FreeCAD-Ship.

Sea waves transport using Boundary Elements Method

JL Cercos-Pita <jlcercos@gmail.com>

April 1, 2013

Contents

Chapter 1

Introduction

1.1 Objective

The objective of this document is introduce briefly how the waves of the seakeeping simulator are propagated.

In the seakeeping simulator Boundary Elements Method (BEM) will be used, that is detailed described in several books, like Ang [1]. Vinayan and Kinnas [2] gives a detailed description of the propagation of waves in a 2D case, and is a good starting point.

We will start briefly describing the governing equations in order to can start working with the 2D problem. First the incident waves over our computational domain will be described, introducing also the potential, discussing then the BEM applied to this case. As we will see the Laplace problem in the 2D case will not be really useful for us.

After that we can start working in the 3D case, that is our real objective. The incident waves will be rewritten, and the Laplace problem and the BEM application purposed again.

Chapter 2

Governing equations

Assuming no viscous fluid (that allows to transform Navier-Stokes equations into Euler ones), and imposing an initial condition such that[1]:

$$\nabla \times \boldsymbol{u} = 0 \tag{2.1}$$

The fluid velocity derives from a scalar function potential ϕ

$$\nabla\phi = \boldsymbol{u} \tag{2.2}$$

Then the Navier-Stokes equations can be rewriten as a Laplacian problem and Bernoulli equation:

$$\triangle\phi = 0 \tag{2.3}$$
$$p = -\rho\left(|\boldsymbol{u}|^2 + g\boldsymbol{z}\right) \tag{2.4}$$

And in order to solve the Laplace problem 2.3 we will use the BEM as described by Ang [1].

[1] With no viscous fluid this condition is preserved along the time

Chapter 3

Waves propagations in 2D plane

3.1 General

In this chapter the 2D case will be discussed, looking for a method to solve the BEM using only the information about the free surface. As we will see is not possible to do it in the 2D case, and may move to 3D case.

3.2 Incident waves

First for all we need to describe the incident waves, that are the waves out of our computational domain.

$$\begin{aligned} z(x,t) &= \sum_{j=1}^{n_{waves}} a_j \sin\left(k_j x - \omega_j t + \delta_j\right) \\ v_z(x,t) &= \sum_{j=1}^{n_{waves}} - a_j \omega_j \cos\left(k_j x - \omega_j t + \delta_j\right) \end{aligned} \tag{3.1}$$

The phase velocity for the waves in the most general case can be written as:

$$c_j = \sqrt{\frac{g\lambda}{2\pi} \tanh\left(\frac{2\pi d}{\lambda}\right)} \tag{3.2}$$

But if the water depth is such that $d > \frac{\lambda}{2}$ we can rewrite the phase velocity 3.2 as

$$c_j^{deep} = \sqrt{\frac{g\lambda}{2\pi}} \tag{3.3}$$

By definition the phase velocity can be also written as $c_j = \omega_j / k_j$, so we can relate the wave length with the period

$$k_j = \frac{\omega_j^2}{g} \tag{3.4}$$

Regarding the velocity potential, let us define it as

$$\phi(x,z,t) = \sum_{j=1}^{n_{waves}} -\frac{a_j \omega_j}{k_j} \cos\left(k_j x - \omega_j t + \delta_j\right) \exp\left(k_j z\right) \tag{3.5}$$

such that

$$\begin{aligned} \left.\frac{\partial \phi(x,z,t)}{\partial z}\right|_{z=0} &= v_z(x,t) \\ \triangle \phi(x,z,t) &= 0 \end{aligned}$$

This potential is valid for deep water waves if the exponent $k_j z$ is small enough. z can take values of the order of $O(a_j)$, so this method is valid for waves which the length is such that

$$\lambda_j \gg 2\pi a_j$$

3.3 BEM applied to Laplace 2D problem

We have defined the sea waves outside from our computational domain Ω, where the waves can be perturbed by floating objects. In the figure 3.1 a schematic view of the computational domain is shown.

The BEM is based in the reciprocal relation application

$$\lambda(X,Z)\phi(X,Z;t) = \int_{\partial\Omega} \phi(x,z;t)\frac{\partial G(x,z,X,Z)}{\partial \boldsymbol{n}(x,z)} - G(x,z,X,Z)\frac{\partial \phi(x,z;t)}{\partial \boldsymbol{n}(x,z)} \mathrm{d}s(x,z) \tag{3.6}$$

Where

$$\lambda(X,Z) = \begin{cases} 1 & \text{if} \quad (X,Z) \in \Omega \\ \frac{1}{2} & \text{if} \quad (X,Z) \in \partial\Omega \\ 0 & \text{if} \quad (X,Z) \notin \Omega \end{cases}$$

$G(x,z,X,Z)$ is the Green's function (a particular solution of the Laplace equation), and the derivative respect to normal denotes the gradient of the function projected over the normal.

$$\frac{\partial f(x,z)}{\partial \boldsymbol{n}(x,z)} = \nabla f(x,z) \cdot \boldsymbol{n}(x,z)$$

Hereinafter let we define the function $H(x,z,X,Z)$ as

$$H(x,z,X,Z) = \frac{\partial G(x,z,X,Z)}{\partial \boldsymbol{n}(x,z)}$$

Therefore BEM allows to, knowing along the contour the potential or the gradient, not both, compute the another one. The parts of the contour where we know both potential and the gradient will enter in the method as part of the independent term in the linear system of equations.

