Exponential basis functions in solution of incompressible fluid problems with moving free surfaces

S.M. Zandi a, B. Boroomand a,*, S. Soghrati b

a Department of Civil Engineering, Isfahan University of Technology, Isfahan 84156-83111, Iran
b Department of Civil and Environmental Engineering, University of Illinois at Urbana-Champaign, 205 North Mathews Avenue, Urbana, IL 61801, USA

A R T I C L E   I N F O

Article history:
Received 28 December 2010
Received in revised form 7 July 2011
Accepted 12 September 2011
Available online 22 September 2011

Keywords:
Free surface
Moving boundaries
Lagrangian formulation
Trefftz method
Exponential basis functions
Meshless method
Fluid-structure interaction

A B S T R A C T

In this paper, a new simple meshless method is presented for the solution of incompressible inviscid fluid flow problems with moving boundaries. A Lagrangian formulation established on pressure, as a potential equation, is employed. In this method, the approximate solution is expressed by a linear combination of exponential basis functions (EBFs), with complex-valued exponents, satisfying the governing equation. Constant coefficients of the solution series are evaluated through point collocation on the domain boundaries via a complex discrete transformation technique. The numerical solution is performed in a time marching approach using an implicit algorithm. In each time step, the governing equation is solved at the beginning and the end of the step, with the aid of an intermediate geometry. The use of EBFs helps to find boundary velocities with high accuracy leading to a precise geometry updating. The developed Lagrangian meshless algorithm is applied to variety of linear and nonlinear benchmark problems. Non-linear sloshing fluids in rigid rectangular two-dimensional basins are particularly addressed.

© 2011 Elsevier Inc. All rights reserved.

1. Introduction

Modelling of free surface fluid flow with moving boundaries has several applications in engineering problems such as the simulation of dam-reservoir interaction, liquid containing tanks subjected to earthquake, mould filling. The continuous variation of the domain shape with time during the numerical solution of such problems is a challenging issue. The main problem is thus the determination of the free surface position in order to update the geometry by considering the available numerical data.

Two fundamental approaches are generally implemented for the numerical solution of transient problems via a grid of points, e.g. in finite element method. The first is the use of Eulerian description, which treats the grid as a fixed computational reference and allows the fluid to move through it. The second approach is the Lagrangian description, in which the computational grid is embedded in the fluid and moves with it. Over the past thirty years, majority of the fluid flow simulations have been performed based on the Eulerian description [1]. However, methods based on the Lagrangian description have recently received more attention especially for solving free surface flow problems considering interaction with structures [2–5].

An intermediate approach is also sought in which the grid points are allowed to move independently of the fluid motion. This approach is named as arbitrary Lagrangian–Eulerian (ALE) method, and may be considered as a hybrid method for fluid
flow. ALE has been widely used in conjunction with finite element method (FEM) for solving free surface flow problems [6–10].

Rigorous and unavoidable grid distortion is expected when Lagrangian methods are applied to problems involving moving boundaries. In order to avoid this problem, some researchers focus on adaptive meshing not only for re-discretization of the fluid domain according to the new geometry but also for improving the accuracy of the solution (see e.g. [11] and the references therein). The major drawback of this method, however, is its high computational cost.

Another classification of the existing approaches for addressing free surface flow problems is according to the necessity of using a grid in the computational procedure; e.g. mesh-based, meshless and particle methods. More recently, researchers show considerable tendency to develop new methods avoiding mesh generation. In particle methods for instance, fluid particles are tracked using the Lagrangian description. The idea is traceable in the method proposed by Gingold and Monaghan [12] for the study of astrophysical hydrodynamic problems, known as smooth particle hydrodynamics (SPH). This idea has been further developed and generalized for the solution of fluid mechanic problems [13–16]. On a similar basis, several computational methods using grids of points, known generally as meshless method, have so far been developed for fluid mechanic problems and sloshing phenomenon (see [17–19] for instance).

In this paper, a new simple method using Lagrangian description is presented for the solution of a class of engineering problems involving incompressible inviscid fluid flow with moving boundaries interacting with rigid structures (weak interaction). In this method, the solution is expressed as a linear combination of exponential basis functions (EBFs), with complex-valued exponents, which satisfy the governing equations. Constant coefficients of the solution series are evaluated through point collocation on the domain boundaries via a complex discrete transformation technique. A time marching algorithm is used to solve the non-linear problems. Hence, in each time step, with the use of EBFs the Laplace equation for pressure is solved at the beginning and end of the step, with the aid of an intermediate geometry. Other fluid variables such as acceleration, velocity and displacement are calculated accordingly. The geometry is then updated based on the Lagrangian formulation of motion through an implicit algorithm.

The transformation, proposed in [20–22], allows us to use a larger number of basis functions compared with the number of boundary information. The method has been applied to static and time harmonic elastic problems and described in more details in [23]. Application of the method to elasticity problems such as plates can be seen in [24,26]. Also, in [27] the authors have recently used the method for analyzing fully incompressible elastic materials and incompressible steady Stokes flow in different domain shapes. Here, the developed Lagrangian meshless algorithm is applied to variety of linear and nonlinear benchmark problems such as liquid sloshing in rectangular basins, solitary wave propagation, etc.

With the use of basis functions satisfying the Laplace equation, the proposed method falls in the category of Trefftz methods (see [28] for a good review). However, the application of Trefftz method in problems with moving boundaries has not been previously reported in the literature.

The layout of the paper is as follows. In Section 2 the model used for incompressible inviscid fluid flow is given. The solution procedure is explained in Section 3 which includes the definition of EBFs for the Laplace equation and the method of using the complex discrete transformation technique for imposition of the boundary conditions. We shall give a step-by-step summary of the procedure in the same section. In Section 4, some benchmark problems are solved and the numerical results are compared with those obtained from analytical methods and other numerical approaches to show the capability of the proposed procedure.

