constraint* object identifies each particle as belonging to the fluid or structural domains, the interface between these domains, or as separate from any domain. To this end, two groups of element tags (one for fluid and one for structure) are stored in a *Pressure\_Constraint* object and set via the *connect/disconnect* pair of methods. Methods such as *isStructure* shown in Fig. 2 return the particle type based on its connections and allow elements to determine their state accordingly. Another important task of a *Pressure\_Constraint* object is to store a *Node* object internally in order to keep track of the pressure and pressure gradient unknowns. This is in addition to external *Node* objects that keep track of the displacement, velocity, and acceleration of all fluid and structural particles. The tag of the internal pressure node can be returned through the *getPressureNode* method so that fluid elements can obtain the particle state and return their contributions to the governing equations.

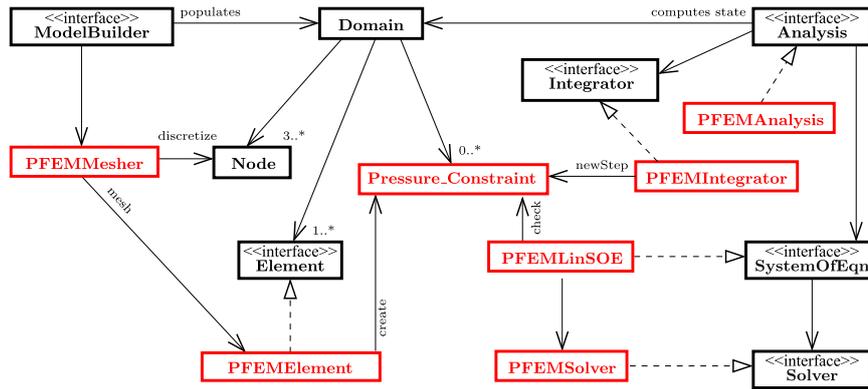


Fig. 1. Class diagram of the PFEM implementation in OpenSees (new classes for the PFEM are shown in red). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

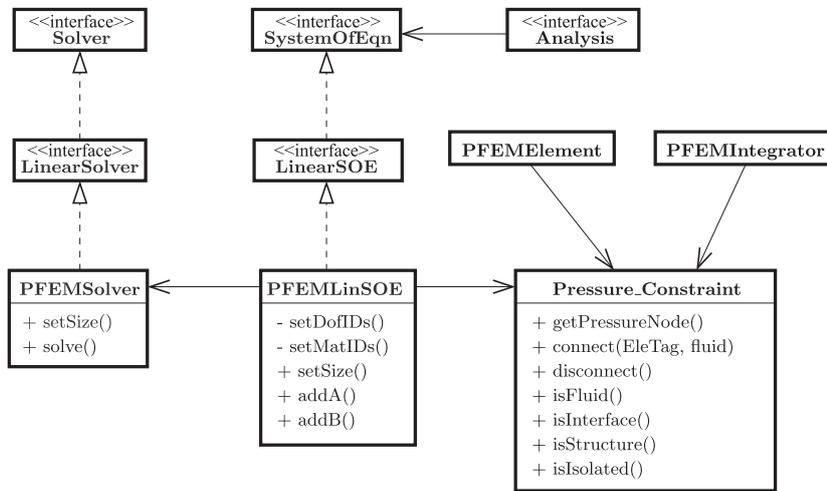


Fig. 2. Class diagram of the PFEM solver in OpenSees.

With a fluid mesh in place, the *PFEMIntegrator* class is able to implement the implicit Euler time integration method using particle velocity as the primary unknown along with the pressure and pressure gradient. At the start of each time step, the integrator calls *Pressure\_Constraint* objects to update the state of each isolated particle and to assemble the governing equations for particles that are connected to a mesh of fluid elements. The *PFEMAnalysis* class sets the maximum and minimum time steps for the simulation and may reduce the time step if convergence is not achieved.