Since we know the potential out from the domain Ω, we can know the potential and the gradient along the inlet $\partial\Omega_{Inlet}$ and outlet $\partial\Omega_{Outlet}$.

Regarding the bottom, we only can assert that the gradient along the normal, that is the vertical velocity of the fluid, is null.

Therefore, the inlet and outlet have relatively good properties because we know all the data, so will not be additional work into the linear system matrix that must be inverted, but the bottom don't have this desirable property.

Nevertheless, since the geometry of the inlet and outlet is different from the free surface, and the bottom must be explicitly considered, all the contour must be discretized with an undesirable computational cost associated.

So we are interested to know if we can replace our computational domain Ω for other where only the free surface contour is involved, moving the Inlet, the Outlet, and the bottom infinity far. We could change our computational domain if the Green's function, and their gradient, goes to zero as we go far enough. In 2D we can found a Green's function for the Laplace problem such that

$$G(x,z,X,Z) = \frac{1}{4\pi}\log\left((x-X)^2+(z-Z)^2\right) \tag{3.7}$$

$$H(x,z,X,Z) = \frac{1}{2\pi}\frac{(x-X,z-Z)}{\left((x-X)^2+(z-Z)^2\right)} \tag{3.8}$$

Whose limits when the radius goes to infinite can be found

$$\lim_{(x-X)^2+(z-Z)^2\to\infty} G(x,z,X,Z) = \infty \tag{3.9}$$

$$\lim_{(x-X)^2+(z-Z)^2\to\infty} H(x,z,X,Z) = 0 \tag{3.10}$$

So in the 2D Laplace problem, if we try to send the Inlet, the Outlet, or the bottom to the infinite we can't use BEM because the Green's function is not well defined, diverging with the distance.

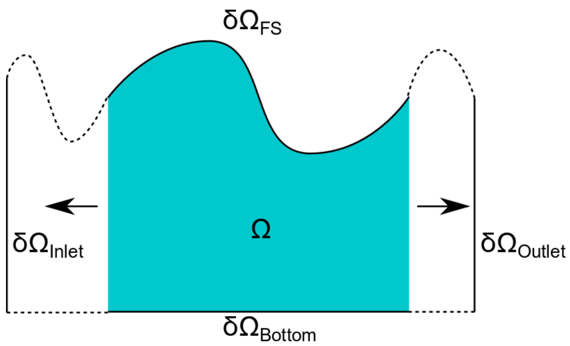

Figure 3.1: Computational domain Ω

3.4 Conclusions to Laplace 2D problem

We have briefly discussed the wave propagation problem using BEM in a 2D case, seeing that in this case we need to consider all the contours, including the Inlet, the Outlet, and the bottom. This approach is successfully applied in a 2D problem in the reference [2]. In 2D problem considering all the contours is not a heavy problem because the number of nodes is usually small compared with a 3D case, but in a 3D problem a computational less expensive method is required.

We will not continue with the 2D Laplace problem because the solution will not be useful for us in the 3D case.

Chapter 4

Waves propagations in 3D

4.1 General

In this chapter the 3D case will be discussed, looking for a method to solve the BEM using only the information about the free surface. This case is our main objective in order to can setup 6-DOF seakeeping simulations.

4.2 Incident waves

We can rewrite the sea waves system outside from our computational domain Ω

$$\begin{array}{rcl} z(x,y,t) & = & \sum_{j=1}^{n_{waves}} a_j \sin\left(k_j\left(x\cos(\beta)+y\sin(\beta)\right)-\omega_j t+\delta_j\right) \\ v_z(x,y,t) & = & \sum_{j=1}^{n_{waves}} -a_j\omega_j \cos\left(k_j\left(x\cos(\beta)+y\sin(\beta)\right)-\omega_j t+\delta_j\right) \\ \phi(x,z,t) & = & \sum_{j=1}^{n_{waves}} -\frac{a_j\omega_j}{k_j} \cos\left(k_j\left(x\cos(\beta)+y\sin(\beta)\right)-\omega_j t+\delta_j\right)\exp\left(k_j z\right) \\ k_j & = & \frac{\omega_j^2}{g} \end{array} \tag{4.1}$$

Where β is the heading angle, being 0 for stern waves. For this wave system still being valid the phase velocity from the equation 3.3. The purposed potential is compatible with the Laplace equation 2.3 as well.

