Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 193 (2004) 3935–3956
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A weak coupling algorithm for seabed–wave interaction analysis

J.G. Wang [a,*], T. Nogami [b,1], G.R. Dasari [b,2], P.Z. Lin [b]

[a] Tropical Marine Science Institute, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore
[b] Department of Civil Engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260, Singapore

Received 30 December 2002; received in revised form 22 October 2003; accepted 17 February 2004

Abstract

This paper proposes a weak coupling algorithm for the seabed–wave interaction analysis. This algorithm is based on the error integration over a time interval and space along fluid–seabed interfaces. With this weak coupling algorithm, seabed–wave interaction analysis can be easily carried out with a fluid solver in fluid domain and a meshless solver in seabed domain independently. The fluid solver is based on a two-step projection method for Navier–Stokes equation. The free surface of seawaters during wave propagation is traced through the volume of fluid (VOF). The meshless solver is based on the Biot's consolidation theory of saturated poroelasticity which considers the interaction of soil deformation and pore water pressure in seabed. Data exchange at the interface between fluid and porous medium domains is carried out through a point interpolation method. Interface condition is approximately satisfied in a variational sense through an iteration scheme between two solvers. This algorithm does allow not only non-matching mesh along the interface but also non-matching time step in different subdomains. Numerical examples show that this weak coupling algorithm is able to capture the deformation of interface and improve the computation accuracy and efficiency.

Keywords: Wave load; Biot consolidation theory; Oscillatory pore water pressure; Wave–soil mass interaction; Finite difference method; Meshless method

1. Introduction

The interaction of wave–porous medium such as seabed is a common engineering problem in coastal/offshore engineering because there are two domains occupied by fluid and porous medium, respectively. A simple model for this interaction problem is shown in Fig. 1. It has two subdomains, fluid domain and porous medium domain (seabed). Fluid domain is full of seawater and has a free surface at the top surface.

* Corresponding author. Tel.: +65-874-6591; fax: +65-779-1635.
E-mail address: tmswjg@nus.edu.sg (J.G. Wang).
[1] Retired.
[2] Current address: ExxonMobil Upstream Research Company, 3120 Buffalo Speedway, Houston, TX 77252-2189, USA.

0045-7825/$ - see front matter
doi:10.1016/j.cma.2004.02.011

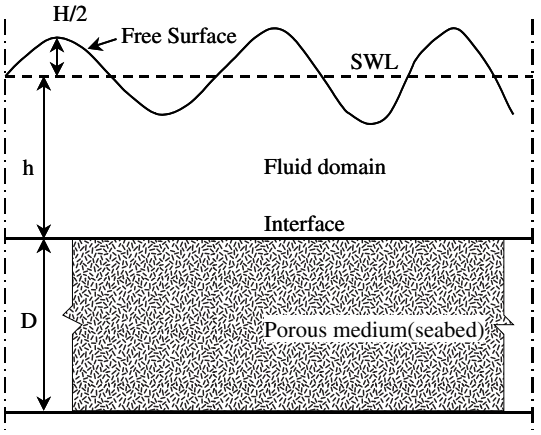

Fig. 1. Interaction problem of wave and porous medium.

When a wave is input from one side boundary, this wave will propagate in the water and generate wave load on the fluid–seabed interface. If the seawater is shallow, the wave load is large enough to cause the porous medium deformation and pore water flow. The seabed is deformable and saturated with pore water. When the wave load applies to the seabed surface, excess pore water pressure and deformation of soil skeleton will be generated in the seabed. The interface between the fluid domain and the porous medium will deform and the pore water will flow in or out, too. This deformation of seabed produces a movable seabed surface. This flow in/flow out (seepage) from the porous medium becomes a distributed sink for fluid domain. Both deformation and seepage will affect the dispersion of water wave in the fluid domain [1], thus the water pressure along the interface. On the other hand, the water pressure along the interface will cause deformation and seepage in the porous medium. This interface can finally reach a balance position in this seabed–wave interaction problem. How to analyze such an interaction problem and to achieve the balance position are key issues in coastal/offshore engineering as well as in the field of fluid–structure interactions [2].

Many numerical methods have been proposed to solve the interaction problem of fluid–structure [3–7]. A key issue is how to treat the compatibility condition (interface condition) along the interface between fluid and solid domains. There are two classes of numerical algorithms to treat this compatibility condition: The first one is the strong coupling algorithm which was proposed for the interaction problem of structure and fluid [5]. At the interface, a compatibility condition, which ensures fluid and solid domains neither detach nor overlap during deformation, is expressed as follows [6]:

$$\mathbf{V}_{\mathrm{F}}^{t} \cdot \mathbf{n} = \mathbf{V}_{\mathrm{S}}^{t} \cdot \mathbf{n}, \tag{1}$$

where $\mathbf{V}_{\mathrm{F}}^{t}$, $\mathbf{V}_{\mathrm{S}}^{t}$ are the velocities of fluid and structure at the time t, respectively, and $\mathbf{n}$ is the unit vector along normal direction of the interface. Theoretically, this coupling algorithm is complete because detaching and overlapping can be completely avoided at any time. However, this strong coupling algorithm has several demerits: First, this algorithm requires the discrete equations for structure and fluid domains to form a big system equation. In this system equation, stiffness of solid and fluid are very different, thus the system stiffness equation has bad condition number, usually causing numerical instability. Second, this algorithm requires matching mesh size along the interface. Matching mesh size on the both sides of interface is difficult in practice. For example, solid domain usually requires bigger mesh size than fluid domain in order to achieve the same accuracy. Thus, it is difficult to generate a matching mesh on both sides of the solid–fluid interface, particularly for the situations that mesh is independently generated in each subdomain. Treat-

ment of non-matching mesh is an essential issue in the solid–fluid interaction problems. Some affordable algorithms have been presented by using the Lagrange multipliers to impose this compatibility condition [7]. Arbitrary Lagrangian Eulerian formulation (ALE) method is a typical hybrid method for the non-matching mesh [8]. The methods based on the Lagrange multiplier technique satisfy the compatibility condition in a variational sense and require complex formulations and additional degrees of freedom. Kim proposed an interface element method (IEM) for non-matching interfaces in space based on moving least-square-based meshless method [9,10]. This IEM is better than ALE because no intermediate interface media is required and the accuracy is higher. Third, time-step size may be non-matching among subdomains. As indicated in Eq. (1), the compatibility condition requires the same variables along the interface at the same time. As we know, the time-step size depends on mesh size, material stiffness and wave frequency. In the fluid–structure interaction problem, fluid domain requires much smaller time-step size than solid domain. If the same time-step size is applied to both solid and fluid domains, the computation cost is much higher, particularly in solid domain. Spurious effect may appear for porous medium problem if too small time-step size is adopted [11]. Fourth, big system stiffness equation requires much more computational resources. Fluid domain usually requires much finer mesh, thus much bigger freedom is introduced. If porous medium and fluid domains combine together, the scale of the system stiffness equation becomes very big. Such a system requires big amount of memory and more CPU time.

Domain decomposition [12], partitioned procedure [13] and staggered scheme [14] were developed for the fluid–structure problems through an iteration scheme. Because fluid and structure occupy different subdomains, their system equations can be setup individually within their subdomains. This iteration scheme makes the subdomains satisfy the compatibility of deformation at domain interfaces. Therefore, the iteration scheme along the interface is the key issue for the success of domain decomposition algorithm. A simplified iteration scheme, one-time-step delay algorithm, was widely used in coupling algorithms as an approximation [15,16]. This algorithm relaxes the constraint in Eq. (1) into

$$\mathbf{V}_{\mathrm{F}}^{t} \cdot \mathbf{n} = \mathbf{V}_{\mathrm{S}}^{t+\Delta t} \cdot \mathbf{n}. \tag{2}$$

This algorithm partially separates the fluid solver and the solid solver in space. If the time-step size is sufficiently small, Eq. (2) has sufficient accuracy [17]. However, this algorithm still requires same time-step size and same node distribution along the interface. Otherwise, non-matching interface algorithm has to be introduced [7,9,10].

This paper proposes a weak coupling algorithm at the interface for the seabed–wave interaction analysis. The interface condition between fluid and porous medium domains is assumed to satisfy following weak coupling criterion:

$$\int_0^T \|(\mathbf{V}_{\mathrm{F}}^{t} - \mathbf{V}_{\mathrm{S}}^{t}) \cdot \mathbf{n}\| \, \mathrm{d}t \leqslant \varepsilon, \tag{3}$$

where ε is a tolerance error. Such a treatment can largely relax the constraints on the spatial distributions of nodes along the interface as well as the time-step size in different subdomains. This treatment has two important advantages: First, different solvers are used in fluid and porous medium domains independently during whole process $[0, T]$ (where T is an time interval). For example, finite difference method with two-step projection method is used in fluid domain and radial point interpolation meshless method (radial PIM) [18] is used in porous medium domain. Second, non-matching is allowed not only for the spatial distributions of nodes along the interface, but also for the time-step size in different subdomains.

This paper is organized as follows: Section 2 describes the governing equations for fluid and porous medium domains, respectively. In porous medium domain, the interaction of pore water pressure and soil deformation is described by the Biot's consolidation theory [19]. In fluid domain, the wave propagation with free surface is governed by the Navier–Stokes equation and volume of fluid (VOF) [20]. The

compatibility condition along the interface of fluid and porous medium domains is given, too. The numerical strategy for fluid domain and porous medium domain is discussed in Section 3. A fluid solver is based on the two-step projection method. The free surface of seawaters is traced through VOF in Eulerian coordinates and the velocity at the bottom is changeable. A meshless solver for the porous medium is developed from the Biot's consolidation theory. This solver is suitable for the movable surface problems under wave load. The treatment of interface condition through an iteration scheme is discussed in Section 4. A data mapping between fluid and porous medium domains is proposed through radial point interpolation technique. The initial condition is also suggested for the iteration scheme. Numerical examples on sinusoidal wave are studied to check the convergence of the weak coupling algorithm. Section 6 compares the results of the current weak coupling algorithm with closed-form solution. The applicability of the current algorithm to standing and solitary waves is explored. The conclusions are given in Section 7.