### 5.2. Fractional step method

In addition to identifying the domain to which particles belong, the *Pressure\_Constraint* class serves as a bridge between the analysis and model classes of OpenSees in order to link the finite element model to the predictor–corrector approach of the FSM. On the analysis side, new implementations of the *LinearSOE* and *LinearSolver* interfaces shown in Fig. 2 are required in order to carry out the FSM and partition the matrices in Eq. (30) based on the model information from *Pressure\_Constraint* objects. To this end, the *setDofIDs* method of the *PFEMLinSOE* class, which inherits the *LinearSOE* interface, obtains the node types from the *Pressure\_Constraint* objects and sets the matrix partitions and assigns equation numbers. The *setMatIDs* method is then called in order to initialize the partitioned matrices and residual vector of Eq. (30) for assembly via implementations of the *addA* and *addB* methods. Using the partitioned matrices stored in the *PFEMLinSOE* object, the *PFEM-*

*Solver*, which implements the *LinearSolver* interface, carries out the FSM and returns the solution for incremental velocities, pressures, and pressure gradients.

## 6. Examples

Examples are presented herein to verify and validate the PFEM implementation in OpenSees and to demonstrate its application to fluid–structure interaction. Fluid sloshing in a container is used for verification, then collapse of a water column is shown for validation, followed by time history analysis of a coastal structure subjected to wave loading. This final example demonstrates how structural models comprised of frame finite elements and fiber sections (typically employed for earthquake loading) can be subjected to wave loading via the PFEM implementation in OpenSees.

### 6.1. Fluid sloshing

Sloshing, or the free oscillation in a container, of an incompressible fluid is a common verification problem for PFEM implementations [32,33]. The PFEM model of the fluid, shown in Fig. 3(a), has the initial free surface profile shown in Fig. 3(b) corresponding to the first anti-symmetric mode of oscillation

$$\eta(x, 0) = a \sin\left(\frac{\pi x}{b}\right) \quad (35)$$

where *a* is the amplitude of oscillation and *b* is the container width.

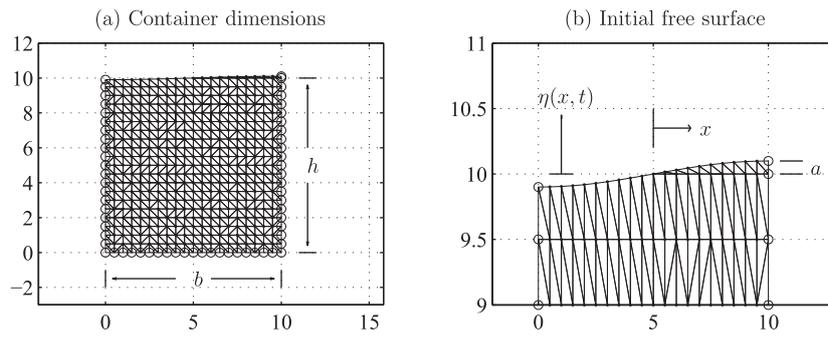


Fig. 3. Initial mesh for fluid sloshing problem.

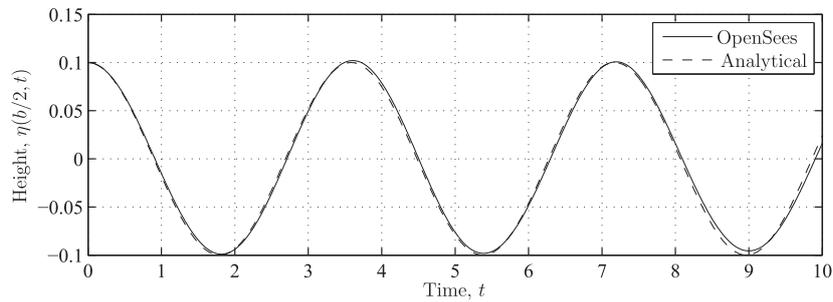


Fig. 4. Time history of fluid sloshing problem.