2. Governing Lagrangian equations

We consider an incompressible inviscid Newtonian fluid occupying a 2D bounded domain $\Omega$ with the boundaries as $\partial\Omega = \Gamma_f \cup \Gamma_S$. Here, $\Gamma_f$ represents the free surface and $\Gamma_S$ indicates the fluid-structure interface which is assumed to be slippery impermeable boundaries (Fig. 1). The governing equations of the motion are then momentum conservation, in the absence of viscosity, as

$$\rho \frac{Du}{Dt} = -\nabla p + \rho g \quad \text{in} \Omega,$$

(1)
and incompressibility condition
\[ \nabla \cdot \mathbf{u} = 0. \]  
(2)

In (1) and (2) \( \mathbf{u} \) is a vector containing the Cartesian components of the velocity field, \( \rho \) is the density and \( p \) is pressure. The vector \( \mathbf{g} = [0, -g]^T \) is used to define the source term vector which includes the gravity acceleration \( g \). Also \( D/Dt \) denotes the total or material derivative of the quantity and \( \nabla \) is the well known gradient operator.

By evaluating the divergence of (1), and considering (2), the following Laplace equation is obtained
\[ \nabla^2 p = 0 \quad \text{in} \quad \Omega. \]  
(3)

Now, instead of (1) and (2), we can consider (1) and (3) as a system of governing differential equations. To solve the problem, one may first find a solution for (3) satisfying appropriate boundary conditions for pressure and then use (1) to calculate the velocity field. The boundary conditions for (3) are either defined through specifying the pressure values, e.g. at the free surface, or by prescribing the derivatives of pressure, e.g. at fluid-structure interfaces. In the latter case (1) is reused to define the conditions from the information available for acceleration of the fluid-structure interfaces. Before specifying the boundary conditions we assume pressure as
\[ p = p_H - \rho gy, \]  
(4)

where \( y \) is the vertical coordinate. Introducing (4) in (3) results in the following equation
\[ \nabla^2 p_H = 0. \]  
(5)

Also by considering (4), Eq. (1) takes the following form
\[ \rho \frac{D\mathbf{u}}{Dt} = -\nabla p_H, \]  
(6)

which can be used in place of (1) when (5) is considered in place of (3).

2.1. Boundary conditions

In this subsection we express the fluid boundary conditions in terms of pressure. Considering \( \mathbf{n} \) as the outward vector normal to the fluid boundaries (Fig. 1), the boundary condition at the fluid-structure interface, \( \Gamma_S \), may be defined by considering
\[ \mathbf{n}^T \mathbf{u} = \mathbf{n}^T \mathbf{u}_s, \quad \text{or} \quad \mathbf{n}^T \mathbf{u} = \mathbf{n}^T \mathbf{u}_s, \]  
(7)

where \( \mathbf{u}_s \) and \( \mathbf{u}_s = \frac{D\mathbf{u}_s}{Dt} \) denotes the predefined velocity and acceleration of the solid contacting the fluid. By considering the second relation in (7) as the main condition, the first relation is automatically satisfied during the time marching process noting that the initial normal velocities of the fluid and solid are similar. The boundary condition for the problem defined in (5) is obtained by calculating the inner product of \( \mathbf{n} \) with (6) while using the second relation in (7). This leads to the following relation
\[ \frac{\partial p_H}{\partial n} = -\rho \mathbf{n}^T \mathbf{u}_s \quad \text{on} \quad \Gamma_S. \]  
(8)

The right hand side of (8) may be interpreted as the normal stress required for applying condition (6). Note that with the slippery condition at the fluid and solid interface, there is no restriction neither for the tangential velocity/acceleration component (i.e. \( \mathbf{m}^T \mathbf{u} = \mathbf{m}^T \mathbf{u}_s \) when \( \mathbf{m}^T \mathbf{n} = 0 \)) nor for the tangential stress. This means that no boundary layer effect is expected at the solid interface. This effect is in fact consistent with assuming that the fluid is inviscid.

At the free surface \( \Gamma_F \), pressure must be equal to atmospheric pressure which is considered constant not only on the surface but also throughout the domain. Therefore, one may define pressure as the difference between the actual fluid pressure and the atmospheric one and simply write the following condition
\[ p = 0 \quad \text{on} \quad \Gamma_F. \]  
(9)

By considering (4) the above equation may be rewritten as
\[ p_H = \rho gy \quad \text{on} \quad \Gamma_F, \]  
(10)

which plays the role of boundary condition at the free surface.

3. The solution procedure

In this section, first the meshless method using EBFs for spatial discretization of Laplace equation is described and then the implicit time integration and geometry updating will be explained.
3.1. Exponential basis functions (EBFs)

The pressure EBFs (Laplace equation EBFs) may be found by assuming

\[ p_i(x, y, x, \beta) = A(x, \beta)e^{\alpha x + \beta y}, \]

where \(x\) and \(y\) are the coordinates of a generic point in \(\Omega\), \(x\) and \(\beta\) are the complex valued exponents and \(A(x, \beta)\) is a constant independent of the coordinates. By inserting (11) in (5), the following equation is obtained

\[ x^2 + \beta^2 = 0. \]

From the above characteristic equation one may find \(x\) in terms of \(\beta\) or vice versa as follows

\[ x = \pm i\beta \quad \text{or} \quad \beta = \pm ix. \]

In which \(i = \sqrt{-1}\). With the relation obtained for the exponents, e.g. for \(x = \pm i\beta\), one can write

\[ p_i = \int_{\Omega} (A_1(\beta)e^{i\beta^2x + \beta y} + A_2(\beta)e^{-i\beta^2x + \beta y})d\Omega, \]

In the above relation \(\Omega\) is an appropriate area or locus on the Gaussian plane. The unknown coefficients \(A_1(\beta)\) and \(A_2(\beta)\) are to be found so that the boundary conditions (8) and (10) are satisfied. This is a very difficult task in many problems and therefore one may think of a discrete form of (14), for instance when the integral is to be calculated numerically, and simply write

\[ \hat{p}_i = \sum w_i(A_1(\beta_i)e^{i\beta_i^2x + \beta_i y} + A_2(\beta_i)e^{-i\beta_i^2x + \beta_i y}), \]

where “\(^{*}\)” is a counter for a set of quadrature points used for the integration and \(w_i\) represents the associated weight. With \(A_1(\beta_i)\) and \(A_2(\beta_i)\) still being two sets of unknowns, one may think of replacing \(wA_1(\beta_i)\) and \(wA_2(\beta_i)\) with two sets of new unknown coefficients, as \(c_1^i\) and \(c_2^i\), to be found from boundary conditions. Thus in its simpler form, Eq. (15) may be written as

\[ \hat{p}_i = \sum (c_1^i e^{i\beta_i^2x + \beta_i y} + c_2^i e^{-i\beta_i^2x + \beta_i y}). \]

