Simple computer code for inviscid sloshing in a 2D rectangular basin with spectral-finite difference method

Danial Amini Baghbadorani[1]

[1]Independent Researcher, 3rd Floor, No. 34, Golestan Dead End., Dehghan Alley, Bazarche St., Shahrestan Bridge, Moshtagh 2 St., P.O. Box 81581-36647, Esfahan, Iran. Email: danial.amini@gmail.com

Submitted to: Journal of Ocean Engineering and Marine Energy
Article type: Short Communication
Date: 1 Nov. 2018

Abstract

Keywords: Useless

Introduction: Standing waves in a stationary 2D rectangular basin for an homogeneous, inviscid and irrotational fluid is an old and well-documented problem which occurs in water and fuel tankers, lakes and reservoirs. Despite the numerous analytical and numerical solutions available, a simple computer program employing the spectral-finite difference method is needed, particularly for classroom use in university water wave mechanics courses and performing simple calculations. Here is a schematic in Fig. 1. The variables are x and y= horizontal and vertical coordinates; u and w = horizontal and vertical velocity components; L = basin length, h = initial water height, H = wave height, η = wave elevation and ϕ = velocity potential; t = time, ρ = fluid density and g = 9.81 m/s^2 = gravity's acceleration.

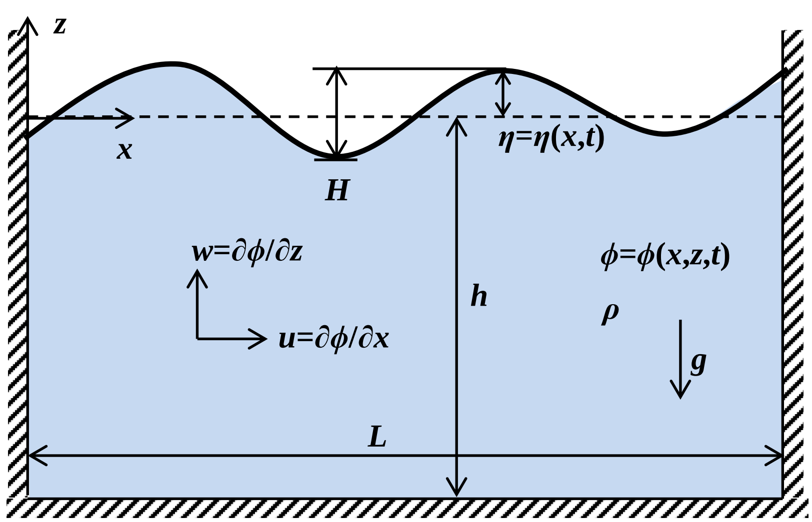

Fig. 1. Schematic of 2D rectangular basin with standing waves

Existing equations of … a

Governing equations

For standing waves formed by a homogeneous, inviscid and irrotational (ideal) fluid in a 2D rectangular basin, the governing equations along with solid wall and free surface boundary conditions are the following:

$$\begin{aligned}
&\frac{\partial^2\phi}{\partial x^2}+\frac{\partial^2\phi}{\partial z^2}=0 \quad 0\le x\le L,-h\le z\le\eta;\quad \phi=\phi(x,z,t);\quad \eta=\eta(x,t);\\
&\frac{\partial\phi}{\partial x}=0 \text{ at } x=0,x=L;\quad \frac{\partial\phi}{\partial z}=0 \text{ at } z=-h;\\
&\frac{\partial\phi}{\partial z}=\frac{\partial\eta}{\partial t}+\frac{\partial\eta}{\partial x}\frac{\partial\phi}{\partial x},\quad \frac{\partial\phi}{\partial t}+\frac{1}{2}\left[\left(\frac{\partial\phi}{\partial x}\right)^2+\left(\frac{\partial\phi}{\partial z}\right)^2\right]+g\eta=0 \text{ at } z=\eta
\end{aligned} \tag{1}$$

The Laplace equation is the continuity equation. The last two equations are kinematic and dynamic (Bernoulli) boundary conditions. Following the paper of Sobey (2009), the equations are rewritten in dimensionless form. Defining $k=\pi/L$, we have $k\doteq[1/\mathrm{L}]$, $\sqrt{gk}\doteq[1/\mathrm{T}]$. As a result, the following transforms are needed