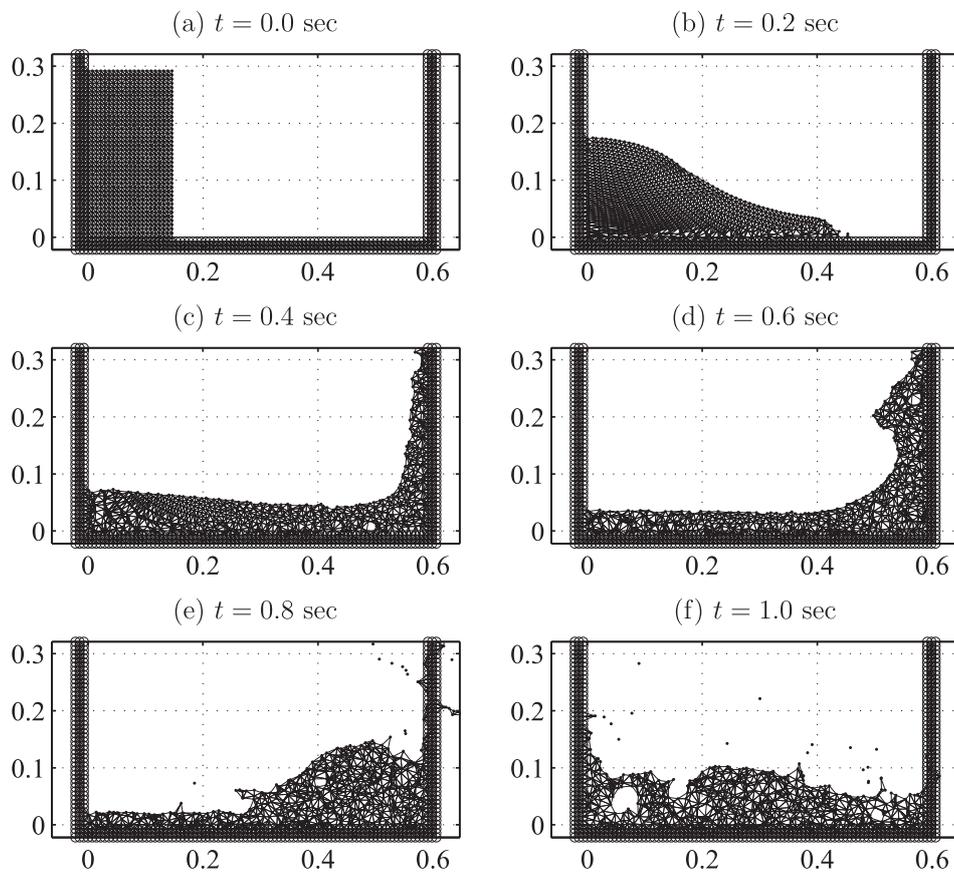


Fig. 5. Initial mesh and simulated response of water column collapse.

When the amplitude is small compared to the container height,  $h$ , the analytical solution for the motion of the free surface is

$$\eta(x, t) = a \sin\left(\frac{\pi x}{b}\right) \cos(\sigma t) \quad (36)$$

The frequency of oscillation,  $\sigma$ , is calculated from the dispersion relationship

$$\sigma^2 = \frac{\pi g}{b} \tanh\left(\frac{\pi h}{b}\right) \quad (37)$$

where  $g$  is the gravitational constant.

For numerical simulation via the PFEM, the mesh shown in Fig. 3(a) with container dimensions,  $b = h = 10$  m, and amplitude,  $a = 0.1$  m, has 453 nodes and 819 elements. The simulation time step is  $\Delta t = 0.001$  s. The sloshing time history of the free surface at  $x = b/2$  is shown in Fig. 4. The simulation results agree with the closed form solution save for the numerical approximation of the implicit Euler time integration.