In the above equations \(\hat{p}_i\) is an approximation to \(p_i\) and the summation is taken for the total number of points used on Gaussian plane for \(\beta\). Similar relation can be written when \(\beta_i = \pm ix_i\), in these series, \(x_i\) and \(\beta_i\) are complex valued exponents that control the amplitude and fluctuations of the basis functions in the domain. Considering all cases, one may write the solution series in a general form as follows

\[ \hat{p}_i = \sum_{j=1}^{N} c_j^i e^{j\beta_j x + \beta_j y}, \]

in which \(N\) is the total number of selected EBFs to achieve adequate accuracy. Note that in the above expression the EBFs obtained from calculating \(x_i\) in terms of \(\beta_i\) and vice versa are included. It should also be noted that \(x_i\) and \(\beta_i\) in (17) are related via the characteristic Eq. (12). It is worthwhile to mention that the appropriate locus or area on Gaussian plane is still one of the unknowns. Suitability of a locus or an area is determined by satisfaction of the boundary conditions. To alleviate the complexity of the problem, one may use a set of predefined points as quadrature points, on Gaussian plane, and evaluate the unknown coefficients through boundary conditions (as explained in the next section). The suitability of the selected points may be examined by re-evaluation of the boundary conditions. This may be performed in an adaptive procedure through defining suitable error norms (see Section 3.5). However, our experience in solution of many engineering problems shows that in practice there is no need for such adaptive procedure [23–27].

3.2. Imposition of boundary conditions using a discrete transformation

In this section, we shall follow the method proposed in [20–27] for imposition of the boundary conditions. To this end, by considering \(p_{hi}\) in its general form (17) and applying a collocation approach at \(M\) boundary points one may write the following relation

\[ P_b = \sum_{j=1}^{N} c_j \mathbf{V}_j. \]

In the above relation, \(P_b\) is a vector containing all point-wise boundary conditions, \(\mathbf{V}_j\) is a normalized vector containing the contribution of each exponential basis to the boundaries arranged in the same manner as \(P_b\), and \(c_j\) is proportional to the coefficient \(c_j\) in (17) considering the normalization factor used in \(\mathbf{V}_j\). For instance supposing that \(m\) boundary points, out of \(M\), are allocated to free surface \(T_F\) (Fig. 2) with prescribed pressure and \(n = M – m\) points are allocated to fluid-structure
In the above equation, \( \mathbf{P} \) is a symmetric matrix. Since the rank of \( \mathbf{G} \) might be less than \( M \), \( \mathbf{R} \) is evaluated as

\[
\mathbf{R} = \mathbf{G}^+,
\]

where \( \mathbf{G}^+ \) is pseudo inverse of \( \mathbf{G} \). Now we can evaluate \( \tilde{p}_{ih} \) by the following equation

\[
\tilde{p}_{ih} = \Re \left[ \sum_{j=1}^{N} \frac{1}{S_j} e^{s_j x_i y_j} \mathbf{V}_j^T \mathbf{R} \mathbf{P}_b \right].
\]
\[
\hat{p} = \Re \left( \sum_{j=1}^{N} \frac{1}{S_j} e^{x_j t_n} V_j^T \right) R P_B - \rho g y. \tag{31}
\]

Note that the scaling factor used in (22) and (31) may be interpreted as a pre-conditioner matrix for the matrix \( R \) (multiplied from right and left). With the simple boundary conditions as (8) and (9) there is no need for other types of pre-conditioning processes (see [23] for other forms). Also there is no need for domain decomposition techniques.

### 3.3. Selection of \( \alpha \) and \( \beta \)

The parameters \( \alpha \) and \( \beta \) play an important role in the variation of EBFs throughout the solution domain and thus have a prominent effect on the projection matrix \( R \). Since these parameters can take on complex values, we define a grid of points on Gaussian plane. The solution accuracy may differ by changing the grid. A detailed discussion on the selection of \( \alpha \) and \( \beta \) is out of the scope of this paper and the reader may refer to [23] for further information. In this reference, the authors have experimented. As shown in [23], the obtained accuracy with either of these approaches is satisfactory in all the cases studied.

In this paper we have used the heuristic strategy (see Appendix A). Note that the coefficients \( \alpha_j, \beta_j \) are chosen once at the beginning of the solution, i.e. \( t = 0 \), and are treated as constant coefficients during the time marching algorithm.

### 3.4. The time splitting and the geometry updating

Suppose that the total solution time has been divided into small increments. In the proposed algorithm here, the Laplace Eq. (5) should be solved twice at each time step. Assume that the solution has been advanced to the \( n \)th time step (\( t^n = t^{n+1} - t^n \)) and all static and kinematic variables from \( t = 0 \) to \( t = t^n \) are available. Let \( x^n \) and \( u^n = u(x^n, t^n) \) denote the current configuration and the current velocity, respectively. The vector of boundary conditions in the current configuration and time can then be evaluated using (19) in the following form

\[
P^n_B = P_B(x^n, t^n). \tag{32}
\]

Also for the contribution of the EBFs according to (22) we have

\[
V^n_j = V_j(x^n, t^n). \tag{33}
\]

The projection matrix \( R^n \) may be calculated as

\[
G^n = \sum_{j=1}^{N} V^T_j V_j, \quad R^n = (G^n)^+. \tag{34}
\]

Now \( \hat{p}_B^{n+1} = \hat{p}_B(x^{n+1}, t^{n+1}) \) can be evaluated using (30) as follows

\[
\hat{p}^n_B = \Re \left[ \sum_{j=1}^{N} \frac{1}{S_j} e^{x_j t_n} V_j^T \right] R^n P^n_B. \tag{35}
\]

Considering \( a \) as fluid acceleration vector, inserting (35) in (6) results in the following relation

\[
a^n = a(x^n, t^n) = -\frac{1}{\rho} \Re \left( \sum_{j=1}^{N} \frac{1}{S_j} \left\{ \frac{\alpha_j}{\beta_j} e^{x_j t^n} V_j^T \right\} R^n P^n_B \right). \tag{36}
\]

By calculating acceleration at all boundary points, the new geometry \( \hat{x}^{n+1} \), termed as intermediate configuration here, can be evaluated as

\[
\hat{x}^{n+1} = x^n + u^n \Delta t + \frac{1}{2} a^n \Delta t^2. \tag{37}
\]