4.3 BEM applied to Laplace 3D problem

4.3.1 Computational domain

We have a domain similar to the shown in the figure 3.1, but in this case 2 more boundaries must be considered in the missed direction, that we will call $\partial\Omega_{Front}$ and $\partial\Omega_{Back}$. As in the 2D case we will apply the reciprocal relation

$$\lambda(X,Y,Z)\phi(X,Y,Z;t) = \int_{\partial\Omega} \phi(x,y,z;t)\frac{\partial G(x,y,z,X,Y,Z)}{\partial \boldsymbol{n}(x,y,z)} - G(x,y,z,X,Y,Z)\frac{\partial\phi(x,y,z;t)}{\partial \boldsymbol{n}(x,y,z)}\mathrm{d}s(x,y,z) \tag{4.2}$$

We are focused into compute the gradient of the velocity potential along the free surface knowing the velocity potential value in each point one. Let we define the function $H(x,y,z,X,Y,Z)$ again

$$H(x,y,z,X,Y,Z) = \frac{\partial G(x,y,z,X,Y,Z)}{\partial \boldsymbol{n}(x,y,z)}$$

As in the Laplace equation for the 2D case, described in the chapter 3, we want to expand the domain Ω such that all the boundaries except the free surface will the infinity far, adding the boundary $\partial\Omega_{FS,I}$, where we know the velocity potential and their gradient from 4.1. In the figure 4.1 a schematic view of the expanded domain can be seen.

The main advantage is that, as happens with the Inlet and the Outlet, we know all the needed data about the velocity potential, so we can significantly reduce the linear system matrix dimensions, and as happens with the bottom, the geometry is so quite similar to the Ω_{FS} one, so no additional discretization or memory storage is needed.

This trick will only works if the Green's function $G(x, y, z, X, Y, Z)$, and their gradient $H(x, y, z, X, Y, Z)$, goes to zero as (x, y, z) goes to infinite. In 3D Laplace problems we can use the following Green's function:

$$G(x, y, z, X, Y, Z) = \frac{1}{\sqrt{(x-X)^2 + (y-Y)^2 + (z-Z)^2}} \tag{4.3}$$

$$H(x, y, z, X, Y, Z) = -\frac{(x-X, y-Y, z-Z)}{\left((x-X)^2 + (y-Y)^2 + (z-Z)^2\right)^{3/2}} \tag{4.4}$$

That in the limit

$$\lim_{(x-X)^2+(y-Y)^2+(z-Z)^2\to\infty} G(x, y, z, X, Y, Z) = 0 \tag{4.5}$$

$$\lim_{(x-X)^2+(y-Y)^2+(z-Z)^2\to\infty} H(x, y, z, X, Y, Z) = 0 \tag{4.6}$$

So in this case, if the potential of the incidents waves is a good function along all the free surface we can move from the domain shown in the figure 3.1 to the shown in the figure 4.1, due to along the other boundaries the Green's functions $G(x, y, z, X, Y, Z)$ and $H(x, y, z, X, Y, Z)$ are nulls, not computing in the equation 3.6.

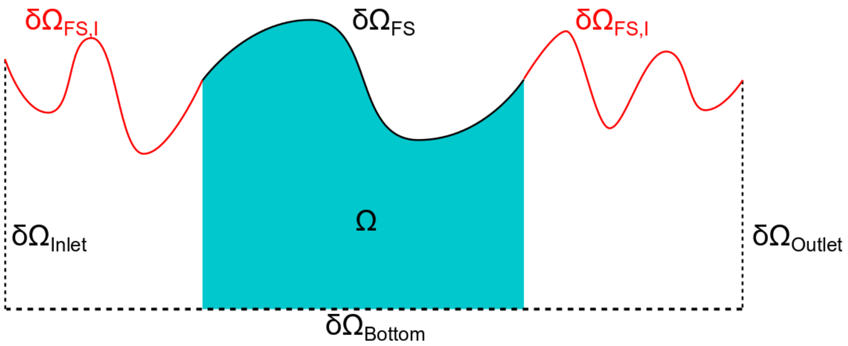

Figure 4.1: Computational domain Ω

4.3.2 Boundary conditions (BC)

In order to can purpose an evolution process we need to use the boundary conditions along the free surface. In the most general way we can rewrite the Bernoulli equation, that is the dynamic free surface boundary condition (DFSBC), as (see [3]):

$$\frac{D\phi(x, y, z; t)}{Dt} = \frac{1}{2}|\nabla\phi|^2 - gz(x, y; t) - \frac{p_0}{\rho} - \boldsymbol{U}(t)\cdot\nabla\phi(x, y, z; t) - \frac{\partial \boldsymbol{U}(t)}{\partial t}\cdot(x, y, 0) \tag{4.7}$$