$$\begin{aligned}
&kz=z^*,\quad k\eta=\eta^*,kx=x^*,kL=\pi,\quad kh=q,t\sqrt{gk}=t^*,\Delta t\sqrt{gk}=\Delta t^*,g^*=1,\\
&\phi=\phi^*\frac{\sqrt{gk}}{k^2},\ \frac{\partial\phi}{\partial z}=\sqrt{\frac{g}{k}}\frac{\partial\phi^*}{\partial z^*},\quad \frac{\partial\phi}{\partial x}=\sqrt{\frac{g}{k}}\frac{\partial\phi^*}{\partial x^*},\quad \frac{\partial\phi}{\partial t}=\frac{g}{k}\frac{\partial\phi^*}{\partial t^*}\\
&\frac{\partial\eta}{\partial x}=\frac{\partial\eta^*}{\partial x^*},\frac{\partial\eta}{\partial z}=\frac{\partial\eta^*}{\partial z^*},\frac{\partial\eta}{\partial t}=\sqrt{\frac{g}{k}}\frac{\partial\eta^*}{\partial t^*}
\end{aligned} \tag{2}$$

And the governing equations become

$$\begin{aligned}
&\frac{\partial^2\phi^*}{\partial x^{*2}}+\frac{\partial^2\phi^*}{\partial z^{*2}}=0 \text{ at } 0\le x^*\le\pi,-q\le z^*\le\eta^*;\phi^*=\phi^*\left(x^*,z^*,t^*\right);\eta^*=\eta^*\left(x^*,t^*\right);\\
&\frac{\partial\phi^*}{\partial x^*}=0 \text{ at } x^*=0,\pi;\ \frac{\partial\phi^*}{\partial z^*}=0 \text{ at } z^*=-q;\frac{\partial\phi^*}{\partial z^*}=\frac{\partial\eta^*}{\partial t^*}+\frac{\partial\eta^*}{\partial x^*}\frac{\partial\phi^*}{\partial x^*} \text{ at } z^*=\eta^*;\\
&\frac{\partial\phi^*}{\partial t^*}+\frac{1}{2}\left[\left(\frac{\partial\phi^*}{\partial x^*}\right)^2+\left(\frac{\partial\phi^*}{\partial z^*}\right)^2\right]+\eta^*=0 \text{ at } z^*=\hat{\eta}^*
\end{aligned} \tag{3}$$

The analytical solution to the Laplace equation with bottom and side boundary conditions is

$$\phi^{*n}=\sum_{j=0}^{J}a_j^n\cos jx^*\left[\cosh jz^*+\tanh jq\sinh jz^*\right],\quad \eta^{*n}=\sum_{j=1}^{J}b_j^n\cos jx^* \tag{4}$$

Where a_j and b_j= coefficients of potential and free surface functions; j = counter for Fourier series terms; J = number of Fourier terms; n = time step, and η^* is assumed to be a cosine series. The series of η doesn't include j=0 because mean surface water is z=0. Differentiating on x and z is done on the trigonometric functions to yield:

$$\begin{aligned}
&u^{*n}=\frac{\partial\phi^*}{\partial x^*}=-\sum_{j=0}^{J}a_j^n\sin jx^*\left[\cosh jz^*+\tanh jq\sinh jz^*\right]\\
&w^{*n}=\frac{\partial\phi^*}{\partial z^*}=\sum_{j=0}^{J}a_j^n\cos jx^*\left[\sinh jz^*+\tanh jq\cosh jz^*\right],\eta_x^{*n}=-\sum_{j=1}^{J}b_j^n\sin jx^*,
\end{aligned} \tag{5}$$

differentiation with respect to time is done by finite difference

$$\eta^{*n}_{t^*}=\frac{\partial \eta^{*n}}{\partial t^*}=\sum_{j=1}^{J}\frac{b_j^{n+1}-b_j^n}{\Delta t^*}\cos jx^*,$$
$$\phi^{*n}_{t^*}=\frac{\partial \phi^{*n}}{\partial t^*}=\sum_{j=0}^{J}\frac{a_j^{n+1}-a_j^n}{\Delta t^*}\cos jx^*\left[\cosh jz^*+\tanh jq\sinh jz^*\right]=\sum_{j=0}^{J}\frac{a_j^{n+1}-a_j^n}{\Delta t^*}F_j\left(z^*\right) \tag{6}$$

Where F is the function appearing in ϕ function. In the free surface boundary conditions, the terms not containing time derivatives are grouped together for convenience

$$FK^n\left(x^*\right)=w^*\left(\left\{a^n\right\},\eta^{*n},x^*\right)-\eta^*_{x^*}\left(\left\{b^n\right\},x^*\right)u^*\left(\left\{a^n\right\},\eta^{*n},x^*\right)$$
$$FD^n\left(x^*\right)=-\frac{1}{2}\left[u^*\left(\left\{a^n\right\},\eta^{*n},x^*\right)\right]^2-\frac{1}{2}\left[w^*\left(\left\{a^n\right\},\eta^{*n},x^*\right)\right]^2-\eta^*\left(\left\{b^n\right\},x^*\right) \tag{7}$$