We shall employ such a configuration for calculating the accelerations of the boundary points at the end of the time step. The coordinates given by (37) serve just as intermediate positions for the boundary points for calculating the boundary conditions to solve Laplace equation (5) at time \( t = t^{n+1} \). In view of (19) and (32), the vector of boundary conditions for the intermediate configuration can be evaluated as

\[
P_B^{n+1} = P_B(\hat{x}^{n+1}, t^{n+1}). \tag{38}
\]

Also for the vectors representing the contribution of EBFs

\[
V_j^{n+1} = V_j(\hat{x}^{n+1}, t^{n+1}). \tag{39}
\]

The projection matrix is then obtained as

\[
G^{n+1} = \sum_{j=1}^{N} V_j^{n+1} V_j^{n+1 T}, \quad R^{n+1} = (G^{n+1})^+. \tag{40}
\]
Now the acceleration vector can be evaluated in a manner analogous to (36) as

$$ a^{n+1} = -\frac{1}{\rho} \frac{1}{r} \left\{ \sum_{i=1}^{N} \frac{1}{g_{i}^{n+1}} \{ \beta_{j} \} e^{a_{x}^{0}y_{j}^{n+1}+y_{j}^{n+1}} \right\} R^{n+1-1} P_{B}^{n+1}. $$

(41)

After calculating $a^{n} = a(x^{n}, t^{n})$ and $a^{n+1} = a(x^{n+1}, t^{n+1})$ at all boundary points, assuming linear acceleration within each time step, the final configuration $x^{n+1}$ of the boundary points at the end of step can be evaluated as

$$ x^{n+1} = x^{n} + u(x^{n}, t^{n})\Delta t + \frac{1}{2} (2a(x^{n}, t^{n}) + a(x^{n+1}, t^{n+1}))\Delta t^2. $$

(42)

The velocity field at the boundary points of the final configuration can then be calculated as

$$ u^{n+1} = u(x^{n+1}, t^{n}) + \frac{1}{2} (a(x^{n+1}, t^{n}) + a(x^{n+1}, t^{n+1}))\Delta t. $$

(43)

The new configuration may be regarded as a new intermediate one and the procedure may be repeated until a unique $x^{n+1}$ is obtained, but our experience shows that excellent accuracy is achievable just by one step.

Having calculated $x^{n+1}$ and $u^{n+1}$, one may use them as the initial position and velocity, respectively, of the boundary points at the beginning of the next time step.

3.5. Error indicator and regenerating boundary points

Similar to other boundary mesh based or meshless methods, here deviation of the numerical solution from the exact one may be understood by evaluating the discrepancy of the approximated boundary values from the exact ones [23]. It must be noted that since the EBFs satisfy the elliptic Eq. (5), i.e. there is no residual in the interior parts of the domain, the residuals of the boundary conditions can be employed as an appropriate error indicator. Considering the content of Section 3.1, an error indicator may be devised by the re-evaluation of the boundary collocated values. This means that by (30) in hand, one may find approximated vector of $P_{B}$ named here as $P_{B}$ so that

$$ P_{B} = \{(\hat{p}_{0})_{1} (\hat{p}_{0})_{2} \cdots (\hat{p}_{0})_{m} | (\hat{p}_{0})_{1} (\hat{p}_{0})_{2} \cdots (\hat{p}_{0})_{m} \}^T, $$

(44)

where

$$ (\hat{p}_{0})_{k} = [\hat{p}_{0}]_{x_{k-1}y_{k}}, $$

(45)

$$ (\hat{p}_{0})_{k} = [\hat{p}_{0}]_{x_{k-1}y_{k}}. $$

(46)

By a collocation error defined as

$$ e = P_{B} - \tilde{P}_{B}, $$

(47)

one may define an error norm as an indicator for the errors in the collocated boundary values. A similar error norm may be defined for other boundary points different from those used for the collocation. Such indicators are suitable for judging the performance of the transformation technique used in Section 3.1. The smallness of the error norms indicates the suitability of the EBFs and boundary points used.

In problems with large displacements where the surfaces deform considerably, the distribution of boundary points may become irregular after several time steps and thus the accuracy of the calculated pressure may decrease. This can be checked by evaluating the errors in each time step as in (47). To alleviate this effect one may use a regularization technique for rearranging the boundary nodes. In this study, we suppose that the free surface profile between two adjacent boundary points, in Fig. 2 for instance, at the end of the step is linear and can be approximated as

$$ y = \frac{y_{i+1} - y_{i}}{x_{i+1} - x_{i}} (x - x_{i}) + y_{i}, $$

(48)

where $(x_{i}, y_{i})$ and $(x_{i+1}, y_{i+1})$ are the coordinates of the two adjacent boundary points. Now we can easily generate new regular points with uniform spacing on free surface boundary. These new boundary points should be used as $x^{n+1}$ in (43) for evaluating the initial velocities of the next step. This simple strategy does not impose remarkable computational cost and so in this research we regenerate the boundary points in each time step without evaluating the errors.

3.6. The step by step procedure of the method

To give an insight into the implementation of the method, we present the step by step procedure. After defining the problem geometry,