Where p_0 is the atmospheric pressure, that we will consider null, and $\boldsymbol{U}(t)$ is the ship velocity, in this case will be a null vector. Since the material derivative denotes

$$\frac{\mathrm{D}f}{\mathrm{D}t} = \frac{\partial f}{\partial t} \nabla \phi(x, y, z; t) \cdot \nabla f$$

We can rewrite the dynamic boundary condition for this specific case as

$$\frac{\partial \phi(x, y, z; t)}{\partial t} = -\frac{1}{2} |\nabla \phi|^2 - g z(x, y; t) \tag{4.8}$$

Regarding the kinematic free surface boundary condition (KFSBC), in the most general way we can write that

$$\frac{\mathrm{D}z(x, y; t)}{\mathrm{D}t} = \frac{\partial \phi(x, y, z; t)}{\partial z} - \boldsymbol{U}(t) \cdot \nabla z(x, y; t) \tag{4.9}$$

Where we can expand the material derivative, writing the KFSBC for this specific case

$$\frac{\partial z(x, y; t)}{\partial t} = \frac{\partial \phi(x, y, z; t)}{\partial z} - \nabla \phi(x, y, z; t) \cdot \nabla z(x, y; t) \tag{4.10}$$

4.3.3 Time integration scheme

We may start the simulation in a initial condition where we know the full free surface shape and velocity potential, including the gradients.

$$\begin{array}{lcl} z(x, y; t = 0) & = & z_0(x, y) \\ \phi(x, y, z; t = 0) & = & \phi_0(x, y, z) \\ \nabla z(x, y; t = 0) & = & \nabla z_0(x, y) \\ \nabla \phi(x, y, z; t = 0) & = & \nabla \phi_0(x, y, z) \end{array} \tag{4.11}$$

In the computational part of the free surface is enough to know the free surface shape $z_0(x, y)$ and the velocity potential $\phi_0(x, y, z)$.

For simplicity we will use an Euler's integration scheme, but the same method can be easily applied for any other explicit time integrator, like the Adams-Bashforth ones.

In each time step we start knowing the shape of the free surface, and the velocity potential, and basically we want to compute these fields for the next time step. To do it the following steps are followed:

1. We use BEM to compute the velocity potential gradient, as will be described in the section 4.3.4.

$$\nabla \phi(x, y, z; t) = \mathrm{BEM}\left(\phi(x, y, z; t), z(x, y; t)\right) \tag{4.12}$$

2. We use the KFSBC to compute the derivative of the free surface elevation, and the DFSBC to know the derivative of the velocity potential.

$$\frac{\partial z(x, y; t)}{\partial t} = \mathrm{KFSBC}\left(\nabla \phi(x, y, z; t)\right) \tag{4.13}$$

$$\frac{\partial \phi(x, y, z; t)}{\partial t} = \mathrm{DFSBC}\left(z(x, y; t), \nabla \phi(x, y, z; t)\right) \tag{4.14}$$

3. And then we can perform the time integration.

$$z(x, y; t + \Delta t) = z(x, y; t) + \Delta t \frac{\partial z(x, y; t)}{\partial t} \tag{4.15}$$

$$\phi(x, y, z; t + \Delta t) = \phi(x, y, z; t + \Delta t) + \Delta t \frac{\partial \phi(x, y, z; t)}{\partial t} \tag{4.16}$$

4.3.4 Discrete Laplace solution using the BEM

As we have seen in the previous sections we want to use the BEM in order to compute the velocity potential gradient in the free surface from the velocity potential value known. In the equation 4.2 we can starting dividing the domain contour as shown in the figure 4.1, getting the computational one, denoted by $\partial\Omega_{FS}$, and the extension one, denoted by $\partial\Omega_{FS,I}$, where the free surface and the velocity potential and gradient is known for all the time instants.

$$\begin{aligned} &\tfrac{1}{2}\phi(X,Y,Z;t) = \\ &\int_{\partial\Omega_{FS}} \phi(x,y,z;t)\frac{\partial G(x,y,z,X,Y,Z)}{\partial \boldsymbol{n}(x,y,z)} - G(x,y,z,X,Y,Z)\frac{\partial \phi(x,y,z;t)}{\partial \boldsymbol{n}(x,y,z)}\mathrm{d}s(x,y,z) \\ &\int_{\partial\Omega_{FS,I}} \phi(x,y,z;t)\frac{\partial G(x,y,z,X,Y,Z)}{\partial \boldsymbol{n}(x,y,z)} - G(x,y,z,X,Y,Z)\frac{\partial \phi(x,y,z;t)}{\partial \boldsymbol{n}(x,y,z)}\mathrm{d}s(x,y,z) \end{aligned} \tag{4.17}$$