### 6.2. Water column collapse

Where the sloshing time history provided verification using a closed-form solution of steady state response, the collapse of a water column is a standard example for validation of Lagrangian

formulations of fluid flow undergoing highly nonlinear mesh distortions [32]. The initial configuration of the water column is shown in Fig. 5(a) with 1392 nodes and 2429 elements. Using a simulation time step of  $\Delta t = 0.001$  s, the evolution of the free surface shown in Fig. 5(b)–(f) agrees qualitatively with previously published results [32,34].

Quantitative comparisons of the OpenSees simulation with experimental results of the water column collapse [35] are obtained by tracking: (1) the horizontal motion of the water's leading edge on the bottom surface of the tank; and (2) the vertical motion of the top of the water column. The coordinate time histories are plotted against experimental results in Fig. 6. The simulated horizontal motion of the leading edge in Fig. 6(a) shows higher velocity than the experimental results. This result is consistent with prior simulations and can be attributed to friction between the fluid and bottom wall [34]. Fig. 6(b) shows the simulated vertical motion for the top of the water column, which is dominated by gravity, agrees well with the experimental results.

### 6.3. Coastal structure

The final example demonstrates fluid–structure interaction via the simulation of wave loading on a coastal structure. As previously described, the PFEM implementation allows arbitrary struc-

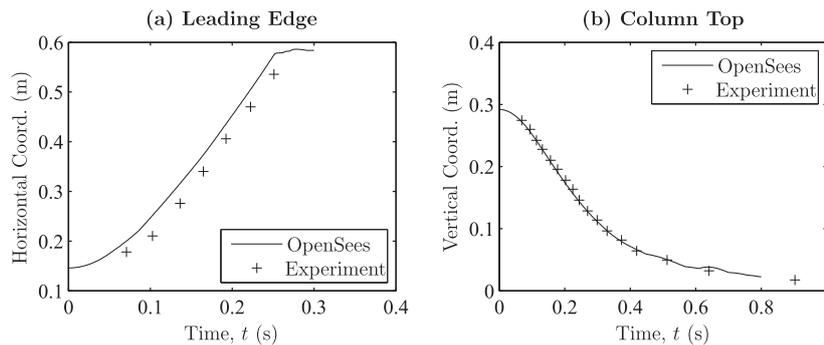


Fig. 6. Comparison of OpenSees simulation with experimental results of water column collapse.

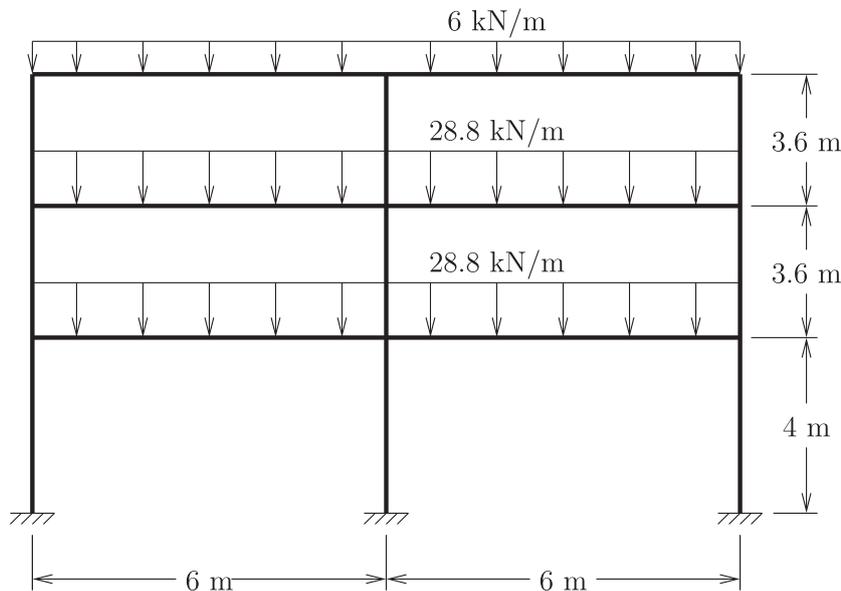


Fig. 7. Geometry and floor loads of reinforced concrete frame example.