1. Choose a series of points on the boundary of the domain. At corners choose two close points to define distinct normal vectors.
2. Initialize the velocities of the boundary points if the problem does not start at rest.
3. Choose a grid of points on Gaussian plane. The bounds of the values are preliminarily determined by considering the oscillation of the boundary values [23] (see Appendix A).
4. For each point of the grid in step 3, for instance when $\alpha_i = \pm j \beta_i$, evaluate the set of EBFs as given in (16) and analogously proceed for when $\beta_i = \pm j \alpha_i$.
5. Evaluate the collocated values of the boundary conditions in the current time and configuration in a vector as defined in (32), that is $\mathbf{P}_B = \mathbf{P}_B(x^n, t^n)$.
6. For each EBF constructed in step 4, evaluate its contribution on the boundaries, that is $\mathbf{V}_j^n$.
7. Evaluate $\mathbf{G}^i = \sum_{j=1}^{N_F} \mathbf{V}_j^n \mathbf{V}_j^n$.
8. Evaluate $\mathbf{R}^i = (\mathbf{G}^i)^{\ast}$.
9. Evaluate the Lagrangian acceleration for each boundary points at the beginning of the step as given in (36).
10. Calculate the coordinates of the intermediate configuration $x^{n+1}$ by (37).
11. Evaluate the collocated values of the boundary conditions at time $t = t^{n+1}$, and on the intermediate configuration, in a vector as defined in (38), that is $\mathbf{P}_B^{n+1} = \mathbf{P}_B(x^{n+1}, t^{n+1})$.
12. For each EBF of step 4, evaluate its contribution on the new boundaries, that is $\mathbf{V}_j^{n+1}$.
13. Evaluate $\mathbf{G}^{n+1} = \sum_{j=1}^{N_F} \mathbf{V}_j^{n+1} \mathbf{V}_j^{n+1}$.
14. Evaluate $\mathbf{R}^{n+1} = (\mathbf{G}^{n+1})^{\ast}$.
15. Evaluate the Lagrangian acceleration for each boundary points at the end of the step as given in (41).
16. Calculate the final position of the boundary points through (42). (One may consider this final configuration as a new intermediate configuration and repeat from 11 until the difference between the last two configurations become sufficiently small.)
17. Regenerate a set of new regular boundary points according to the free surface profile in the previous step and consider them as $x^{n+1}$.
18. Calculate the velocities of the new boundary points through (43).
19. Set $x^{n+1}$ and $u^{n+1}$ as the initial configuration and velocity of each boundary point for the next time step.
20. Repeat from 5 for the next time step.

4. Numerical results

In this part we present the results of the method applied to some benchmark problems. In some examples we shall discuss on the errors at the boundaries and at a particular time. The following error norms are defined (see (20), (21), (45), (46) and (47))

$$
e_S = \sqrt{\frac{\sum_{k=1}^{N_S}|(\partial p_B)_k - (\partial \hat{p}_B)_k|^2}{N_S}}, \quad e_F = \sqrt{\frac{\sum_{k=1}^{N_F}|(p_B)_k - (\hat{p}_B)_k|^2}{N_F}}.
$$

In the above relations $N_S$ and $N_F$ denote the number of points used for error evaluation at the free and solid surfaces, respectively. While studying the error at the collocation points, we set $N_S = m$ and $N_F = n$ (see (20) and (21)) and obviously in that case we have $N_S + N_F = M$. However, one may use (49) for points other than those used for collocation to give more insight to the level of satisfaction of the boundary conditions (in that case $N_S \neq m, N_F \neq n$). Nevertheless, the norms defined in (49) are not non-dimensional and inherently reflect the magnitude of the quantities used. To present a relative dimensionless norm for the whole problem we define

$$
\eta = \left( \frac{\sum_{k=1}^{N_S}|(p_B)_k - (\hat{p}_B)_k|^2}{\sum_{k=1}^{N_S}|(p_B)_k|^2} + h^2 \frac{\sum_{k=1}^{N_F}|(\partial p_B)_k - (\partial \hat{p}_B)_k|^2}{\sum_{k=1}^{N_F}|(\partial p_B)_k|^2} \right)^{\frac{1}{2}},
$$

in which $h$ denotes a characteristic length (e.g. average height of the water in the tank) used for summing up quantities with different dimensions.

**Fig. 3.** Water container under harmonic excitation (Example 4.1).
4.1. Water container under harmonic excitation

We consider the problem of water sloshing in a rectangular container under harmonic excitation as the first numerical experiment. Sloshing effect is seen in many engineering applications such as liquid containing tanks or dams subjected to earthquake. Many numerical methods have so far been employed to solve sloshing problems, such as FEM [2], finite difference methods (FDM) [29], and boundary element methods (BEM) [30]. The main aim is to accurately calculate hydrodynamic pressure which is considered as a key-factor in designing the structure. An extended history of the sloshing problems has been given in [31].

A rectangular rigid tank is considered as the numerical example (Fig. 3). The length of the container $L$ and the still water depth $h$ are 80 cm and 10 cm, respectively. The tank width, $B$, for base shear calculation is 14.1 cm. These parameters have been chosen from reference [30]. The water density and gravity acceleration are $\rho = 1000 \text{ kg/m}^3$ and $g = 9.81 \text{ m/s}^2$, respectively. According to linear wave theorem, the first natural frequency $\omega_1$ is as follows

$$\omega_1 = \sqrt{\frac{\pi g}{L} \tanh \frac{\pi h}{L}}.$$  \hspace{1cm} (51)

Hence, the first natural frequency of this reservoir is 3.79 rad/s. For numerical modeling, 142 points are used on boundaries (38 points are on the left and right walls and 104 points are on the bottom and free surfaces). No regularization is used for the points in this problem. The number of EBFs used for this simulation is 400.

The tank is subjected to ground oscillation in the horizontal direction with the following acceleration

$$X_g(t) = A_x \omega_f^2 \cos(\omega_f t).$$  \hspace{1cm} (52)

In (52) $A_x$ and $\omega_f$ are, respectively, the displacement amplitude and the frequency of the harmonic excitation.

![Fig. 4. The results obtained for the rectangular tank (Example 4.1) subjected to harmonic excitation with forced frequency: (a) $\omega_f = 1.90 \text{ rad/s}$, (b) $\omega_f = 3.79 \text{ rad/s}$, (c) $\omega_f = 11.38 \text{ rad/s}$.](image-url)
For the evaluation of the base shear force $F_b$ we write

$$F_b = \int_{\Gamma_3} p_x d\Gamma_3,$$

in which $p_x = p_n x$. Also, by calculating the difference of the hydrostatic forces at two sides of the lateral walls, the hydrostatic base shear force is calculated by the following equation (see also [30])

$$F_b = \frac{\rho g B}{2} (h_R^2 - h_L^2).$$

In the above relation, $h_R$ and $h_L$ are the wave elevations on the right and left walls, respectively.

Here, the simulation is carried out for harmonic ground motion with the amplitude of $A_f = 0.04$ cm and frequencies of 1.9, 3.79 and 11.38 rad/s. We have used $\Delta t = 0.01$ s for $\omega_f = 1.9$ rad/s and $\Delta t = 0.005$ s for $\omega_f = 3.79$ and 11.38 rad/s.

Fig. 4 demonstrates the results of the simulation which are in excellent agreement with those given in [30] obtained by BEM. The results are including the wave height at the right lateral wall and the base shear force. Free surface profiles of the rectangular tank subjected to harmonic excitation with forced frequency $\omega_f = 3.79$ rad/s are shown in Fig. 5.