Where we already assumed that $(x,y,z) \in \partial\Omega$. We can start discretizing the velocity potential, assuming that the potential and their gradient changes smoothly enough. The our contours can be divided according to the grid:

$$\begin{aligned} \tfrac{1}{2}\phi_a = \; & \sum_{b=1}^{n_{FS}} \phi_b \int_{\partial\Omega_{FS}^b} H(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) - \sum_{b=1}^{n_{FS}} \frac{\partial \phi_b}{\partial \boldsymbol{n}_b}\int_{\partial\Omega_{FS}^b} G(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) \\ & \sum_{b=1}^{n_{FS,I}} \phi_b \int_{\partial\Omega_{FS,I}^b} H(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) - \sum_{b=1}^{n_{FS,I}} \frac{\partial \phi_b}{\partial \boldsymbol{n}_b}\int_{\partial\Omega_{FS,I}^b} G(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) \end{aligned} \tag{4.18}$$

The functions $G(x,y,z,X,Y,Z)$ and $H(x,y,z,X,Y,Z)$, according to the equations 4.3 and 4.4, are well defined in all the subintervals where $(x,y,z) \neq (X,Y,Z)$, and are so quite smooth, so we will change all the integrals that accomplish it for point evaluations.

$$\begin{aligned} \tfrac{1}{2}\phi_a = \; & \phi_a \int_{\partial\Omega_{FS}^a} H(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) - \frac{\partial \phi_a}{\partial \boldsymbol{n}_a}\int_{\partial\Omega_{FS}^a} G(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}) \\ & \sum_{\substack{b=1\\ b\neq a}}^{n_{FS}} \left(\phi_b H_{ba} - \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} G_{ba}\right) S_b + \sum_{b=1}^{n_{FS,I}} \left(\phi_b H_{ba} - \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} G_{ba}\right) S_b \end{aligned} \tag{4.19}$$

The remaining integrals must be treated carefully since the functions are singular in the center of the interval. $H(x,y,z,X,Y,Z)$ is an odd function, so the limit of the integral when the radius of the interval goes to zero is null, being well defined. Regarding the function $G(x,y,z,X,Y,Z)$ is an even function of order:

$$G(\boldsymbol{x},\boldsymbol{x_a}) = O\left(\frac{1}{|\boldsymbol{x} - \boldsymbol{x_a}|}\right)$$

Which their integral is defined if the function $z(x,y)$ is well defined as well. So the remaining integrals can be numerically computed, being mindful that:

1. Can't be evaluated at the point $\boldsymbol{x_a}$.

2. Changes too fast around the point $\boldsymbol{x_a}$.

We will discuss later how to solve this integrals, for the moment we will define new functions such that:

$$\hat{G}_{ab} = \begin{cases} \int_{\partial\Omega_{FS}^a} G(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}); & \text{if } a = b \\ G(\boldsymbol{x_b},\boldsymbol{x_a})S_b; & \text{if } a \neq b \end{cases} \tag{4.20}$$

$$\hat{H}_{ab} = \begin{cases} \int_{\partial\Omega_{FS}^a} H(\boldsymbol{x},\boldsymbol{x}_a)\mathrm{d}s(\boldsymbol{x}); & \text{if } a = b \\ H(\boldsymbol{x_b},\boldsymbol{x_a})S_b; & \text{if } a \neq b \end{cases} \tag{4.21}$$

So we can rewrite the equation 4.19

$$\frac{1}{2}\phi_a = \sum_{b=1}^{n_{FS}} \left(\phi_b \hat{H}_{ba} - \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} \hat{G}_{ba} \right) + \sum_{b=1}^{n_{FS,I}} \left(\phi_b \hat{H}_{ba} - \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} \hat{G}_{ba} \right) \tag{4.22}$$

Where we can move all the terms of the computational free surface that affects to the gradient of the velocity potential (that is the value that we want to compute) to left hand side, and let all the other ones in the right hand side of the equation

$$\sum_{b \in \partial\Omega_{FS}} \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} \hat{G}_{ba} = -\frac{1}{2}\phi_a + \sum_{b \in \partial\Omega_{FS}} \phi_b \hat{H}_{ba} + \sum_{b \in \partial\Omega_{FS,I}} \left(\phi_b \hat{H}_{ba} - \frac{\partial \phi_b}{\partial \boldsymbol{n}_b} \hat{G}_{ba} \right) \tag{4.23}$$

The equation 4.23, that has been written for the velocity potential at one point $\boldsymbol{x}_a$, can be written for all the points of the free surface along the computational domain using the matrix notation

$$\mathcal{G} \left[\frac{\partial \phi}{\partial \boldsymbol{n}} \right] = \left(\mathcal{H} - \frac{1}{2}\mathcal{I} \right) [\phi] + \mathcal{H}_{FS,I} [\phi]_{FS,I} - \mathcal{G}_{FS,I} \left[\frac{\partial \phi}{\partial \boldsymbol{n}} \right]_{FS,I} \tag{4.24}$$

Note that the area of the elements S_b has been included into the matrices. The equation 4.24 is a linear system of equations that can be numerically solved, either inverting the matrix, or using an iterative method. The matrix inversion is probably the best way for linear seakeeping codes, where the same matrix will be ever used, but in this case the iterative method is the faster way.