2. Governing equations for porous medium and fluid domains

2.1. Governing equation for porous seabed

This section considers only seabed domain V. If the seabed is fully saturated, it has two components of soil skeleton and pore water. Soil skeleton and pore water interact at micro-level when an external load such as wave load is applied to the seabed. This interaction can be described by macro-level Biot's consolidation theory [19]. The Biot's consolidation theory has following six physical concepts:

- Equilibrium equation of soil–water mixture

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = 0 \quad \text{in } V. \tag{4}$$

Its incremental form in time interval $[t, t+\Delta t]$

$$\frac{\partial \Delta\sigma_{ij}}{\partial x_j} + \Delta b_i = -\left(\frac{\partial \sigma^t_{ij}}{\partial x_j} + b^t_i\right) \quad \text{in } V. \tag{5}$$

- Relationship of displacement and strain for soil skeleton

$$\Delta\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial \Delta u_i}{\partial x_j} + \frac{\partial \Delta u_j}{\partial x_i}\right) \quad \text{in } V. \tag{6}$$

- Constitutive law of soil skeleton in differential form

$$\mathrm{d}\sigma'_{ij} = D_{ijkl}\,\mathrm{d}\varepsilon_{kl} \quad \text{in } V. \tag{7}$$

- Darcy's seepage law for pore water flow

$$q_i = -\frac{K_{ij}}{\gamma_w}\frac{\partial P}{\partial x_j} \quad \text{in } V. \tag{8}$$

- Terzaghi's effective stress principle

$$\sigma_{ij} = \sigma'_{ij} + P\delta_{ij}. \tag{9}$$

- Continuity equation

$$\frac{\partial \varepsilon_v}{\partial t} = \frac{\partial q_i}{\partial x_i}, \tag{10}$$

where σ_{ij}, σ'_{ij} and P are total stress tensor, effective stress tensor and excess pore water pressure at any time t and b_i the unit body force. Δu_i is the displacement increment and $\Delta\sigma_{ij}$, $\Delta\varepsilon_{ij}$ total stress and strain increments in time interval $[t, t+\Delta t]$. q_i is the discharge of excess pore water in ith direction. γ_w is the density of water, of which the value can be taken as 10 kN/m^3 in SI system. D_{ijkl} is the material matrix of soil skeleton determined by constitutive law of materials. K_{ij} is permeability tensor of soil skeleton which usually has non-zero components K_x in x-direction and K_y in y-direction, respectively. ε_v is the volume strain of soil skeleton:

$$\varepsilon_v = \frac{\partial u_i}{\partial x_i}. \tag{11}$$

For soil skeleton boundary

$$\begin{cases} u_i = \bar{u}_{i0} & \text{on } S_{\bar{u}} \times [0,\infty), \\ \sigma'_{ij} n_j = \overline{T}_i & \text{on } S_\sigma \times [0,\infty), \end{cases} \tag{12}$$

where $\mathbf{n} = \{ n_1 \quad n_2 \quad n_3 \}$ is the outwards normal direction and n_i is its directional cosine.
For pore fluid boundary

$$\begin{cases} P = P_0 & \text{on } S_p \times [0,\infty), \\ q_i = q_{i0} & \text{on } S_q \times [0,\infty). \end{cases} \tag{13}$$

For periodicity boundary

$$\begin{cases} P_l(x;t) = P_r(x+\mathbf{Y};t) \\ u_{li}(x;t) = u_{ri}(x+\mathbf{Y};t) \end{cases} \quad \text{on } S_Y \times [0,\infty), \tag{14}$$

where $\mathbf{Y}$ is the wavelength. Subscripts 'l' and 'r' refer to the left and right side boundaries.
Initial condition

$$\begin{cases} u_i = 0 \\ P = 0 \end{cases} \quad \text{on } V \times 0^-. \tag{15}$$

The whole boundary of domain V is closed: $S_{\bar{u}} \cup S_\sigma \cup S_Y = S_p \cup S_q \cup S_Y$. The interface boundary condition at the top surface of the porous medium domain is of special importance in the current algorithm. We will discuss it in details later.

2.2. Governing equation for wave propagation in fluid domain

The motion of incompressible fluid in a bounded domain Ω is described by Navier–Stokes equations. In the Eulerian coordinates, continuity and momentum equations are expressed as follows:

$$\frac{\partial U_i}{\partial x_i} = 0, \quad \frac{\partial U_i}{\partial t} + U_j \frac{\partial U_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial P}{\partial x_i} + g_i + \frac{1}{\rho}\frac{\partial \tau_{ij}}{\partial x_j} \qquad \text{in } \Omega, \tag{16}$$

where $i, j = 1, 2$ for two-dimensional flows. U_i denotes the ith component of the velocity, ρ the density of fluid, P the water pressure, g_i the ith component of gravitational acceleration. τ_{ij} is the viscous stress tensor which is (where μ is a viscosity coefficient):

$$\tau_{ij} = \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\delta_{ij}\frac{\partial U_k}{\partial x_k}\right). \tag{17}$$

A big issue to simulate the wave propagation in open space is to trace the free surface during computation. Because Eulerian coordinates basically are fixed, additional governing equation must be introduced to

describe this motion of free surface. The volume of fluid (VOF) method [20] provides a robust alternative for updating the free surface. Its basic idea is to trace the density change within each computational cell. The free surface can be determined through the density distribution at the corner of each cell [3]. A governing equation for the density is as follows:

$$\frac{\partial \rho}{\partial t} + U_i \frac{\partial \rho}{\partial x_i} = 0. \tag{18}$$

In the VOF, the density of a fluid in each computational cell is defined as the averaged density in the cell. The $\rho = 1$ if a cell is full of water, otherwise, $0 \leqslant \rho < 1$.

2.3. Boundary conditions

The boundaries for fluid domain include the interface between fluid and porous medium, free surface boundary and virtual boundary at both sides.

2.3.1. Conventional fluid boundary conditions

The virtual boundary is not physical boundary. This boundary is designed to reduce the size of computational domain. In this paper, virtual boundary refers to the left and right hand sides of fluid domain. Wave sources in seas are usually from horizontal infinity. All domain-based numerical methods such as finite difference method and finite element method are difficult to treat infinite domain. A simple way is to truncate the infinite domain into a finite computable domain through a cut-off boundary condition or virtual boundary. This virtual boundary is sometimes called periodic boundary for sinusoidal wave. Free surface is another important boundary for wave propagation in seawater. The top surface of fluid domain in Fig. 1 is the free surface. This free surface boundary condition has kinetic and dynamic components. The kinetic condition describes the change of geometry, while the dynamic condition gives the continuity of stress components that is as follows:

$$P - \mu\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i}\right) n_i n_j = \sigma_n, \tag{19}$$

where σ_n is a prescribed normal stress applied on the free surface.

2.3.2. Interface conditions

Interface between fluid and porous media domains is of special importance. Present interface algorithm is different from the algorithm that regards the porous medium as rigid skeleton [3,21] because the porous medium is deformable. Porous medium domain has the velocity of soil particles and the flux of pore water as well as pore water pressure, while fluid domain has flow velocity and water pressure. Fig. 2 describes the relationship of the flux, velocity of particles and flow velocity along the interface. If the flow velocity of pore water is relative to the skeleton of porous medium, the mass conservation equation, if the fluid is incompressible, can be expressed as follows:

$$\begin{aligned} &(\mathbf{q}_I \cdot n + \dot{\mathbf{u}}_I(1-n)) \cdot \mathbf{n} = \mathbf{U}_{II} \cdot \mathbf{n} \\ &P_I = P_{II}, \end{aligned} \tag{20}$$

where P_I is pore water pressure in porous medium and P_{II} water pressure in fluid domain. $\mathbf{U}_{II}$ is the flow velocity in fluid domain. n is the porosity of porous medium and $\mathbf{n}$ is the normal unit of interface. $\dot{\mathbf{u}}_I$ refers to the velocity of soil particles which is obtained through the differentiation of displacement: $\dot{\mathbf{u}}_I = \frac{\partial \mathbf{u}}{\partial t}$. The flux $\mathbf{q}_I$ is obtained through the Darcy law after the distribution of pore water pressure is known: $\mathbf{q}_I = \frac{\mathbf{k}}{\gamma_w} \frac{\partial P}{\partial \mathbf{n}}$.

If fluid viscosity is included, a shear stress induced by the horizontal fluid flow should be applied to the surface of the porous medium. This shear stress is computed using Eq. (17).

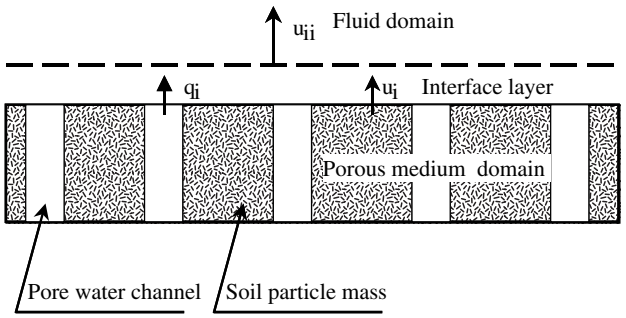

Fig. 2. Continuity of the mass at the interface (u_{ii}: the velocity of fluid particles, u_i: the displacement of soil particles, and q_i: discharge of pore water).