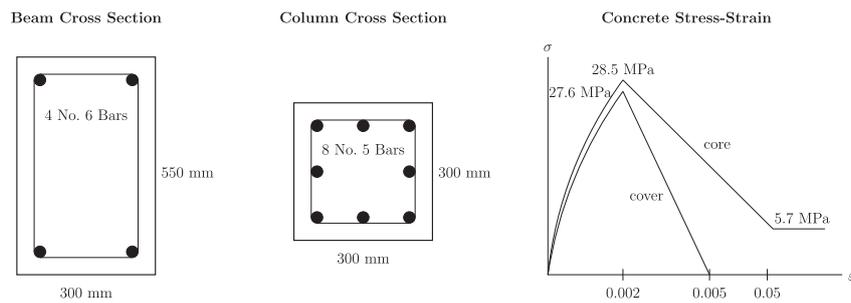


Fig. 8. Beam and column cross-sections of reinforced concrete frame.

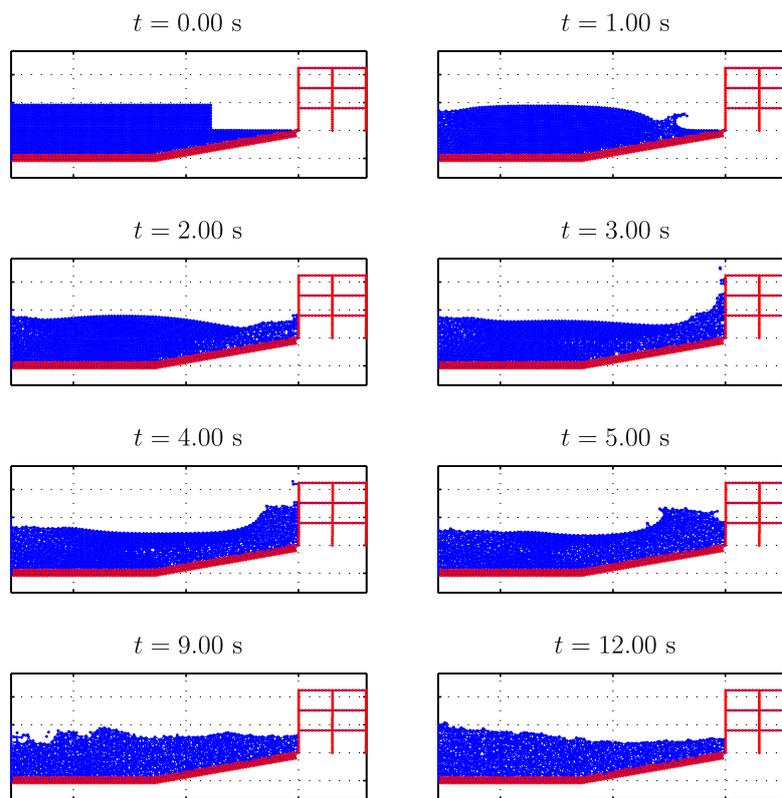


Fig. 9. Time history of wave runup on coastal structure.

tural finite elements to be used so that the existing modules of OpenSees can be exploited for detailed analyses. Although the simulation does not capture essential three-dimensional FSI effects, it demonstrates the capabilities.

The structural model is of the interior frame of a reinforced concrete building analyzed by Madurapperuma and Wijeyewickrema [36] for impact of water borne debris (see Fig. 7). Dead load on all members consists of self-weight and beam live loads were computed assuming uniform 4.8 kPa on floor slabs and 1.0 kPa on the roof with tributary width of 6 m. Combined dead and live load were used in assigning lumped masses to the frame nodes. The frame members are discretized into ten displacement-based frame finite elements, each with constant axial strain and linear curvature approximations (*dispBeamColumn* in OpenSees). Although frame finite elements typically use a relatively coarse mesh, the resulting element lengths are comparable to the characteristic size of the fluid mesh so that a complete fluid–structure interface is developed during the simulation. The corotational transformation [37] captures geometric nonlinear response of the frame.