The method described along this section allows to us to compute the gradient of the velocity potential along the free surface knowing the potential in the computational free surface, and both velocity potential and the gradient along the extended free surface.

4.3.5 Integrals computation

In the equations 4.20 and 4.20 we have introduced two inconvenient integrals. Even though the functions are not well defined when $\boldsymbol{x} = \boldsymbol{x_a}$, their integrals it is. For instance, if we can assume that the free surface is fully planar ($z = 0$), the integrals can be analytically computed

$$\int_{y_a-\delta y}^{y_a+\delta y} \int_{x_a-\delta x}^{x_a+\delta x} G(x, y, 0, x_a, y_a, 0)\, \mathrm{d}x\, \mathrm{d}y = \frac{\delta x \, \operatorname{asinh}\left(\frac{\delta y}{\delta x}\right) + \delta y \, \operatorname{asinh}\left(\frac{\delta x}{\delta y}\right)}{\pi}$$

$$\int_{y_a-\delta y}^{y_a+\delta y} \int_{x_a-\delta x}^{x_a+\delta x} H(x, y, 0, x_a, y_a, 0)\, \mathrm{d}x\, \mathrm{d}y = 0$$

But can not be analytically computed for every function $z(x, y)$, being necessary to compute it in a numerical way.

In the figure 4.2 a schematic representation of the integration method is shown. Let we want to compute the integral for an element of the grid (x_a, y_a), then we subdivide the element in **a even number** of subelements of area $dx \cdot dy$, so we can assert that any subelement will be evaluated in the point (x_a, y_a), but as near as we want because we can ever add more subelements; then we can numerically approximate the integral by (here in after we will use only the function $G(\boldsymbol{x}, bsx_a)$, because same method can be applied to the function $H(\boldsymbol{x}, bsx_a)$).

$$\int_{\partial\Omega_{FS}^a} G(\boldsymbol{x}, \boldsymbol{x}_a) \mathrm{d}s(\boldsymbol{x}) \simeq \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} G(x_i, y_j, z(x_i, y_i), x_a, y_a, z(x_a, y_a))\, dx\, dy \tag{4.25}$$

Of course the value $z(x_a, y_a)$ is known, but not the function $z(x_i, y_i)$ because is evaluated in points that are not reflected in the grid, so we must compute this points from the available data. We will start renormalizing the coordinates such that:

$$\begin{aligned} u &= \frac{x - (x_a - Dx)}{2Dx} \\ v &= \frac{y - (y_a - Dy)}{2Dy} \end{aligned} \tag{4.26}$$

So we know the value of z for all the combinations of $u = 0, 0.5, 1$ and $v = 0, 0.5, 1$. Then, to can evaluate the function $z(u, v)$ for all u, v values we may to build a Spline surface with the data known from the 9 points shown in the figure 4.2. The Spline surface can be characterized as

$$z(u, v) = k_0 + k_u u + k_v v + k_{uv} uv + k_{uu} u^2 + k_{vv} v^2 + k_{uuv} u^2 v + k_{uvv} uv^2 + k_{uuvv} u^2 v^2 \tag{4.27}$$

In the equation 4.27 we have 9 unknown coefficients, but we have available $z(u, v)$ for 9 points, so have 9 unknowns with 9 equations that can be set as a linear system of equations, that results in the following coefficients:

$$\begin{aligned}
k_0 &= z(0,0) \\
k_u &= -z(1,0) + 4z\left(\tfrac{1}{2},0\right) - 3z(0,0) \\
k_v &= -z(0,1) + 4z\left(0,\tfrac{1}{2}\right) - 3z(0,0) \\
k_{uv} &= z(1,1) - 4z\left(1,\tfrac{1}{2}\right) + 3z(1,0) - 4z\left(\tfrac{1}{2},1\right) + 16z\left(\tfrac{1}{2},\tfrac{1}{2}\right) \\
&\quad -12z\left(\tfrac{1}{2},0\right) + 3z(0,1) - 12z\left(0,\tfrac{1}{2}\right) + 9z(0,0) \\
k_{uu} &= 2z(1,0) - 4z\left(\tfrac{1}{2},0\right) + 2z(1,0) \\
k_{vv} &= 2z(0,1) - 4z\left(0,\tfrac{1}{2}\right) + 2z(1,0) \\
k_{uuv} &= -2z(1,1) + 8z\left(1,\tfrac{1}{2}\right) - 6z(1,0) + 4z\left(\tfrac{1}{2},1\right) - 16z\left(\tfrac{1}{2},\tfrac{1}{2}\right) \\
&\quad +12z\left(\tfrac{1}{2},0\right) - 2z(0,1) + 8z\left(0,\tfrac{1}{2}\right) - 6z(0,0) \\
k_{uvv} &= -2z(1,1) + 4z\left(1,\tfrac{1}{2}\right) - 2z(1,0) + 8z\left(\tfrac{1}{2},1\right) - 16z\left(\tfrac{1}{2},\tfrac{1}{2}\right) \\
&\quad +8z\left(\tfrac{1}{2},0\right) - 6z(0,1) + 12z\left(0,\tfrac{1}{2}\right) - 6z(0,0) \\
k_{uuvv} &= 4z(1,1) - 8z\left(1,\tfrac{1}{2}\right) + 4z(1,0) - 8z\left(\tfrac{1}{2},1\right) + 16z\left(\tfrac{1}{2},\tfrac{1}{2}\right) \\
&\quad -8z\left(\tfrac{1}{2},0\right) + 4z(0,1) - 8z\left(0,\tfrac{1}{2}\right) + 4z(0,0)
\end{aligned} \tag{4.28}$$