3. Numerical strategy in fluid and porous medium domains

Iteration scheme is used in the current numerical strategy. Fig. 3 gives the iteration loop among fluid domain Ω, porous medium domain V and interfaces. This loop has three components, fluid solver, porous solver and interface solver. Fluid solver computes the fluid velocity and water pressure in domain Ω and water pressure along the interface when the fluid velocity is given at the interface. This is a FDM solver which is developed from discretizing Navier–Stokes equation in Section 2.2. Porous solver is formulated by discretizing the Biot's consolidation theory in Section 2.1 through meshless methods [11,22,23]. It can give displacement and pore water pressure in domain V and particle velocity as well as flux along the interface if pore water pressure is given along the interface. Interface solver maps the data across the interface and checks the convergence of the iteration.

3.1. Porous solver for porous medium domain

Meshless method is employed to discretize the Biot's consolidation theory in Section 2.1. Because the seabed surface may experience large deformation or moving boundary during wave propagation, meshless methods are better than FEM for such problems. Let us consider a time interval of $[t, t + \Delta t]$. The

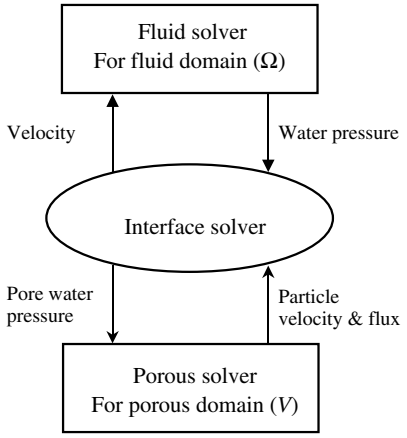

Fig. 3. Iteration among different domain solvers.

displacement increments of soil skeleton are $\Delta\mathbf{u}$ in the time interval and pore water pressure is $\mathbf{P}$ at the time $t + \Delta t$. We discretize above spatial variables $\Delta\mathbf{u}$ and $\mathbf{P}$ with a radial PIM [18] and temporal domain by finite difference method. The final system equation is obtained as follows [11,22,23]:

$$\begin{bmatrix} \mathbf{K} & \mathbf{K}_V \\ \mathbf{K}_V^{\mathrm{T}} & -\Delta t\theta\mathbf{K}_p \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u} \\ \mathbf{P} \end{Bmatrix} = \begin{Bmatrix} \Delta\mathbf{F}_{\mathrm{b}} + \Delta\mathbf{F}_{\mathrm{t}} + \mathbf{F}_{\mathrm{r}} \\ \Delta t(1-\theta)\mathbf{K}_p\mathbf{P}^t \end{Bmatrix}, \tag{21}$$

where $\mathbf{K}$ is the stiffness matrix of soil skeleton, $\mathbf{K}_V$ corresponds to the seepage force on the soil skeleton. θ is a parameter for time integration [11]. The force in Eq. (21) has three sources: body force increment ($\Delta\mathbf{F}_{\mathrm{b}}$), traction force increment ($\Delta\mathbf{F}_{\mathrm{t}}$) and residual force ($\mathbf{F}_{\mathrm{r}}$) due to un-equilibrium at the previous time step. Let the $\Delta\mathbf{u}_a$, $\mathbf{P}_a$ denote the displacement increments and pore water pressure within porous medium domain V. $\Delta\mathbf{u}_I$, $\mathbf{P}_I$ are the displacement increment and pore water pressure along the interface, respectively. Therefore, above equation can be rewritten as follows:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{bmatrix} \begin{Bmatrix} \Delta\mathbf{u}_a \\ \mathbf{P}_a \\ \Delta\mathbf{u}_I \end{Bmatrix} = \begin{Bmatrix} \mathbf{F}_{ua}(\mathbf{P}_I) \\ \mathbf{F}_{pa}(\mathbf{P}_I) \\ \mathbf{F}_{uI}(\mathbf{P}_I) \end{Bmatrix}. \tag{22}$$

This equation indicates that pore water pressure $\mathbf{P}_I$ generates not only pore water pressure and displacement within the porous medium domain, but also the displacement increment $\Delta\mathbf{u}_I$ as well as flux along the interface. Once the fluid solver gives the wave load, the response of porous medium is completely determined.

3.2. Solver for fluid domain

Navier–Stokes equation is basically strongly non-linear due to its advection term. An effective numerical algorithm, two-step projection method [24,25], is well developed for the numerical solutions of Navier–Stokes equation. After spatial and temporal discretizations, the Navier–Stokes equation has following discrete form:

$$\begin{bmatrix} S_{11} & S_{12} & S_{14} \\ S_{21} & S_{22} & S_{24} \\ S_{41} & S_{42} & S_{44} \end{bmatrix} \begin{Bmatrix} U_b \\ P_b \\ P_{II} \end{Bmatrix} = \begin{Bmatrix} F_{vb}(U_{II}) \\ F_{pb}(U_{II}) \\ F_{IIb}(U_{II}) \end{Bmatrix}, \tag{23}$$

where U_b, P_b are velocity and water pressure within fluid domain and P_{II} the water pressure along the interface. The velocity U_{II} along the interface can be calculated after the Biot's consolidation problem is solved in porous medium domain. Therefore, the water pressure P_{II} along the interface is completely determined by Eq. (23). Eq. (23) is a non-linear equation, because its stiffness matrix is the function of water pressure and velocity, $S_{ij} = S_{ij}(U_b, P_b, P_{II})$. An iteration algorithm such as two-projection method is implemented to solve this equation.

4. Treatment of interface condition

4.1. Weak coupling at the interface

Treatment of compatibility condition along the interface is an important issue for the problems of fluid–solid interactions [2,3,21]. Different coupling schemes have different treatments. For example, a strong coupling scheme [5] is formulated if the porous medium domain expressed by Eq. (22) is coupling with the fluid domain expressed by Eq. (23) at each time step and at each interface node. This strong coupling scheme requires much more computer resource and special numerical techniques. We can relax the com-

patibility condition to a variational sense or integration over zone, formulating a weak coupling scheme as expressed in Eq. (3). Weak coupling algorithm allows different domains to use different numerical strategy such as finite element method, finite difference method or meshless method. This relaxation of compatibility conditions can allow different nodes and time-step sizes within each subdomain.

The current iteration algorithm is shown in Fig. 4. Along the interface of fluid domain, the input data is the velocity of fluids which is the summation of the particle velocity and flux of pore water from the porous medium as indicated in Fig. 2. The output data along this interface is water pressure P_{II}^n where superscript n denotes the nth iteration between fluid and porous medium domains. This water pressure is used as the input of pore water pressure along the interface of porous medium (see Eq. (20)). For porous medium domain, the output along the interface is the flux (relative velocity) of pore water and particle velocity of soil. Their summation is denoted as V_a^n which is equal to the velocity of fluid U_{II}^n. If following relative errors are satisfied, the iteration is said to be convergent:

$$\frac{\|V_a^{n+1} - V_a^n\|}{\|V_a^n\|} \leqslant \varepsilon_1 \ (\%), \qquad \frac{\|P_{II}^{n+1} - P_{II}^n\|}{\|P_{II}^n\|} \leqslant \varepsilon_2 \ (\%), \tag{24}$$

where the norm is defined as an integral over time interval $[0, T]$ as well as the interface:

$$\|V_a^n\| = \int_0^T \int_S |V_a^n| \,\mathrm{d}s \,\mathrm{d}t, \qquad \|V_a^{n+1} - V_a^n\| = \int_0^T \int_S |V_a^{n+1} - V_a^n| \,\mathrm{d}s \,\mathrm{d}t, \tag{25}$$

where S refers to the interface. ε_1, ε_2 are relative errors for the velocity of soil particle and flux from porous medium domain as well as water pressure for fluid domain at the interface, respectively. Eq. (25) is also applicable to water pressure $\mathbf{P}_{II}$.

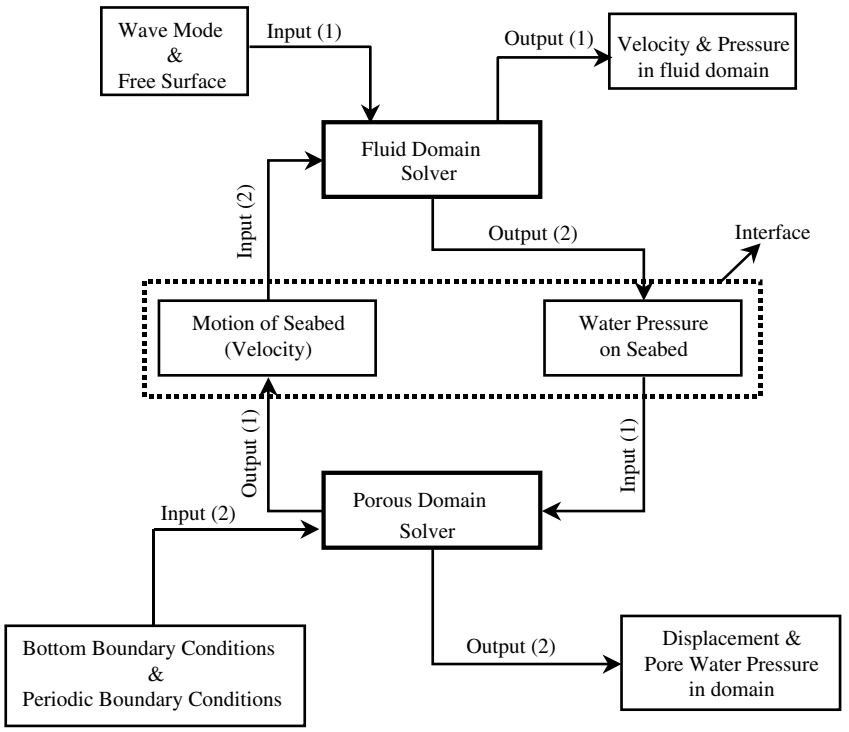

Fig. 4. Iteration algorithm between two solvers.