### 4.1.1. Errors in the boundary conditions

To give more insight to the solution process, here we present an error analysis for the boundary conditions. We consider the problem with $\omega_f = 1.9$ at $t = 0$ and perform several analyses with different numbers of boundary points and EBFs. For each solution we calculate the rank of $G$ and its condition number. Fig. 6 shows the variation of the rank of $G$ with respect to the number of points ($M$) and EBFs ($N$) used. As is seen, except for few cases, almost in all cases the matrix $G$ is rank-deficient (a solid line is used to show the limit of obtaining a full rank matrix). Interesting to note is that the matrix rank approaches to a

![Fig. 5. Free surface profiles of the rectangular tank (Example 4.1) subjected to harmonic excitation with forced frequency $\omega_f = 3.79$ rad/s at different time steps.](image-url)
constant value when considering $N$ as constant and increasing the number of points $M$. This shows that for a given number of EBFs one can find an optimal value for the number of boundary points. On the other hand, it can be seen that for a certain number of boundary points, the matrix rank grows while increasing the number of EBFs. To give more insight to the problem we calculate the condition number of $G$. Since the matrix is usually rank-deficient, its actual condition number is infinite. We define the following pseudo-condition number as a replacement

$$P_{\text{con}} = \frac{S_{\text{max}}}{S_{\text{min}}}, \quad S_{\text{max}} = \max_i |s_i|, \quad S_{\text{min}} = \min_i |s_i|,$$

(55)

with $s_i$, $i = 1, \ldots$, rank ($G$) representing the set of non-zero singular values of $G$ (see [32]). Fig. 7 depicts the variation of the pseudo-condition number with respect to the number of points and EBFs used for collocation. Again except for few cases the number is very large (its inverse is about the machine’s floating-point precision). This is observed for cases with more than a certain number of points. To investigate the effects, we calculate the error norms defined in (49) and (50). The results are shown in Fig. 8. The errors are evaluated for points used for collocation, i.e. $N_F + N_S = M$, and some other points in between, i.e. $(N_F + N_S) > M$ (for this latter case we use $N_F + N_S = 480$ in this example). In all cases we arrange the points at free and solid surfaces so that $N_F = (11/4)N_S$. Fig. 8a shows that for the cases in which $G$ is nearly full rank, the errors at the collocation points are relatively small but at the same time the errors at other points are not in the same order. By increasing the number of boundary points, the two types of error approach to each other. This means that, for such cases, the boundary conditions are satisfied with a rather uniform error distribution at all points (including those used for the collocation) despite the fact that the pseudo-condition number is large. This effect is consistent with the observations made in [33] in solution of problems with the method of fundamental solutions. A similar conclusion may be made from Fig. 8b depicting the variation of the relative error norm $\eta$ defined in (50). The figure inherently shows the magnitude of the errors compared with the values of the main quantities.

4.2. Standing wave in a rectangular tank

Water oscillation with an initial free surface profile in a rectangular tank has been simulated in many references by different numerical methods [2,4,8,14,16]. Consider a rectangular rigid tank with a cosine wave at the free surface as the initial condition (Fig. 9).
The initial free surface profile is given as
\[ g_0(x) = A \cos(k(x + \lambda/2))/C_1, \]
where \( g_0(x) \) is the initial free surface displacement, \( A \) and \( k \) are the wave amplitude and the wavelength, respectively. Also, \( k = 2\pi/\lambda \) is the wave number. An analytical solution has been given in [34] for the wave height at the center of the tank considering nonlinear effects. According to the solution, the wave elevation is the summation of a linear term as
\[ g_1(t) = A \cos(\omega_2 t), \]
and a second order term as
\[ g_2(t) = \frac{1}{8g} \left\{ 2(A\omega_2)^2 \cos(2\omega_2 t) + \frac{A^2}{\omega_2^2} \left[ k_2^2 g^2 + \omega_2^4 - \left( k_2^2 g^2 + 3\omega_2^4 \right) \cos(\omega_2 t) \right] \right\}, \]
where
\[ k_2 = n\pi/\lambda, \]
\[ \omega_n = [k_n g \tanh(k_n h)]^{1/2}. \]

In this simulation we have used \( \rho = 1000 \) kg/m\(^3\) and \( g = 9.81 \) m/s\(^2\). The length of the reservoir is \( L = \lambda = 2 \) m and the depth of water is \( h = 0.5 \) m in the equilibrium state. The wave amplitude is considered as \( A = 0.1h \). For modelling this problem, we have used 306 boundary points and 400 EBFs. The time increment \( \Delta t \) is chosen as 0.01 s. In Fig. 10, wave height at the center of the tank is depicted considering both the linear theory \( \eta = \eta_1 \) and the second order theory \( \eta = \eta_1 + \eta_2 \). The results show the accuracy and capability of the method in simulating nonlinear effects. Hydrodynamic pressure, in Fig. 11, is evaluated as
\[ p_{\text{hydrodynamic}} = p - \rho g h_w, \]
in which \( h_w \) is the water column height above the point under consideration in the fluid domain. As is seen in this Figure, the details of the pressure distribution are fully captured.

4.2.1. Errors in the boundary conditions
Here again we present an error analysis for the boundary conditions. Note that in contrast with the previous example, there is no excitation through the solid surfaces of the tank. We consider the problem at \( t = 0 \) and as before perform several analyses.
with different numbers of boundary points and EBFs. Fig. 12 shows the variation of the rank of $G$ with respect to the number of points ($M$) and EBFs ($N$) used. Here again, except for few cases the matrix $G$ is rank-deficient. Similar effects observed in the previous example are again seen here, e.g. the matrix rank approaches to a constant value when considering $N$ as constant and increasing the number of points $M$. Fig. 13 depicts the variation of the pseudo-condition number with respect to the number
Fig. 12. Variation of the rank of $G$ with respect to the number of points and EBFs used for collocation (Example 4.2).

Fig. 13. Variation of the pseudo condition number defined in (55) with respect to the number of points and EBFs used for collocation (Example 4.2).

Fig. 14. Variation of the boundary errors (Example 4.2) with respect to the number of points and EBFs used for collocation; (a) variation of $e_F$, (b) variation of relative error $\eta$.

Fig. 15. Rectangular basin with a half cosine wave (Example 4.3).
Fig. 16. Surface wave elevation for 0.1 m amplitude.