Wherein FK and FD are right hand side of kinematic and dynamic boundary conditions calculated at $z^*=\eta^*$ and $\{\cdot\}$ refers to vector of variables. This form of writing is used because calculation of η^* and $\eta^*_{x^*}$ needs knowledge about $\{b^n\}$ and x^*, and calculation of ϕ, u and w at $z^*=\eta^*$ needs $\{a_n\}$, x^* and η^* at that point. Consequently,

$$\sum_{j=1}^{J}\frac{b_j^{n+1}-b_j^n}{\Delta t^*}\cos jx^*=FK^n\left(x^*\right),\sum_{j=0}^{J}\frac{a_j^{n+1}-a_j^n}{\Delta t^*}F_j\left(x^*,\eta^{*n}\right)=FD^n\left(x^*\right) \tag{8}$$

Notice that for the dynamic boundary condition the following expression is wrong because time differentiation is performed before substituting $z^*=\eta^*$.

$$\sum_{j=0}^{J}\frac{a_j^{n+1}F_j\left(x^*,\eta^{*n+1}\right)-a_j^nF_j\left(x^*,\eta^{*n}\right)}{\Delta t^*}=FD^n\left(x^*\right) \tag{9}$$

Proceeding with Eq. (8), coefficients $\{b^{n+1}\}$ are directly found from orthogonality of Fourier series:

$$b_j^{n+1}=b_j^n+RL_j^n,RL_j^n=\frac{2\Delta t^*}{\pi}\int_0^{\pi}\cos jx^*FK^n\left(x^*\right)dx^*,1\le j\le J \tag{10}$$

This is not possible for dynamic boundary condition because it is not orthogonal (even though $\cosh jz^* + \tanh jq \sinh jz^* \approx 1$. As a result, a system of equations should be constructed

$$\frac{2}{\pi}\sum_{j=0}^{J}\int_0^{\pi}a_j^{n+1}F_j\left(x^*,\eta^{*n}\right)\cos kx^*dx^*=\frac{2}{\pi}\int_0^{\pi}\left[\phi\left(\left\{a_j^n\right\},x^*\right)+\Delta t^*FD^n\left(x^*\right)\right]\cos kx^*dx^* \tag{11}$$

For $0\le k\le J$. Also, $\phi\left(\left\{a_j^n\right\},x^*,\eta^{*n}\right)=\sum_{j=0}^{J}a_j^{n+1}F_j\left(x^*,\eta^{*n}\right)$. This cast be rewritten as

$$\left[C_{kj}^n\right]_{J+1\times J+1}\left\{a_j^{n+1}\right\}_{J+1\times 1}=\left\{RD_k^n\right\}_{J+1\times 1} \tag{12}$$

Solving this equation gives the value of $\{a_j\}$ vector. It is not necessary to including k=0 in this equation because k=0 is not coupled with k≥1 equations, but this is a slight difference in calculations. Eqs. (10) and (12) are the forward difference formula. Corrector-predictor gives better results which can be expressed by:

$$\text{Corrector: } b_j^p=b_j^n+RL_j^n,\quad \left[C_{kj}^n\right]\left\{a_j^p\right\}=\left\{RD_k^n\right\}$$
$$\text{Predictor: } b_j^{n+1}=b_j^n+\frac{RL_j^n+RL_j^p}{2},\quad \left[C_{kj}^n\right]\left\{a_j^{n+1}\right\}=\frac{\left\{RD_k^n\right\}+\left\{RD_k^p\right\}}{2} \tag{13}$$

The calculations mentioned here were written as a small code in Julia language and available in https://github.com/DanialAmini/Sloshing-Finite-Difference. Julia is a free numerical programming language created by scientists in MIT.

Solution method

For solving the equations, finite difference method in time is used as performed by Forbes (2010). The governing equations are written as

$\eta_t =$	(14)

Results

Highlights:

fdaf

Cover letter:

JKjkls s

Suggested Reviewers:

First Name:
Last Name:
Academic degree:
Position:
Department:
Institution:
Email address:
Reason:

Forbes, L. K. J. J. o. E. M. (2010). "Sloshing of an ideal fluid in a horizontally forced rectangular tank." 66(4), 395-412.

Sobey, R. J. P. o. t. I. o. C. E., Issue EM4 (2009). "Analytical solutions for steep standing waves. Engineering and Computational Mechanics 162." 185, 197.