4.2. Initial values for iterations

Linear wave theory provides a good initial value for iterations. Let a sinusoidal wave propagate in irrotational and incompressible waters. If the seabed surface is non-movable and impermeable, the water pressure at the seabed interface is obtained from the linear wave theory [26]:

$$P = \frac{\rho g H}{2\cosh(kh)} \cos(kx - \omega t) \tag{26}$$

and the velocity of fluid particles is

$$\begin{aligned} U_x|_{z=-h} &= \frac{gkH}{2\omega} \frac{1}{\cosh kh} \cos(kx - \omega t), \\ U_y|_{z=-h} &= 0, \end{aligned} \tag{27}$$

where dispersion equation determines the relation of the angular velocity ω, the wave number k, and the depth of still water h:

$$\omega^2 = kg \tanh kh, \tag{28}$$

where H is wave height, and g the gravity. U_x is the velocity along x-direction and U_y is the velocity along y-direction. That the vertical velocity is zero means there is no flow in the normal direction. If the fluid is real (no viscosity), fluid particles can slip along the interface. However, when wave load applies to deformable seabed, the soil particle will move and pore water will flow in or out, thus the vertical velocity is non-zero. This non-zero velocity can be computed through iterations between fluid and porous medium domains.

4.3. Data mapping at the interface

The data exchange between fluid and porous medium domains can be carried out through a data-mapping algorithm along the interface. This data-mapping algorithm is important to the current weak coupling algorithm. This is a unique channel to exchange data because the current weak algorithm solves the responses of fluid and porous medium domains independently. If porous medium and fluid domains have identical node distribution along both sides of the interface as shown in Fig. 5(a), no data mapping is required in space. In practice, it is difficult to keep identical node distribution along the interface for both

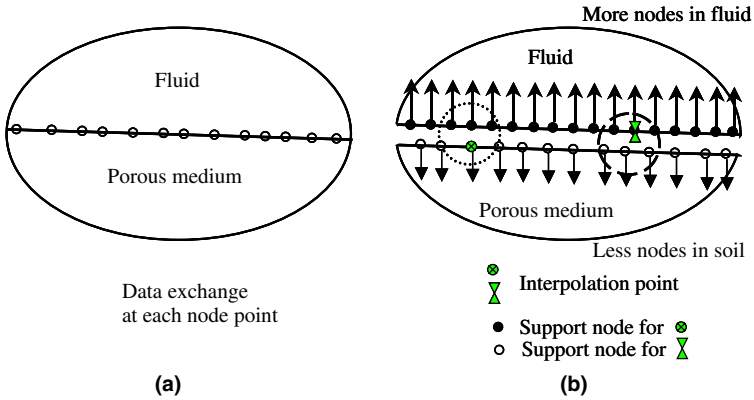

Fig. 5. Data-mapping algorithm at the interface: (a) strong coupling with one-to-one data mapping and (b) weak coupling with radial PIM mapping.

fluid and porous medium domains because the stiffness is much bigger difference. For the non-matching node distribution as shown in Fig. 5(b), a data mapping is required to exchange the data between two sides of the interface. Furthermore, fluid domain usually requires much smaller time-step size than porous medium domain. Data mapping is also necessary in time domain. This paper adopts the radial PIM [18] to construct the interpolations for both spatial and temporal domains along the interface. The principle for the radial PIM is briefly described as below.

There is a set of arbitrarily distributed points $P_i(\mathbf{x}_i)(i = 1, 2, \ldots, n)$ within an influence domain, where n is the number of nodes, and approximation function $u(\mathbf{x})$ has value u_i at each node point $\mathbf{x}_i$. Radial PIM method constructs this $u(\mathbf{x})$ through these nodes and the function values at these points. We express this approximation function $u(\mathbf{x})$ by a linear combination of radial basis $B_i(\mathbf{x})$ and polynomial basis $p_j(\mathbf{x})$:

$$u(\mathbf{x}) = \sum_{i=1}^{n} B_i(\mathbf{x}) a_i + \sum_{j=1}^{m} P_j(\mathbf{x}) b_j = \mathbf{B}^{\mathrm{T}}(\mathbf{x})\mathbf{a} + \mathbf{P}^{\mathrm{T}}(\mathbf{x})\mathbf{b}, \tag{29}$$

where a_i is the coefficient for $B_i(\mathbf{x})$ and b_j the coefficient for $p_i(\mathbf{x})$ (usually, $m < n$). The vectors are defined as

$$\begin{aligned}
&\mathbf{a}^{\mathrm{T}} = [\, a_1 \quad a_2 \quad a_3 \quad \cdots \quad a_n \,], \\
&\mathbf{b}^{\mathrm{T}} = [\, b_1 \quad b_2 \quad \cdots \quad b_m \,], \\
&\mathbf{B}^{\mathrm{T}}(\mathbf{x}) = [\, B_1(\mathbf{x}) \quad B_2(\mathbf{x}) \quad B_3(\mathbf{x}) \quad \cdots \quad B_n(\mathbf{x}) \,], \\
&\mathbf{P}^{\mathrm{T}}(\mathbf{x}) = [\, p_1(\mathbf{x}) \quad p_2(\mathbf{x}) \quad \cdots \quad p_m(\mathbf{x}) \,].
\end{aligned} \tag{30}$$

The radial basis function $B_i(\mathbf{x})$ is only function of distance in space. For example, in two-dimensional space, the radial basis function is

$$B_i(\mathbf{x}) = B_i(r_i) = B_i(x, y), \tag{31}$$

where r_i is a distance between interpolating point (x, y) and node (x_i, y_i) defined as

$$r_i = [(x - x_i)^2 + (y - y_i)^2]^{\frac{1}{2}}. \tag{32}$$

Polynomial basis functions have following monomial terms:

$$\mathbf{P}^{\mathrm{T}}(\mathbf{x}) = \begin{bmatrix} 1 & x & y & x^2 & xy & y^2 & \cdots \end{bmatrix}. \tag{33}$$

Let the $u(\mathbf{x})$ pass through all n scattered points within the influence domain, and the coefficients a_i and b_j in Eq. (29) can be completely determined. For example, the interpolation at the kth point has

$$u_k = u(x_k, y_k) = \sum_{i=1}^{n} a_i B_i(x_k, y_k) + \sum_{j=1}^{m} b_j P_j(x_k, y_k), \quad k = 1, 2, \ldots, n. \tag{34}$$

The polynomial term is an extra-requirement. Following constraint can insure the uniqueness of this approximation:

$$\sum_{i=1}^{n} P_j(x_i, y_i) a_i = 0, \quad j = 1, 2, \ldots, m. \tag{35}$$

Above two equations are expressed in matrix form as follows

$$\begin{bmatrix} \mathbf{B}_0 & \mathbf{P}_0 \\ \mathbf{P}_0^{\mathrm{T}} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}^e \\ \mathbf{0} \end{Bmatrix} \quad \text{or} \quad \mathbf{G} \begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \begin{Bmatrix} \mathbf{u}^e \\ \mathbf{0} \end{Bmatrix}, \tag{36}$$

where the nodal vector for function values is

$$\mathbf{u}^e = [\, u_1 \quad u_2 \quad u_3 \quad \cdots \quad u_n \,]^{\mathrm{T}}. \tag{37}$$

The coefficient matrix $\mathbf{B}_0$ on unknowns $\mathbf{a}$ is

$$\mathbf{B}_0 = \begin{bmatrix} B_1(x_1,y_1) & B_2(x_1,y_1) & \cdots & B_n(x_1,y_1) \\ B_1(x_2,y_2) & B_2(x_2,y_2) & \cdots & B_n(x_2,y_2) \\ \vdots & \vdots & \vdots & \vdots \\ B_1(x_n,y_n) & B_2(x_n,y_n) & \cdots & B_n(x_n,y_n) \end{bmatrix}_{n\times n}. \tag{38}$$

The coefficient matrix $\mathbf{P}_0$ on unknowns $\mathbf{b}$ is

$$\mathbf{P}_0 = \begin{bmatrix} P_1(x_1,y_1) & P_2(x_1,y_1) & \cdots & P_m(x_1,y_1) \\ P_1(x_2,y_2) & P_2(x_2,y_2) & \cdots & P_m(x_2,y_2) \\ \vdots & \vdots & \vdots & \vdots \\ P_1(x_n,y_n) & P_2(x_n,y_n) & \cdots & P_m(x_n,y_n) \end{bmatrix}_{n\times m}. \tag{39}$$

The distance is directionless, $B_k(x_i,y_i) = B_i(x_k,y_k)$. Unique solution is obtained if the inverse of matrix $\mathbf{G}$ or $\mathbf{B}_0$ exists:

$$\begin{Bmatrix} \mathbf{a} \\ \mathbf{b} \end{Bmatrix} = \mathbf{G}^{-1} \begin{Bmatrix} \mathbf{u}^e \\ \mathbf{0} \end{Bmatrix}. \tag{40}$$

The approximation function or interpolation is finally expressed as

$$u(\mathbf{x}) = \begin{bmatrix} \mathbf{B}^{\mathrm{T}}(\mathbf{x}) & \mathbf{P}^{\mathrm{T}}(\mathbf{x}) \end{bmatrix} \mathbf{G}^{-1} \begin{Bmatrix} \mathbf{u}^e \\ \mathbf{0} \end{Bmatrix} = \boldsymbol{\phi}(\mathbf{x})\mathbf{u}^e, \tag{41}$$

where the matrix of shape functions $\boldsymbol{\phi}(\mathbf{x})$ is defined by

$$\begin{aligned} &\boldsymbol{\phi}(\mathbf{x}) = [\, \phi_1(\mathbf{x}) \quad \phi_2(\mathbf{x}) \quad \cdots \quad \phi_i(\mathbf{x}) \quad \cdots \quad \phi_n(\mathbf{x}) \,], \\ &\phi_k(\mathbf{x}) = \sum_{i=1}^{n} B_i(\mathbf{x})\overline{G}_{i,k} + \sum_{j=1}^{m} P_j(\mathbf{x})\overline{G}_{n+j,k}, \end{aligned} \tag{42}$$

where $\overline{G}_{i,k}$ is the (i,k) element of matrix $\mathbf{G}^{-1}$.