Fiber discretized cross-sections at the element Gauss points capture material nonlinear response of the frame members. The cross-section dimensions, reinforcing details, and concrete properties of the frame are shown in Fig. 8. Light transverse reinforcement provides residual concrete compressive strength in the core regions of the members. Zero tensile strength is assumed for the concrete (*Concrete01* in OpenSees) and the longitudinal reinforcing steel is assumed bilinear with elastic modulus 200 GPa, yield strength 420 MPa, and 1% kinematic strain hardening (*Steel01*).

Wave loading emanates from a tsunami bore of height 4 m traveling at an initial velocity of 2 m/s with an out-of-plane thickness 0.3 m equal to the out-of-plane thickness of the frame. Tributary loading from exterior walls is not taken into account during the simulation. The wave runup, breaking, and surge are shown in Fig. 9 at various snapshots of the response time history. Floor displacement response time histories are shown in Fig. 10(a). The peak displacements occur at about 4 s, well after the initial wave impact, which occurred prior to 2 s. As the wave loading draws down after 5 s, the time history shows damped vibration about

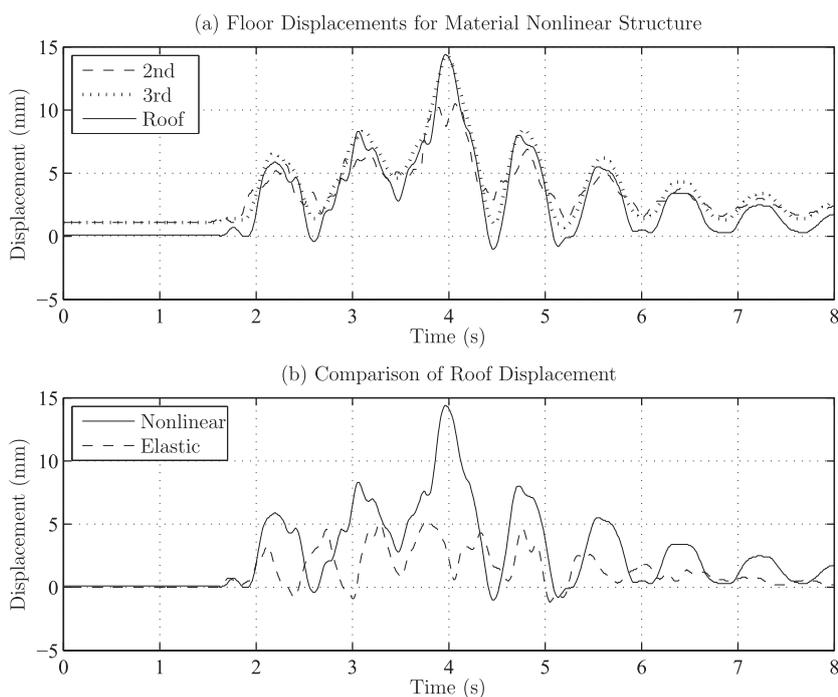


Fig. 10. Time history of floor displacements for material nonlinear and elastic structural models.

small permanent floor displacements, indicating material nonlinearity in the structure. A comparison of the roof displacement time history with an analysis where the concrete and reinforcing steel respond linear-elastically with their initial stiffness is shown in Fig. 10(b). The peak roof displacement of the elastic model is about three times less than that of the material nonlinear model and, as expected, there is no permanent displacement. Further calculations are required in order to find the internal forces of the structure; however, the time history of floor displacements shown in Fig. 10 reveal the capabilities of modeling wave impact on structures using the nonlinear frame elements that are prevalent in OpenSees.