So using the equation 4.25 to 4.28 we can compute the integrals in the equations 4.20 and 4.21.

4.4 BEM test

4.4.1 General

A Python script has been provided with this document in the subfolder **test**. In the script all this theory is tested in order to know if the BEM is well purposed, and the errors that can be expected from the method application.

In the test, for the moment, only one wave will be set, and the computational free surface will be big enough to contain 2 wave lengths. In this case a wave period of $T = 2.5s$ is used, resulting in a wave of 10 meters.

In the figure 4.3 the wave used, that runs in the x direction, is shown. The free surface will be extended while $G(x, y, z, X, Y, Z) > 0.1$.

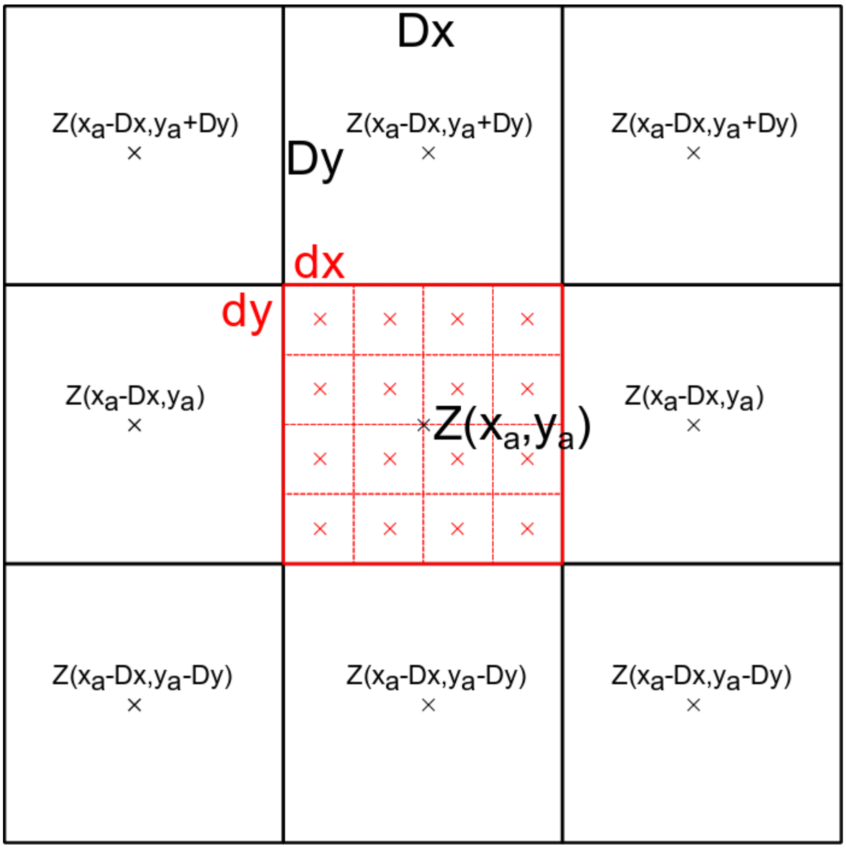

Figure 4.2: Integration method scheme

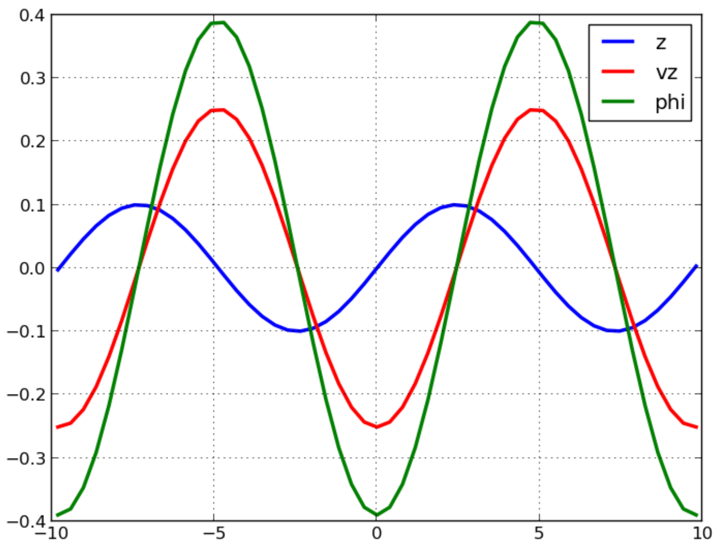

Figure 4.3: Wave used in the test

4.4.2 Direct method

The direct method consist in evaluate the velocity potential in several points using the equation 4.22, testing the error get.