Gaussian radial function is widely used in mathematics:

$$B_i(x,y) = \exp\left(-c\left(\frac{r_i}{r_{\max}} \right)^2 \right), \tag{43}$$

where c $(c \geqslant 0)$ is a shape parameter [27], and $r_{\max}$ is the radius of an influence domain.

Take an original function $y = \sin(x)$ as an example to check the interpolation accuracy. Fig. 6 shows the interpolation accuracy for the function when the interpolation procedure is as follows: First, a cluster of data point is generated along the x-axis. These nodes are uniformly distributed with an increment of 0.5 in domain $[0, 7.5]$. These node points are used to determine shape functions in which the shape parameter is taken as $c = 2.0$. Second, functions values at these node points are calculated from the original function $y = \sin(x)$. Thus, an interpolation is completely determined by shape functions and nodal values. Third, error function is computed through comparing interpolation and original function in the domain $[0, 7.5]$. In this domain, a set of new nodes with an increment of 0.1 is generated. Interpolation and original function values are calculated at each node. Their difference was summarized to form an error function. The same procedure is applied to form error function of derivative. Fig. 6(a) compares the function and derivatives within this domain and Fig. 6(b) is their error functions. The accuracy of interpolation is non-homogeneous at each node. The accuracy is higher in the central part and lower near edges, but is within 1%. Therefore, the current interpolation method can be applied to the data mapping along interface.

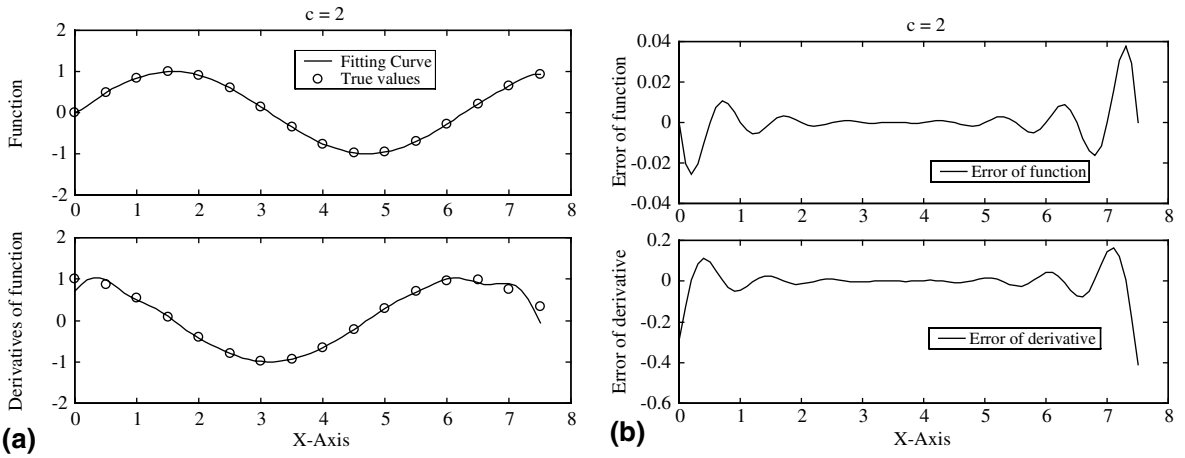

Fig. 6. Accuracy of curve fitting with radial point interpolation method: (a) comparison of data point and interpolation curve and (b) absolute error for curve fitting in whole domain.

For the current interface, above radial PIM is applied to a spatial-temporal coordinates which is composed of a curvilinear coordinate in space and time coordinate. All the equations are the same if a generalized distance including temporal dimension is used.

5. Verification of weak coupling algorithm

5.1. Initial condition

As pointed out in Section 4.2, the seabed is supposed to be fixed in the first iteration. The water pressure in the fluid domain is obtained through the fluid solver. Then the water pressure along the interface is applied to the porous medium. The porous solver computes particle velocity of soil and flux of pore water.

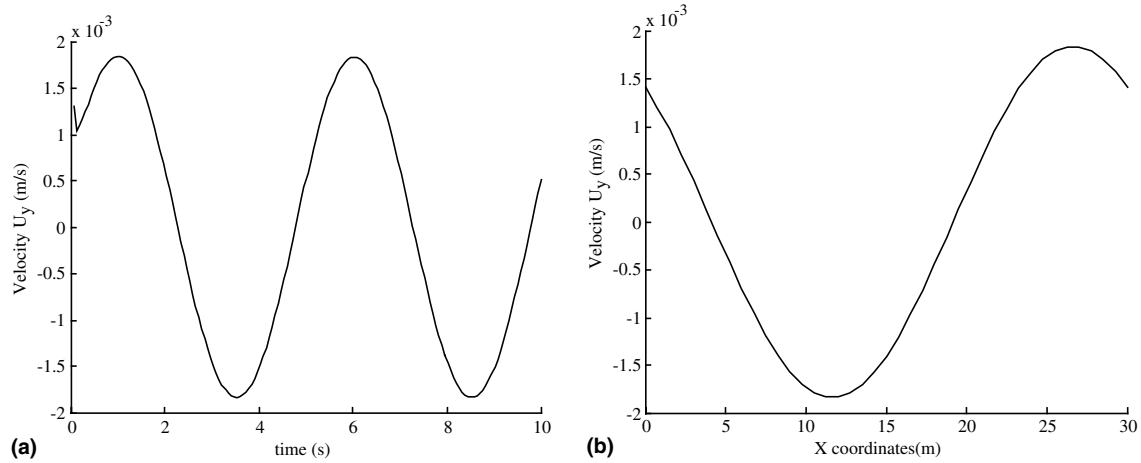

Fig. 7. Velocity U_y distribution along the seabed surface (sinusoidal wave): (a) velocity at the center of the interface and (b) velocity distribution along interface when time $t = 5$ s.

For a sinusoidal wave, the summation of flux and particle velocity is still harmonic as shown in Fig. 7 which has following form:

$$U_y = \dot{\mathbf{u}}^t + \mathbf{q}_t = V_{YA}\sin(kx - \omega t). \tag{44}$$

The particle velocity of soils can be calculated from the displacement history:

$$\dot{\mathbf{u}}^t = \frac{\partial \mathbf{u}}{\partial t} \approx \frac{\mathbf{u}^{t+\Delta t} - \mathbf{u}^t}{\Delta t}, \qquad V_y = \dot{u}_y^t. \tag{45}$$

The flux of pore water (q_t) at the soil–fluid interface is determined through Darcy's law:

$$q_t = \frac{k_y}{\gamma_w}\frac{P^t(y_2) - P^t(y_1)}{y_2 - y_1}, \tag{46}$$

where y_2 is the y-coordinates at the interface and y_1 is a point in porous medium domain but near the interface.

5.2. Computation parameters

A sinusoidal wave is assumed to be period of 5 s, wave height of 0.5 m, and water depth of 5 m. Thus the wavelength is approximate 30 m. The water has viscosity of 1.0×10^{-6} if it is considered. The soil domain is taken as 30 m in horizontal length and 10 m in thickness. We consider stiff soil and soft soil. The stiff soil has the Young modulus of 4.0×10^4 kN/m^2, Poisson's ratio of 0.3, porosity of 0.4 and isotropic permeability of 1.728×10^{-3} m/s. Pore water is assumed to be incompressible. The unit weight of water was taken as 10 kN/m^3. The soft soil has the Young's modulus of 4.0×10^2 kN/m^2 and the other parameters are assumed to be the same as stiff soil. A finite difference grid is used for fluid solver and a meshless model is generated for

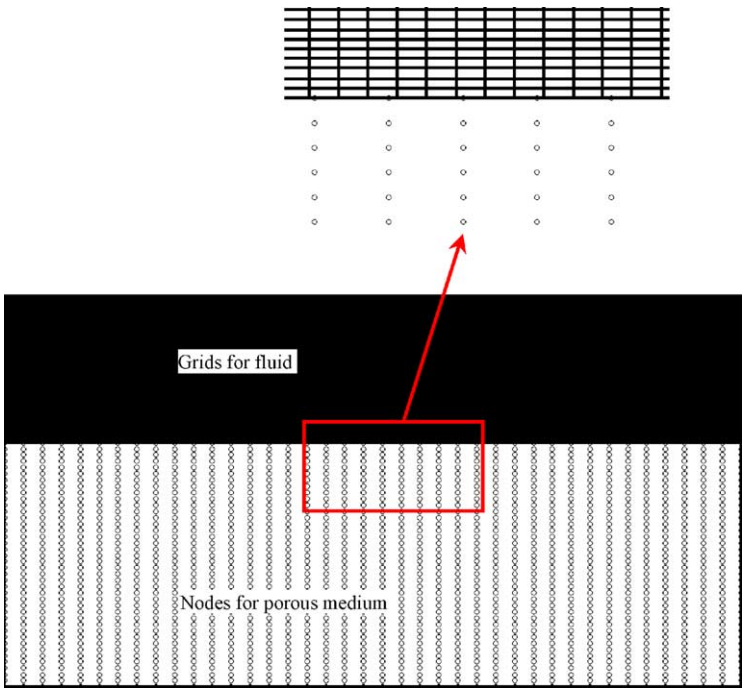

Fig. 8. Grids for finite difference method and node distribution for meshless method.

porous solver. This numerical model is shown in Fig. 8 in which the interface is amplified for details. The finite difference grid spaces 0.3 m horizontally and 0.1 m vertically. The meshless model is constructed with node spacing of 0.75 m horizontally and 0.25 m vertically. This example uses uniform distribution of nodes over whole porous medium domain, because the effect of regular or irregular distributions of nodes is neglectable for meshless solver [11,22,23]. Although both domains are discretized by uniform distribution of nodes, nodes are still un-matching along interface.

