Engineering Analysis with Boundary Elements 64 (2016) 90–100

Contents lists available at ScienceDirect

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Mesh-free simulation of liquid sloshing subjected to harmonic excitations

Nan-Jing Wu [a,*], Shih-Chun Hsiao [b], Han-Lun Wu [b]

[a] Department of Civil and Water Resources Engineering, National Chiayi University, Chiayi City 600, Taiwan
[b] Department of Hydraulic and Ocean Engineering, National Cheng Kung University, Tainan City 701, Taiwan

ARTICLE INFO

Article history:
Received 28 May 2015
Received in revised form
31 August 2015
Accepted 3 December 2015
Available online 18 December 2015

Keywords:
Oscillating tank
Polynomial collocation method
Meshless
Liquid sloshing

ABSTRACT

In this study a mesh-free numerical model for simulating 3-D free-surface potential flows is established. A time-marching scheme in Lagrangian aspect is chosen for the specification of boundary conditions on the moving and deforming free surface while a local polynomial collocation method is applied for solving the Laplace equation at each time step. This collocation method is employed because the partial derivatives of the solution are calculated accurately. The trajectory of each free-surface node can thus be predicted precisely due to the accurate estimation of the partial derivatives of velocity potential, which represent components of the velocity vector at that specific node. The numerical model is applied to the simulation of free surface waves by the liquid sloshing in rectangular, square and cylindrical swaying tanks. Fairly good agreements are observed in the comparison of numerical results with experimental data. Because the partial derivatives of the velocity components are accurately calculated, the pressure distribution in the domain can also be acquired by solving the pressure Poisson equation separately.

1. Introduction

Water wave problems are usually treated as potential flow problems governed by the Laplace equation subjected to two nonlinear free surface boundary conditions. Due to the deformation of the free surface, mesh re-generation is usually indispensable if one uses a grid-based method to solve this kind of problems. Mesh generation, which means construction of the connectivity among the nodes, is a tedious task. Because potential flows are governed by the Laplace equation, the Boundary Element Method (BEM, also denominated as Boundary Integral Equation Method, BIEM) is mostly employed to this kind of problems [1–7].

A mesh-free method, which is named as Method of Fundamental Solutions (MFS), was applied to solve the Laplace equation in the fully nonlinear water wave problems [8–10]. When using MFS, one has to place source points outside the domain. Because the values of the fundamental solutions are just related to the distances from the source points and collocation is only needed on the boundaries, MFS could be regarded as a boundary type RBF collocation method. Though MFS could be employed to fully nonlinear water wave problems, its applicability is still limited because numerical blow-up occurs when the free surface approaches too close to the source points. The singular boundary method (SBM) [11,12] which is regarded as a kind of modified MFS might be an alternative for solving water wave problems.

A domain type RBF Collocation Method that guarantees the accurate estimation of partial derivatives of the velocity potential on the free surface was proposed in Ref. [13]. By using a Lagrangian time-marching scheme, the trajectories of the free surface nodes can be precisely predicted. Both 2-D and 3-D liquid sloshing in a rectangular water tank were simulated. However, the full matrix formed in that method limits its applicability to further large-scale problems. In Ref. [14,15], the 3-D model of Ref. [13] was modified to 2-D for the investigation on the propagation and run-up of nonlinear waves generated by landslides.

Besides treating water wave flows as potential flows, one could also choose Navier–Stokes equation or Reynolds Averaged Navier–Stokes equation models, such as models using Arbitrary Lagrangian–Eulerian method (ALE) [16] or Volume of Fluid method (VOF) [17] to describe the free surface, or models treating the fluid as stacks of separate particles [18,19]. Models using ALE or VOF are grid-based while particle models in Ref. [18,19] are meshless. Results of these models are more close to the real flow. However, these models could be more time consuming and require more computer memory storage.