Fig. 17. Surface wave elevation for 0.5 m amplitude.

Fig. 18. Surface wave elevation for 2.0 m amplitude.
of points and EBFs used for collocation. Again except for few cases the number is very large. To further investigate the effects, we calculate the error norms defined in (49) and (50). The results are shown in Fig. 14. For the case in which the errors are calculated at points different from the collocation ones we use $N_F + N_S = 750$. In all cases we arrange the points at free and solid surfaces so that $N_F = 4N_S$. Fig. 14a and b show that for the cases in which $G$ is nearly full rank, the errors at the collocation points are relatively

Fig. 19. Pressure contours and velocity vectors of large amplitude oscillation in a rectangular basin, $A = 2.0$ m (Example 4.3).
small but at the same time some erroneous results may be expected at other points. Here again by increasing the number of boundary points, the latter type of error decreases and approaches to the former type, leading to a rather uniform error distribution at all points. Comparison of the results in Fig. 14b with those in Fig. 8b shows that the relative errors in this example are of less magnitude. This effect can be due to the fact that in the previous example the tank was excited through the solid walls producing non-zero Neumann conditions as defined in (8) while in this example the problem is excited by free surface producing non-zero Dirichlet boundary conditions as defined in (10).

4.3. Large amplitude water oscillation in a rectangular reservoir

As the next numerical experiment, we consider a rectangular basin with a half cosine wave at free surface (Fig. 15). The initial free surface profile is given by

$$\eta_0(x) = A \cos \left( \frac{\pi(x + L/2)}{L} \right).$$  \hfill (62)

The definitions of parameters in (62) are the same as those in the previous example. According to shallow water theory [35], the fluid velocity field for small amplitude oscillations is as follows

$$u(x, y, t) = \frac{A \sqrt{gh}}{h} \sin \left( \frac{\pi(x + L/2)}{L} \right) \sin \left( \frac{\pi \sqrt{gh} t}{L} \right),$$  \hfill (63)

$$v(x, y, t) = -\frac{A \sqrt{gh} \pi (y + h/2)}{hL} \cos \left( \frac{\pi(x + L/2)}{L} \right) \sin \left( \frac{\pi \sqrt{gh} t}{L} \right),$$

where $u$ and $v$ are the velocity components and $h$ is the still water depth. So the wave elevation at the left and right walls and the center of the basin may be evaluated by

$$\eta_{\text{left}}(t) = A + \int_0^t v(-L/2, h/2, t) dt,$$

$$\eta_{\text{center}}(t) = \int_0^t v(0, h/2, t) dt,$$

$$\eta_{\text{right}}(t) = -A + \int_0^t v(L/2, h/2, t) dt.$$  \hfill (64)

![Fig. 20. Solitary wave propagation: problem definition.](image)

![Fig. 21. The maximum run-up height at the right wall versus initial wave height (Example 4.4).](image)
This problem has been solved in [8] using arbitrary Lagrangian–Eulerian finite element and velocity-vorticity formulation of fluid flow equations. In the reference, 4137 elements have been used and the time increment has been chosen as $\Delta t = 0.02$ s to simulate the problem for $L = 98$ m and $h = 5$ m.

Here, we have used 294 boundary points and 400 EBFs for modelling. The time increment $\Delta t$ and all other parameters are the same as those used in [8]. This simulation is carried out for different wave amplitudes. In Figs. 16 and 17 the wave elevation for the wave amplitudes $A = 0.1$ m and $A = 0.5$ m are shown. The results are in excellent agreement with those given in [8]. By increasing the initial free surface wave amplitude, the difference between linear analytical solution and nonlinear numerical analysis significantly grows.

![Diagram](image1)

**Fig. 22.** Solitary wave propagation (Example 4.4) with $H/d = 0.2$ at time $t = 8.0$ (sec): (a) boundary points; (b) $u$ (m/s); (c) $v$ (m/s); and (d) $p$ (Pa).

![Diagram](image2)

**Fig. 23.** Time history of maximum wave height for $H/d = 0.2$ and $d = 10$ (m) (Example 4.4).
In Fig. 18 the time history of the wave elevation with amplitude of 2.0 m is presented. For such a large amplitude initial wave, the nonlinear effects appear clearly and the wave elevation arises to about 3.5 m. Again comparison between the results and those in [8] shows the capability of the presented method in simulation of large amplitude waves. Fig. 19

**Fig. 24.** Vertical velocity distribution, Example 4.4, for every 2.5 (sec) when $H/d = 0.3$. 
demonstrates the pressure contours and velocity vectors at different time steps; it also shows the free surface configuration at different time steps.

4.4. Solitary wave propagation

Solitary wave propagation is a benchmark problem that is widely used for evaluating the capability of numerical methods in simulating free surface flows. Fig. 20 shows the problem geometry and the definition of the parameters. A wave moves from the center of the reservoir and, after colliding with the right wall, travels back to its initial position. For such a problem experimental [36], analytical [37] and numerical [6,8–10,38,39] results are available in the literature.

Total length of the basin is \( L = 16d \) and \( d = 1. \) The gravity acceleration is \( g = 9.81 \) and the density of water is \( \rho = 1000. \) The initial conditions are calculated according to the solution presented by Laitone [40] for an infinite domain. Based on Laitone’s work, the total wave height, velocity components and pressure for a solitary wave propagating in an infinite domain are given as

\[
h = d + H \text{sech}^2 \left( \sqrt{\frac{3H}{4d}} (x - ct) \right),
\]

\[
u = \sqrt{3gd} + \frac{H}{d} \text{sech}^2 \left( \sqrt{\frac{3H}{4d}} (x - ct) \right),
\]

\[
v = \sqrt{3gd} \left( \frac{H}{d} \right)^{3/2} \left( \text{sech}^2 \left( \sqrt{\frac{3H}{4d}} (x - ct) \right) \right) \tanh \left( \sqrt{\frac{3H}{4d}} (x - ct) \right),
\]

and

\[
p = \rho g (h - y).
\]

in which \( c \) satisfies the following relation

\[
c = \sqrt{gd} \left( 1 + \frac{1}{2} \frac{H}{d} - \frac{3}{20} \left( \frac{H}{d} \right)^2 + O \left( \frac{H}{d} \right)^3 \right).
\]