5.3. Convergence study

The convergence rates are shown in Fig. 9(a) for stiff soil and Fig. 9(b) for soft soil, respectively. The time interval for integration is taken as $[0, 10s]$ for stiff soil and $[0, 8s]$ for soft soil. Only 3–4 iterations are required to satisfy convergence criterion ($\varepsilon_1 = 0.01\%$ and $\varepsilon_2 = 0.01\%$) regardless of the viscosity of fluid. When the soil is soft, the flux and the particle velocity at the interface are higher. This velocity is far away from the initial value as shown at the first iteration in Fig. 9(b). After the first iteration, the velocity and the

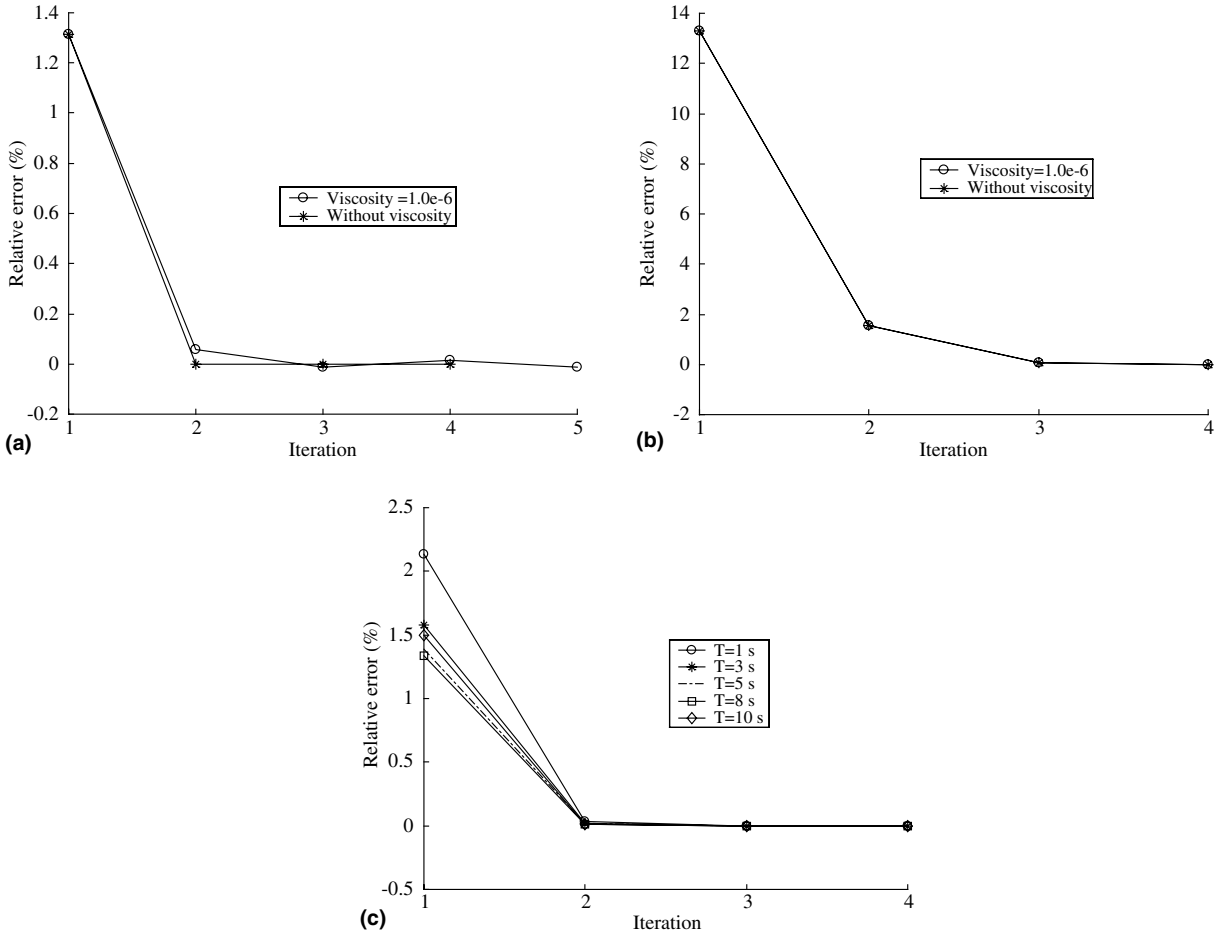

Fig. 9. Convergence rate of the weak coupling algorithm: (a) stiff soil case, (b) soft soil case and (c) convergence under different time intervals (stiff soil).

water pressure approach to true values. Furthermore, the viscosity of fluid has a little effect on the convergence rate, because the convergence rate is higher without viscosity.

The effect of time intervals on convergence rate is shown in Fig. 9(c). This figure is obtained for the stiff soil. When the time interval is short, the convergence is fast although the initial iteration has less accuracy. After only two iterations, the accuracy is sufficient high. Longer time interval involves more iteration, but the total computation time is still lower. Table 1 compares the computation time in each process. The shorter the time interval, the more time consumes in the data exchange and the longer total time is. Determination of a suitable time interval before computation is difficult. If the problem is almost linear in time domain, longer time interval is recommended. Otherwise, shorter time interval is used to capture the non-linear properties along the interface. The accuracy of the current weak coupling algorithm depends on the accuracy of fluid solver, porous solver and interface solver. Fluid solver is non-linear due to free surface and advection. However, the porous solver is linear if the interface deformation is small. When the deformation of porous medium is small, the integration of above relative error can be carried out along only initial interface. Otherwise, the integration should be along the updated interface. Fig. 10 compares the final distribution of water pressures with the closed-form solution of Eq. (26) when the soil is stiff. The current water pressure is a little lower than the closed-form solution, because the closed-form solution assumes a fixed and impermeable seabed.

Table 1
Time consumed in each process

Interval T (s)	Steps for fluid solver	Steps for porous solver	Iterations in each T	Times for data exchange
10	5000	100	4	4
8	5000	100	4	8
5	5000	100	3	6
3	5000	100	3	12
1	5000	100	3	30

Note: (1) two periods are computed, (2) above interval is longer than the time steps for both fluid solver and porous solver. If the interval is smaller than the time step of porous solver but longer than fluid solver, the data exchange time will increase dramatically.

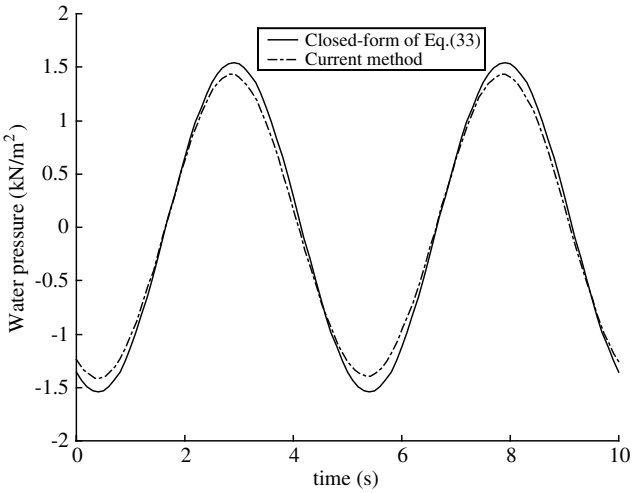

Fig. 10. Comparison of the current method with closed-form solution (stiff soil).

6. Parametric study

This section will explore the advantages of the current weak coupling algorithm through numerical examples and parametric study. Water pressure and velocity of soil particles along the interface are main concerns. The wave types included sinusoidal wave, standing wave and solitary wave.