Modifying the Finite Point Method (FPM) of Ref. [20,21], a local polynomial collocation method for the purpose of solving general partial differential equations was proposed [22]. It is a localized meshless method thus matrix formed in the collocation process is very sparse. In Ref. [13,23], it was suggested that at boundary

* Corresponding author.
E-mail address: njwu@mail.ncyu.edu.tw (N.-J. Wu).

http://dx.doi.org/10.1016/j.enganabound.2015.12.001
0955-7997/

nodes the governing equation is satisfied as well as boundary conditions are could make the numerical results more accurate, especially the partial derivatives around the boundary. Following this, the method proposed in Ref. [22] becomes more robust than conventional collocation methods. Adopting the time-marching scheme for the free surface proposed in Ref. [13] and the local polynomial collocation method proposed in Ref. [22], a numerical model for the simulation of 2-D free surface potential flows was developed [24]. The trajectories of the fluid particles in a swaying tank and the fully nonlinear water wave on the free surface were investigated. Another application of this numerical model was on the study of solitary wave generation by a piston type wave maker [25].

In this study, we extend the 2-D model [24,25] to 3-D. We also make a small modification for the local polynomial collocation method in Ref. [22] so the process of assembling the global matrix is more straightforward. This numerical model is applied to the simulation of free surface waves in the liquid sloshing of rectangular, square and cylindrical swaying tanks.

2. The time marching scheme for simulating free-surface potential flows

For inviscid incompressible fluids, the fluid velocity $\vec{v}(x,y,z,t)$ can be expressed as the gradient of the velocity potential ϕ due to the irrotationality.

$$\vec{v} = u\vec{i} + v\vec{j} + w\vec{k} = \nabla\phi \tag{1}$$

in which $\nabla = \frac{\partial}{\partial x}\vec{i} + \frac{\partial}{\partial y}\vec{j} + \frac{\partial}{\partial z}\vec{k}$ is the gradient operator. The flow is governed by the Laplace equation.

$$\nabla^2\phi = 0 \tag{2}$$

where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is named as the Laplace operator. For free surface flows in the gravity field, two boundary conditions are to be satisfied on the free surface. They are

$$\nabla\phi = \frac{d\vec{x}}{dt}, \vec{x} \in \text{ the entire domain, including the free surface} \tag{3}$$

$$\frac{D\phi}{Dt} = -gz + \frac{1}{2}\nabla\phi\cdot\nabla\phi, \vec{x} \in \text{ the free surface} \tag{4}$$

in which g is the gravity acceleration. These two equations are the kinematic and dynamic boundary conditions respectively. Both of them have been transformed onto the Lagrangian aspect. At a fluid–solid interface, the no-flux boundary condition has to be satisfied. That is

$$\vec{n}\cdot\nabla\phi = \vec{n}\cdot\vec{v}_b, \vec{x} \in \text{ the fluid} - \text{solid interface} \tag{5}$$

where $\vec{n}$ is the unit normal vector outward from the domain and $\vec{v}_b = u_b\vec{i} + v_b\vec{j} + w_b\vec{k}$ is the velocity of the moving solid boundary.

For solving this kind of time-dependent problems, the time domain firstly has to be discretized. At each time step, the Laplace equation needs to be solved once to obtain the velocity potential in the entire domain thus to further determine the velocity. Boundary positions are updated by the given motion of the solid boundaries and the prediction from the time marching process of the free-surface boundary. The second order central difference to Eq. (4) was employed in Ref. [13].

$$\left(\phi|_{\vec{x}=\vec{x}_j}\right)^{(n)} = \left(\phi|_{\vec{x}=\vec{x}_j}\right)^{(n-2)} + 2\Delta t\left[\left(-gz + \frac{1}{2}\nabla\phi\cdot\nabla\phi\right)_{\vec{x}=\vec{x}_j}\right]^{(n-1)} \tag{6}$$

where $\vec{x}_j$ denotes the position of the j^{th} node and this equation is only valid in case the node is on the free surface. In this formulation, the required data on the right-hand side at the n^{th} time step are already known. What one needs to do first is just to determine the position of each traced 'particle', $\vec{x}_j^{(n)}$. When the velocity potential in the entire domain is obtained, the velocity at each of the nodes whether inside the domain or on the boundaries can be estimated accurately. Therefore, by the second-order finite difference scheme in the time domain one can obtain the following formula from Eq. (3).