## 7. Conclusions

The extension of the OpenSees framework to incorporate fluid–structure interaction by the particle finite element method is an important step in structural engineering research for multiple natural hazards involving earthquakes and tsunamis. The implementation was verified for water sloshing and validated for the collapse of a water column. Wave loading was applied to a representative coastal structure from which response quantities of interest to structural engineers were obtained. Modeling of debris impact and scour are the focus of future fluid–structure capabilities in OpenSees, as is the simulation of cascading hazards involving fire after earthquake and tsunamis. In addition, extension of the implementation to three dimensions is underway in order to account for important wave load effects that cannot be captured in a two-dimensional model and to take advantage of the high performance computing capabilities of OpenSees.

## Acknowledgments

This material is based on work supported by the National Science Foundation under Grant No. 0847055. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views

of the National Science Foundation. The authors would like to thank Dr. Frank McKenna at the University of California, Berkeley for assistance with the PFEM implementation in OpenSees.

## References

- [1] Chock G, Carden L, Robertson I, Olsen M, Yu G. Tohoku tsunami-induced building failure analysis with implications for U.S. tsunami and seismic design codes. *Earthquake Spectra* 2013;29(S1):S99–S126.
- [2] van Loon R, Anderson P, van de Vosse F, Sherwin S. Comparison of various fluid–structure interaction methods for deformable bodies. *Comput Struct* 2007;85:833–43.
- [3] Barreiro A, Crespo A, Domínguez J, Gómez-Gesteira M. Smoothed particle hydrodynamics for coastal engineering problems. *Comput Struct* 2013;120:96–106.
- [4] Tedesco JW, McDougal WG, Bloomquist D, Wise LA. Response of concrete armor units to wave-induced hydrodynamic loads. *Comput Struct* 2003;81:983–94.
- [5] Broderick L, Leonard J. Nonlinear water–wave structure interaction. *Comput Struct* 1992;44:837–42.
- [6] Schuëller G. Development of a stochastic wave force function on ocean-structures. *Comput Struct* 1971;1:639–49.
- [7] Bathe KJ. *Finite element procedures*. Prentice-Hall; 1996.
- [8] Girault V, Raviart P. *Finite element methods for Navier–Stokes equations*. Springer-Verlag; 1986.
- [9] Gunzburger M. *Finite element methods for viscous incompressible flows*. Academic Press; 1989.
- [10] Baiges J, Codina R. The fixed-mesh ale approach applied to solid mechanics and fluid–structure interaction problems. *Int J Numer Methods Eng* 2010;81:1529–57.
- [11] Radovitzky R, Ortiz M. Lagrangian finite element analysis of newtonian fluid flows. *Int J Numer Methods Eng* 1998;43:607–19.
- [12] Tezduyar T, Mittal S, Ray S, Shih R. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements. *Comput Methods Appl Mech Eng* 1992;95:221–42.
- [13] Nithiarasu P. An arbitrary Lagrangian Eulerian (ALE) formulation for free surface flows using the characteristic-based split (CBS) scheme. *Int J Numer Methods Fluids* 2006;50:1119–20.
- [14] Oñate E, Idelsohn S, Celigueta M, Rossi R, Martí J, Carbonell J, et al. *Advances in the particle finite element method (PFEM) for solving coupled problems in engineering*. Springer; 2011 [chapter: Particle-based methods].
- [15] Idelsohn S, Pin FD, Rossi R, Oñate E. Fluid–structure interaction problems with strong added-mass effect. *Int J Numer Methods Eng* 2009;80:1261–94.
- [16] Ryzhakov P, Rossi R, Idelsohn S, Oñate E. A monolithic Lagrangian approach for fluid–structure interaction problems. *Comput Mech* 2010;46:883–99.
- [17] Idelsohn S, Martí J, Limache A, Oñate E. Unified Lagrangian formulation for elastic solids and incompressible fluids: application to fluid–structure