First, water pressure along the interface is compared between the current method and closed-form solution. Fig. 11(a) shows a typical distribution of water pressure for the fixed seabed and the current coupling seabed when the soil is soft. The top surface of the soft seabed will move greatly under water pressure. This motion has some impact on the water pressure along the interface. For example, if the velocity of seabed surface can reach the magnitude of 1.15×10^{-1} m/s, the water pressure along the interface will increase approximately 17% as shown in Fig. 11(b). Fig. 12 compares vertical displacements along the

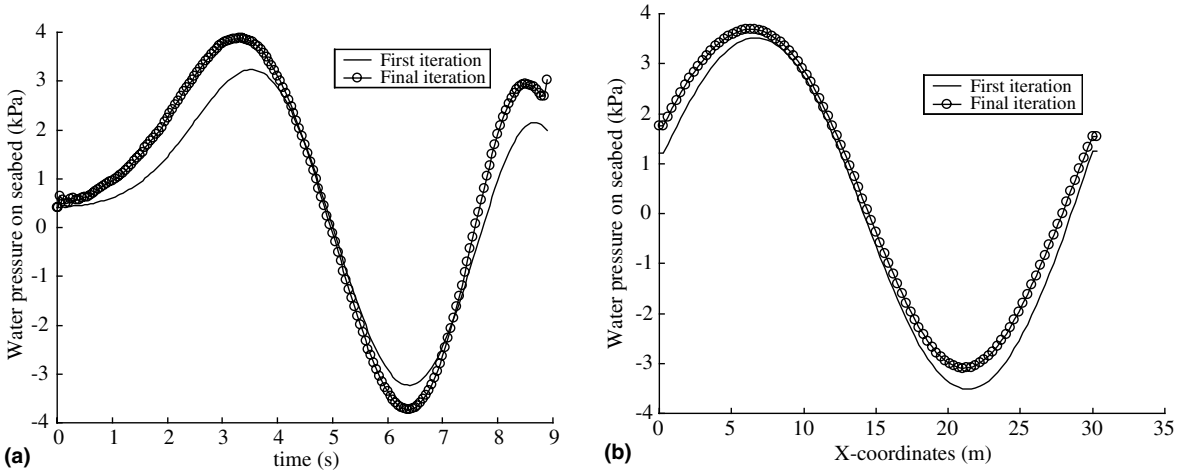

Fig. 11. Initial and final water pressure along interface (soft soil): (a) water pressure variation in time and (b) water pressure variation in space.

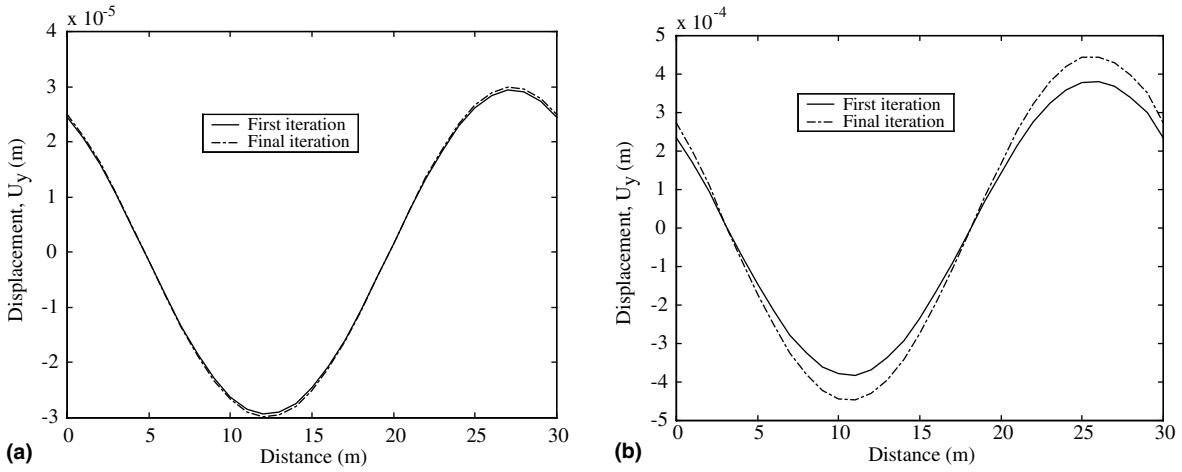

Fig. 12. Comparison of vertical displacement: (a) for stiff soil and (b) for soft soil.

interface at initial and final iterations. The displacement is little different for stiff soil; however, the difference is obvious for soft soil. The current algorithm predicts higher displacement. This result is consistent with the water pressure along the interface. Therefore, the stiffness of seabed has strong effect on the distribution of both water pressure and displacement along the interface. The pore water pressure and drag force within the porous medium will be affected correspondingly. Furthermore, for sinusoidal waves, the wave-induced water pressure along the interface can be normalized by wave height if the ratio of wave height to water depth is less than 0.14 as shown in Fig. 13.

Second, the current weak coupling algorithm is applicable to other linear waves. For example, a standing wave is the superposition of linear waves. If a standing wave exists in the fluid domain, the current algorithm still uses $V_y = 0$ to start the first iteration. After the second iteration, the velocities of seabed surface at different times are shown in Fig. 14. These velocities will revise the distribution of water pressure along the

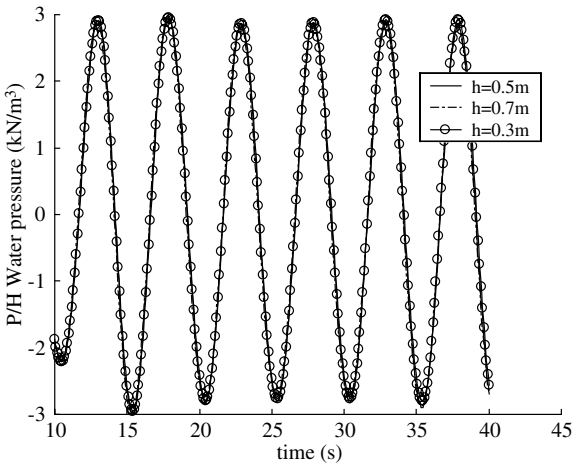

Fig. 13. Effect of wave height on the water pressure along interface.

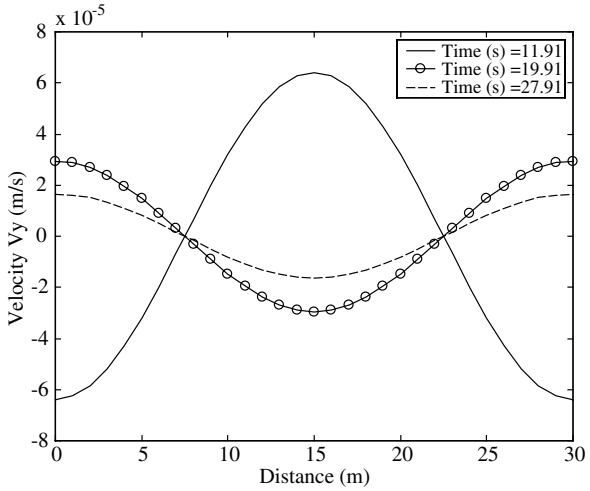

Fig. 14. Vertical velocity of soil particles along the interface under standing wave.

interface in the subsequent iteration. Conventional method assumes the seabed fixed to get water pressure along the interface, and applies this water pressure to a deformable seabed [22,23] and gets pore water pressure and displacement along the interface. Therefore, conventional method is the zero-order iteration of the current algorithm. Obviously, the current method is of higher accuracy because the effect of seabed motion is considered. Furthermore, in geotechnical engineering, closed-form solution for the response of

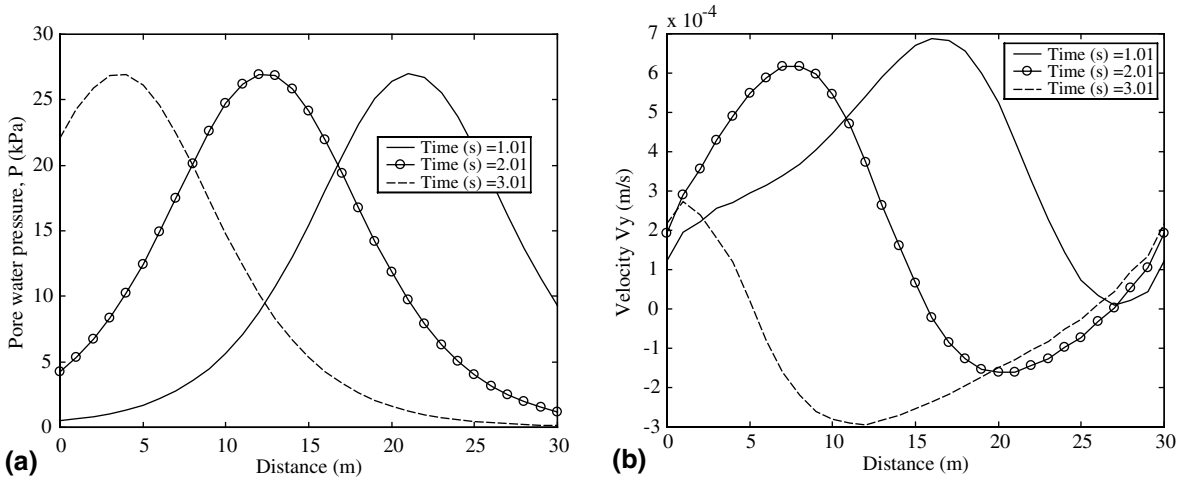

Fig. 15. Water pressure and particle velocity along interface (solitary wave): (a) pore water pressure and (b) vertical velocity of particle.