The initial conditions including velocity components and free surface position are obtained by substituting \( t = 0 \) in Laitone’s solution. The simulation is carried out for \( H/d = 0.1, 0.2, 0.3, 0.4 \) using 221 boundary points and 400 EBFs. It is worthwhile to mention that for a FEM simulation 16000 triangular elements with 4221 nodes are used in [8] and 3838 triangular elements

![Fig. 25. Initial velocity distributions for interaction of two opposite solitary waves, Example 4.5, for \( H_L = 0.3d \) and \( H_R = 0.2d \): (a) \( u \) (m/s); (b) \( v \) (m/s).](image)

<table>
<thead>
<tr>
<th>Example</th>
<th>( H_L = 0.3d, H_R = 0.3d )</th>
<th>( H_L = 0.3d, H_R = 0.2d )</th>
<th>( H_L = 0.2d, H_R = 0.2d )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eq. (71)</td>
<td>0.66525</td>
<td>0.54125</td>
<td>0.42600</td>
</tr>
<tr>
<td>Present method</td>
<td>0.64659</td>
<td>0.52902</td>
<td>0.42170</td>
</tr>
</tbody>
</table>
with 2092 nodes are used in [10] for this problem. Time increment is set to $\Delta t = 0.01$ s. Byatt-Smith [37] presented the following analytical relation for maximum run-up height at the right wall:

**Fig. 26.** Velocity vectors for collision between two opposite solitary waves with $H_L = 0.3d$ and $H_R = 0.2d$. 
\[
\frac{R_{\text{max}}}{d} = 2 \left( \frac{H}{d} \right) + \frac{3}{2} \left( \frac{H}{d} \right)^2 + O \left( \frac{H}{d} \right)^3.
\]  

(70)

The maximum run-up height at the right wall is shown in Fig. 21. The numerical results are in close agreement with the analytical one proposed by Byatt-Smith [37] and in all cases the differences are less than 3% with the second order theory. In Fig. 22, the boundary points, velocity, and pressure distribution are presented. Fig. 23 illustrates the time history of the maximum wave height for the case of \( H/d = 0.2 \) which is in excellent agreement with the results obtained in [9]. In this case, we use \( d = 10 \) for comparison. Fig. 24 depicts the distribution of the vertical velocity for every 2.5 s when \( H/d = 0.3 \).

4.5. Interaction of two opposite solitary waves

In this section, the collision of two opposite solitary waves with different amplitudes is simulated. The problem has been analytically and experimentally discussed in [36,41,42]. Also, it has been numerically simulated in [7] using ALE approach. In [42] the maximum run-up height during the collision has been estimated by the following relation

\[
H_{\text{max}} = H_L + H_R + \frac{1}{2} H_L H_R + \frac{3}{8} H_L H_R (H_L + H_R),
\]

(71)
in which \( H_L \) and \( H_R \) are the initial wave amplitudes on the left and right-hand sides, respectively.

Here we consider a rectangular domain with length \( L = 32d \) while \( d = 1 \). Initial geometry and velocities are calculated using Laitone’s solution. These initial conditions are demonstrated in Fig. 25 for when the left-hand side amplitude is \( H_L = 0.3d \) and the right-hand side amplitude is \( H_R = 0.2d \). The simulation is carried out using present method with 240 boundary points and 480 EBFs. The results of maximum run-up height for different amplitudes are tabulated in Table 1. Fig. 26 shows velocity vectors at different time steps.

5. Conclusions

In this paper we presented a method in which a series of EBFs are used to solve free surface flow through a Lagrangian description. The EBFs are found by defining characteristic equations from the governing differential equations in incompressible inviscid fluid flow problems. The boundary conditions are imposed through a collocation approach and thus the method can be categorized in meshless types. In the presented method, the number of EBFs does not need to be equal to that of the boundary information. A transformation technique is employed for the evaluation of the unknown coefficients. The use of EBFs helps to find boundary velocities with high accuracy leading to a precise geometry updating. The developed Lagrangian meshless algorithm is applied to variety of linear and nonlinear benchmark problems. Excellent agreement is seen between the obtained results and those available in the literature. The results show the capability of the method for simulating free surface flows.

The numerical method used for modeling of sloshing effect can easily be extended to investigate three-dimensional free surface flow problems. Our parallel researches show that one can solve Laplace equation in three-dimension with high accuracy. For solving such problems it is enough to replace the two dimensional EBFs with the three-dimensional ones in the explained procedure after defining the geometry, and its normal vectors to the surfaces, in three-dimensional space. Also the solution of containers with viscous fluids, in which boundary layer effect becomes prominent, is another subject currently focused on by the authors.

Appendix A

The heuristic strategy for choosing \( \alpha \) and \( \beta \) proposed in [23] is as follows

\[
\alpha = \pm \frac{m}{\bar{L}} \left( \frac{k}{\bar{N}} + 2i \right), \quad \bar{m} = 1, \ldots, \bar{M}, \quad \bar{k} = 1, \ldots, \bar{N}
\]

(A-1)

for \( \beta = \pm i \bar{\beta} \). In a similar manner, when \( \alpha = \pm i \bar{\alpha} \) we select

\[
\beta = \pm \frac{m}{\bar{L}} \left( \frac{k}{\bar{N}} + 2i \right), \quad \bar{m} = 1, \ldots, \bar{M}, \quad \bar{k} = 1, \ldots, \bar{N}
\]

(A-2)

In the above relations, we choose \((\bar{M}, \bar{N}) \in \mathbb{N}^2; \bar{\gamma} \in \mathbb{R} \) and \( \bar{L} \) is a characteristic length. The following bounds are found to be appropriate for many cases

\[
5.6 \leq \bar{\gamma} \leq 7.2 \text{ (say } \bar{\gamma} = 2\pi), \quad \bar{L} = 1.6 \max(L_x, L_y), \quad \bar{M}_{\text{min}} = 4, \quad \bar{N}_{\text{min}} = 2, \quad \bar{N}_{\text{max}} = 8
\]

(A-3)

where \( L_x \) and \( L_y \) are the dimensions of the rectangle that circumscribes the domain. According to this pattern, the number of constructed EBFs is \( 8\bar{M} \times \bar{N} \). For modelling of the problems presented in this paper we have used \( \bar{\gamma} = 2\pi, \bar{N} = 5 \) and \( \bar{M} = 10 - 12 \). It should be noted that we calculate \( L_x \) and \( L_y \) at \( t = 0 \).