$$\vec{x}_j^{(n)} = \vec{x}_j^{(n-2)} + 2\Delta t(\nabla\phi|_{\vec{x}=\vec{x}_j})^{(n-1)} \tag{7}$$

Here it should be noted that this equation is valid for all the nodes. For better numerical stability, the Crank–Nicolson formula can then be applied.

$$\vec{x}_j^{(n)} = \vec{x}_j^{(n-1)} + \frac{\Delta t}{2}\left[(\nabla\phi|_{\vec{x}=\vec{x}_j})^{(n)} + (\nabla\phi|_{\vec{x}=\vec{x}_j})^{(n-1)}\right] \tag{8}$$

Note that there is no need to solve the Laplace equation again because there is barely any difference whether the free-surface velocity potential at $\vec{x}_j^{(n)}$ is predicted by using Eq. (7) or by using Eq. (8).

3. Method for solving the governing equation

At each time step, the Laplace equation needs to be numerically solved once. One could choose any numerical method to do this, either grid-based or mesh-free. In this study the local polynomial collocation method proposed in Ref. [22] is chosen for the sake we need accurate partial derivatives of the velocity potential on the free surface. This method was developed for solving general partial 2-D differential equations. In this paper, we extend it to 3-D. The following gives a brief description of this method.

Consider the general 3-D linear second order PDE as

$$L\{\phi\} = c_1\phi + c_2\frac{\partial\phi}{\partial x} + c_3\frac{\partial\phi}{\partial y} + c_4\frac{\partial\phi}{\partial z} + c_5\frac{\partial^2\phi}{\partial x^2} + c_6\frac{\partial^2\phi}{\partial y^2} + c_7\frac{\partial^2\phi}{\partial z^2} + c_8\frac{\partial^2\phi}{\partial x\partial y} + c_9\frac{\partial^2\phi}{\partial y\partial z} + c_{10}\frac{\partial^2\phi}{\partial z\partial x} = S, \vec{x} \in \Omega \tag{9}$$

subjected to the boundary conditions

$$B\{\phi\} = q_1\phi + q_2\frac{\partial\phi}{\partial x} + q_3\frac{\partial\phi}{\partial y} + q_4\frac{\partial\phi}{\partial z} = f, \vec{x} \in \Gamma \tag{10}$$

where $L\{\}$ and $B\{\}$ are both linear operators, Ω denotes the domain, Γ denotes the boundary, and $c_1, c_2, \ldots, c_{10}, q_1, q_2, \ldots, q_4, f$, and S, are all functions of x, y, and z. When q_1 is not zero but $q_2, \ldots, q_4$ are all zero, the boundary condition is Dirichlet type; on the contrary, it is Neumann type. If $q_1, \ldots, q_4$ are all non-zero, the boundary is Robin type. One should keep in mind that Γ could be composed of several patches and at each connection of two or maybe three patches so $c_1, c_2, \ldots, c_{10}, q_1, q_2, \ldots, q_4$, and f could be multi-valued. We express the boundary condition in this general way for conciseness. Here we introduce a number n_{bc}, which indicates the number of boundary conditions to be satisfied at a specific node. It should be noted that n_{bc} is an integer greater than or equal to zero. In case of $n_{bc} = 0$, only the governing equation is to be satisfied. That means obviously indicates that $\vec{x}_j$ is inside the domain. If $n_{bc} \geq 2$, it obviously indicates that $\vec{x}_j$ rests on an edge or at a corner.

In seeking the numerical solutions, the entire domain is distributed with N nodes as needed. At each specific node $\vec{x}_j$, ϕ is