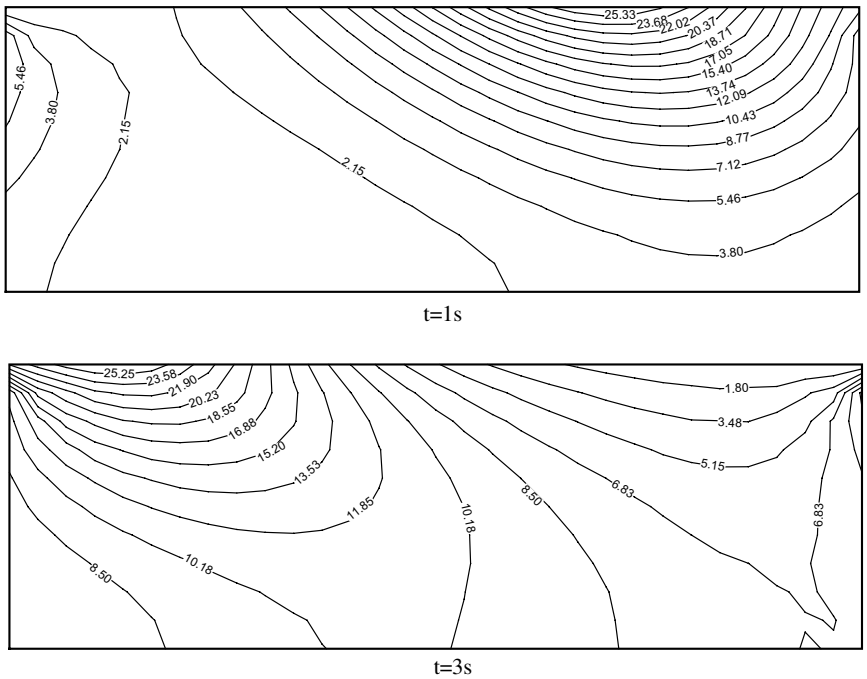

Fig. 16. Contours of pore water pressure at different times (solitary wave).

the seabed is only developed if the water pressure along the interface can be analytically expressed. For general wave types, it is impossible to obtain an analytical expression of the water pressure along interface, thus is impossible to get a closed-form solution of seabed response. In the current algorithm, only numerical distribution of water pressure is required along the interface. This largely simplifies the requirement along interface and improves the computation accuracy.

Third, the current weak coupling algorithm is also applicable to non-linear water waves such as solitary wave. Solitary wave is neither oscillatory nor does it exhibit a trough. A solitary wave lies entirely above the still-water level. Because both its wavelength and its period are infinite, only the ratio of wave height to water depth is the wave parameter. A wavelength is defined as $L = \frac{2\pi}{k}$ and $k = \sqrt{\frac{3H^2}{4h^3}}$. Furthermore, celerity is $c = \sqrt{g(h+H)}$. The current algorithm generates wave-induced water pressure through iteration between the fluid and the porous medium domains. If the wave height is $H = 2.7$ m and the water depth is $h = 5.0$ m, the wavelength is approximately 30 m. Above parameters are used in the computation of fluid domain. The porous medium is stiff soil. Fig. 15(a) shows the pore water pressure along the interface after four iterations and Fig. 15(b) shows the vertical velocity of soil particles. This velocity partially represents the motion of seabed surface under solitary wave. The contours of pore water pressure at $t = 1$ and 3 s are shown in Fig. 16. They are all positive.

7. Conclusions

This paper presents a weak coupling algorithm to study the wave–seabed interaction problem. This algorithm uses spatial and temporal integration over the interface and a time interval to treat the fluid–seabed interface condition. Data exchange is implemented by a radial point interpolation method. This method is particularly suitable for non-matching node distributions and non-matching time-step sizes in fluid and the porous medium domains. Numerical studies were carried out to check the convergence and accuracy of the current algorithm. Following conclusions are made from these studies.

First, the weak coupling algorithm is usable in the seabed–wave interaction problems whether the waves are linear or non-linear. For linear wave, our examples use the ratio of wave height to water depth of 0.14. The convergence can be achieved within 3–5 iterations over a range of time intervals. For non-linear wave such as solitary wave, the convergence can be also achieved within 3–5 iterations.

Second, this weak coupling algorithm allows different subdomains to use different solvers independently. This provides an effective way to capture the characteristics in each subdomain. For example, the fluid solver is based on a two-step projection method, and the porous solver is based on a meshless method. Two solvers are completely different in numerical algorithms. The finite difference method can easily capture the discontinuity of derivatives and the free surface in fluid domain, while the meshless method can easily adapt the local intensity of stress/strain in porous medium. The weak coupling algorithm can take full advantages of the solvers in different domains.

Third, this weak coupling algorithm allows non-matching distribution of nodes not only in space but also in time domain. The data exchange is carried out by the radial point interpolation along the interface of fluid and porous medium domains. This property of non-matching in spatial and temporal dimensions is especially useful in the fluid–solid interaction problems because fluid domain usually requires denser grid nodes and smaller time-step size. The spatial and temporal integration relaxes the consistency constraint along the interface. If a closed-form solution in a subdomain is available, the solver in that subdomain can be completely replaced and the convergence can be also improved. Anyway, the fixed seabed can be always used as the initial value in the iterations of the current algorithm.

Fourth, seabed properties will affect the wave propagation in seawaters and thus the water pressure along the interface. A big difference between the current weak coupling algorithm and the conventional method lies to the particle velocity along the interface. The velocity is important to the sediment transport

near the interface. Stiff soil has lower velocity of particles and soft soil has higher velocity. If this velocity is higher than some critical value, the water pressure at the interface will increase significantly.

Acknowledgement

This study was financially supported by the US Office of Navy Research under grant number N00014-01-1-0457.

References

[1] J.F. Lee, Y.J. Lan, On waves propagating over poro-elastic seabed, Ocean Engrg. 29 (2002) 931–946.
[2] H.J.P. Morand, R. Ohayon, Fluid Structure Interaction, Wiley, New York, 1995.
[3] P.L.F. Liu, P.Z. Lin, K.A. Chang, T. Sakakiyama, Numerical modeling of wave interaction with porous structures, J. Waterways Port, Ocean Coastal Engrg. 125 (6) (1999) 322–330.
[4] M. Hafez, K. Oshima, Computational Fluid Dynamics Review, Wiley, Chichester, 1995.
[5] Q. Zhang, T. Hisada, Investigations of the coupling methods for FSI analysis by FEM, Proc. Jpn. Mech. Soc. (A) 67 (662) (2001) 1555–1562.
[6] B. Alazmi, K. Vafai, Analysis of fluid flow and heat transfer interfacial conditions between a porous medium and a fluid layer, Int. J. Heat Mass Transfer 44 (2001) 1735–1749.
[7] C. Farhat, J. Mandel, The two-level FETI method for static and dynamic plate problems, Part I: an optimal iterative solver for biharmonic systems, Comput. Methods Appl. Mech. Engrg. 155 (1998) 129–151.
[8] J. Sarrate, A. Huerta, A. Donea, Arbitrary Lagrangian Eulerian formulation for fluid–rigid body interaction, Comput. Methods Appl. Mech. Engrg. 190 (24–25) (2001) 3171–3188.
[9] H.G. Kim, Interface element method: treatment of non-matching nodes at the ends of interface between partitioned domains, Comput. Methods Appl. Mech. Engrg. 192 (2003) 1841–1858.
[10] H.G. Kim, Interface element method (IEM) for a partitioned system with non-matching interfaces, Comput. Methods Appl. Mech. Engrg. 191 (2002) 3165–3194.
[11] J.G. Wang, G.R. Liu, P. Lin, Numerical analysis of Biot's consolidation process by radial point interpolation method, Int. J. Solids Struct. 39 (6) (2002) 1557–1573.
[12] G.E. Sandberg, P.A. Hansson, M. Gustavsson, Domain decomposition in acoustic and structure-acoustic analysis, Comput. Methods Appl. Mech. Engrg. 190 (24–25) (2001) 2979–2988.
[13] S. Piperno, C. Farhat, Partitioned procedures for the transient solution of coupled aeroelastic problems, Part II: energy transfer analysis and three-dimensional applications, Comput. Methods Appl. Mech. Engrg. 190 (24–25) (2001) 3147–3170.
[14] B.A. Schrefler, L. Simoni, E. Turska, Standard staggered and staggered Newton schemes in thermo-hydro-mechanical problems, Comput. Methods Appl. Mech. Engrg. 144 (1997) 93–109.
[15] A. Beckert, H. Wendland, Multivariate interpolation for fluid–structure-interaction problems using radial basis functions, Aerospace Sci. Technol. 5 (2001) 125–134.
[16] C. Farhat, M. Lesoinne, Higher-order staggered and subiteration free algorithms for coupled dynamic aeroelasticity problems, in: 36th Aerospace Sciences Meeting and Exhibit, AIAA 98-0516, Reno/NV, 1998.
[17] G.P. Guruswamy, A review of numerical fluid/structures interface methods for computations using high-fidelity equations, Comput. Struct. 80 (2002) 31–41.
[18] J.G. Wang, G.R. Liu, A point interpolation meshless method based on radial basis functions, Int. J. Numer. Methods Engrg. 54 (2002) 1623–1648.
[19] M.A. Biot, General theory of three-dimensional consolidation, J. Appl. Phys. 12 (1941) 155–164.
[20] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.
[21] N. Massarotti, P. Nithiarasu, O.C. Zienkiewicz, Natural convection in porous medium—fluid interface problems—a finite element analysis by using the CBS procedure, Int. J. Numer. Methods Heat Fluid Flow 11 (5) (2001) 473–490.
[22] M.R. Karim, T. Nogami, J.G. Wang, Analysis of transient response of saturated porous elastic soil under cyclic loading using element-free Galerkin method, Int. J. Solids Struct. 39 (2002) 6011–6033.
[23] J.G. Wang, B.Y. Zhang, T. Nogami, Wave-induced seabed response analysis by radial point interpolation meshless method, Ocean Engrg. 31 (2004) 21–42.
[24] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[25] S. Hadush, A. Yashima, R. Uzuoka, S. Moriguchi, K. Sawada, Liquefaction induced lateral spread analysis using the CIP method, Comput. Geotech. 28 (2001) 549–574.
[26] O.S. Madsen, Wave climate of the continental margin: elements of its mathematical description, in: D.F. Stanley, D.J.P. Swift (Eds.), Marine Sediment Transport and Environmental Management, John Wiley, New York, 1976, pp. 65–90.
[27] J.G. Wang, G.R. Liu, On the optimal shape parameters of radial basis functions used for 2-D meshless methods, Comput. Methods Appl. Mech. Engrg. 191 (23–24) (2002) 2611–2630.