If we apply the direct method for all the points of the computational free surface, we can compute the root mean square error as

$$RMS\,(nx, ny) = \sqrt{\frac{1}{nx\,ny} \sum_{i=1}^{nx} \sum_{j=1}^{ny} \left(\phi_{direct}(x_i, y_j) - \phi(x_i, y_j)\right)^2}$$

For $nx = 31$ and $ny = 15$ we have $RMS\,(31, 15) = 0.08$. In the figure 4.4 the analytic velocity potential, and the interpolated using the direct method, for a slice in the middle of the free surface ($y = 0$).

The results quality is good, and can be improved increasing the number of points in the computational free surface.

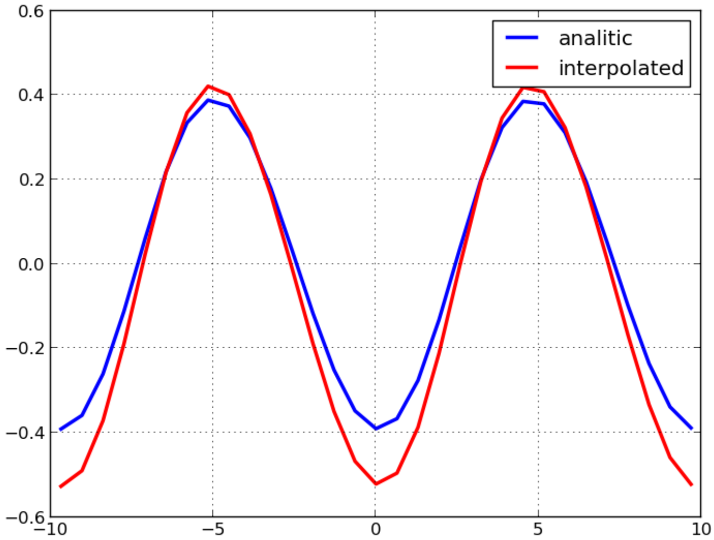

Figure 4.4: Direct method

4.4.3 BEM

In this case we want to apply the equation 4.23, where a linear system of equations is purposed in order to compute the gradient of the velocity potential, projected over the normal, for all the points of the grid. For $nx = 31$ and $ny = 15$ we have $RMS\,(31, 15) = 0.04$. In the figure 4.5 the analytic velocity potential, and the interpolated using the direct method, for a slice in the middle of the free surface ($y = 0$).

The results quality is nice like in the direct method. In order to get enough good results at least 15 points per wave length must be used (like in this application).

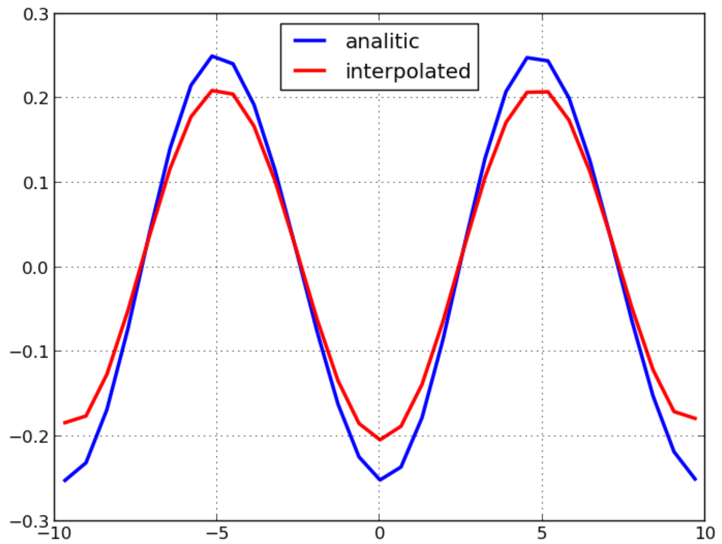

Figure 4.5: BEM solution

Bibliography

[1] Whye-Teong Ang. *A Beginner's Course in Boundary Element Methods*. Cambridge University Press, New York, 2007.

[2] V. Vinayan and S. A. Kinnas. A bem for the propagation of nonlinear planar free-surface waves. *Electronic Journal of Boundary Elements*, 5:17–40, 2007.

[3] Jinghai Yang. *Time domain, nonlinear theories on ship motions*. PhD thesis, University of Hawaii, 2004.