- interaction problems via the PFEM. *Comput Methods Appl Mech Eng* 2008;197:1762–76.
- [18] McKenna F, Fenves GL, Scott MH. Open system for earthquake engineering simulation. Berkeley, CA: University of California; 2000. Available from: <<http://opensees.berkeley.edu>>.
- [19] Scott MH, Kidarsa A, Higgins C. Development of bridge rating applications using OpenSees and Tcl. *J Comput Civil Eng* 2008;22(4):264–71.
- [20] Jiang J, Usmani A. Modeling of steel frame structures in fire using OpenSees. *Comput Struct* 2013;118:90–9.
- [21] Donea J, Huerta A. Finite element methods for flow problems. John Wiley; 2003.
- [22] Gresho PM, Sani RL, Engelman MS. Incompressible flow and the finite element method: advection–diffusion and isothermal laminar flow. John Wiley and Sons; 1998.
- [23] Oñate E. A stabilized finite element method for incompressible viscous flows using a finite increment calculus formulation. *Comput Methods Appl Mech Eng* 2000;182:355–70.
- [24] Oñate E, Garcia J, Idelsohn S, Pin FD. Finite calculus formulations for finite element analysis of incompressible flows. Eulerian, ale and Lagrangian approaches. *Comput Methods Appl Mech Eng* 2006;195:3001–37.
- [25] Oñate E, Nadukandi P, Idelsohn SR, Garcia J, Felippa C. A family of residual-based stabilized finite element methods for stokes flow. *Int J Numer Methods Fluids* 2011;65:106–34.
- [26] Oñate E, Idelsohn SR, Felippa CA. Consistent pressure Laplacian stabilization for incompressible continua via higher-order finite calculus. *Int J Numer Methods Eng* 2011;87:171–95.
- [27] Idelsohn SR, Oñate E. The challenge of mass conservation in the solution of free-surface flows with the fractional-step method: problems and solutions. *Int J Numer Methods Biomed Eng* 2010;26:1313–30.
- [28] Aubry R, Idelsohn S, Oñate E. Fractional step like schemes for free surface problems with thermal coupling using the Lagrangian PFEM. *Comput Mech* 2006;38:294–309.
- [29] McKenna F, Scott MH, Fenves GL. Nonlinear finite-element analysis software architecture using object composition. *J Comput Civil Eng* 2010;24(1):95–107.
- [30] Shewchuk JR. Triangle: engineering a 2D quality mesh generator and delaunay triangulator. In: Lin MC, Manocha D, editors. Applied computational geometry: towards geometric engineering. Lecture notes in computer science, vol. 1148. Springer-Verlag; 1996. p. 203–22 [from the first ACM workshop on applied computational geometry].
- [31] Edelsbrunner H, Mücke EP. Three-dimensional alpha shapes. *ACM Trans Graphics* 1994;13(1):43–72.
- [32] Oñate E, Idelsohn S, Pin FD, Aubry R. The particle finite element method an overview. *Int J Comput Methods* 2004;1(2):267–307.
- [33] Cremonesi M, Frangi A, Perego U. A lagrangian finite element approach for the analysis of fluid–structure interaction problems. *Int J Numer Methods Eng* 2010;84:610–30.
- [34] Koshizuka S, Oka Y. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nucl Sci Eng* 1996;123:421–34.
- [35] Martin J, Moyce WJ. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philos Trans R Soc London Ser A Math Phys Sci* 1952;244(882):312–24.
- [36] Madurapperuma MAKM, Wijeyewickrema AC. Inelastic dynamic analysis of an RC building impacted by a tsunami water-borne shipping container. *J Earthquake Tsunami* 2012;6(1):12500011–017.
- [37] Crisfield MA. Non-linear finite element analysis of solids and structures, vol. 1. John Wiley & Sons; 1991.