Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

Einar Aalvik

Numerical implementation of the pressure-patch model of ships on flows of arbitrary shear profiles

Master's thesis in Mechanical Engineering
Supervisor: Simen Å Ellingesn
June 2019

NTNU
Kunnskap for en bedre verden

Einar Aalvik

Numerical implementation of the pressure-patch model of ships on flows of arbitrary shear profiles

Master’s thesis in Mechanical Engineering
Supervisor: Simen Å Ellingesn
June 2019

Norwegian University of Science and Technology
Faculty of Engineering
Department of Energy and Process Engineering

NTNU
Norwegian University of
Science and Technology

Problem Description

In recent years, significant theoretical progress has been made by the group of Dr S. Å. Ellingsen on wave phenomena atop vertically sheared currents; a particular point of focus has been the waves behind moving wave sources – "ships". Recently, the theory has been confirmed by a series of laboratory experiments at Dr Ellingsen's lab in the Fluid Mechanics building, NTNU. The theory, which has been developed to handle a shear current of arbitrary vertical shear, depends on modelling the ship as a travelling patch of pressure which depresses the surface, yet while this this model goes back more than a century, it is imperfect. Today, a single pressure patch of the same size and shape as the desired wetted ship hull is used, however the actual depression of the water surface beneath the patch will be different depending on the speed and direction with which it travels. Thus, errors are made since the "ship" does not have the effective hull shape it is supposed to.

The project will use a numerical scheme to solve the inverse problem: given a desired wetted hull shape, speed and direction of motion, what must the pressure patch be to produce this shape? The problem may be solved using either Green's function techniques or direct numerical minimization approaches, and will be considered in increasing levels of complexity:

1. Develop a robust method for solving the simplest case: a 2D "ship" in deep water with no current.
2. Generalise the method to 2D ship models of gradually increasing complexity
   - (a) in finite water depth with no current
   - (b) with a linear shear profile
   - (c) with arbitrary 2D shear profile
3. Generalise to 3D ship models
   - (a) With no current (symmetrical solution)
   - (b) With linear shear profile
   - (c) With arbitrary shear current, unidirectional and spiralling
4. Assuming sub-points of 3 can be achieved, the effect of using a correct, rather than approximate, model on the following published predictions and results will be made:
   - (a) Using the method from 3a, test the wave predictions of
     - i. Darmon et al (2014) and Benzaquen et al. (2014) (deep water gravity waves)
     - ii. Ledesma-Alonso et al (2016) for thin films
   - (b) Using the method from 3b, test the predictions of Ellingsen (2014) and Li and Ellingsen (2016)

   (c) Using the method from 3c, test the theoretical predictions of Smeltzer et al (2019), and the recent laboratory measurements of Smeltzer et al (in review).

5. Develop a robust and user-friendly numerical library for use by Ellingsen's group and general publication.

This list is an absolute best-case scenario, assuming no major unforeseen hurdles manifest themselves; if it is completed it will be enough for an article in a leading international journal. Realistically, time will not allow all points to be covered. The project is an ambitious research project in the true sense of the word, with the inherent uncertainty this entails.

Abstract

In this thesis, a numerical method is developed for finding an external pressure distribution that models a ship on the water surface given the shape of the hull, the speed of the moving ship, the water depth, and a depth-dependent shear profile. Using a pressure patch to model a ship has commonly been done in the linear surface waves field over the years. These pressures are, however, typically kept constant for different ship velocities, water depth, and shear flow, and the effective ship hull in the water will change with these conditions. The traditional pressure patch method is therefore not an accurate model for a moving ship, and there is a need for a method that finds the pressure source that correctly models the ship hull, given the geometry of the hull, ship velocity, water depth, and shear flow.

The numerical method is a brute force method where a combination of mesh integration, FFT and finite differences is used to solve the problem. The integral equation used for finding the pressure patch is a Fredholm integral equation of the first kind, and solving the equation for the pressure can be regarded as an inverse problem. By using a Green's function method and discretizing the equation, the problem is solved by solving the linear equation system. Furthermore, the problem is solved in both 2D and 3D, and the numerical method can calculate solutions for an arbitrary depth-dependent shear profile.

The accuracy of the solutions provided by the numerical method is found to be reasonably accurate. The method can reproduce all the points in the given surface elevation with an upper bound relative error of less than 0.5% for all the cases in this study. It was found that the effect of ship velocity, water depth, and shear current profile had significant effects on the pressure patch. Moreover, the calculated pressure patch is dramatically different from the constant pressure patch that is traditionally used, where the traditionally used patch only gives the correct effective ship hull when the ship velocity is zero. However, there are some challenges with obtaining the solution of the inverse problem. As the Green's function is divergent for the point $G(0)$ in addition to being slowly convergent in the points close to zero, the Green's function will not be fully converged. Also, the radiation condition parameter ϵ needs to be set to a relatively high value in order to avoid periodic boundary issues.

Nevertheless, the resulting pressure patches are ensuring the wanted prescribed surface elevation on the water surface. However, the big pressure spikes occurring at the back of the pressure patches are indicating the need for a more physical prescribed surface elevation in the region of the ship. In the real world, a moving ship would be elevated and tilted as the pressure forces act on the hull. Hence, the calculated pressure sources should give a fair representation of moving the ship in the unnatural case of a non-changing effective ship hull.

Sammendrag

I denne oppgaven er det utviklet en numerisk metode for å finne en ekstern trykkfordeling som modellerer et skip på vannoverflaten gitt skrogets form, fartøyets fart, vanndybden og en dybdeavhengig skjærprofil. Trykkflekker har ofte blitt brukt for å modellere skip i kombinasjon med lineære overflatebølger gjennom årene. Disse trykkflekkene holdes imidlertid konstant for forskjellige fartshastigheter, vanndybde og skjærstrøm, som vil gjøre at det effektive skroget i vannet endres etter forholdene. Derfor er den tradisjonelle trykkflekkmetoden ikke en nøyaktig modell for skipsskrog, og det er behov for en metode som finner trykkflekken som korrekt modellerer skipsskroget, gitt skrogets geometri, hastighet, vanndybde og skjærstrømning.

Den numeriske metoden er brute en force-metoden der en kombinasjon av "mesh"-integrasjon, FFT og "finite difference" brukes til å løse problemet. Integral ligningen som brukes til å finne trykklappen, er en Fredholm integral likning av den første typen, og å løse for likningen for trykket kan betraktes som et inverst problem. Ved å bruke en Green's funksjon og diskretisere ligningen, kan problemet løses ved å løse det lineære ligningssystemet. Problemet løses i både 2D og 3D, og den numeriske metoden kan beregne løsninger for en vilkårlig dybdeavhengig skjærprofil.

Nøyaktigheten av løsningene som er gitt ved den numeriske metoden er funnet å være ganske gode. Metoden kan reprodusere alle punktene til den gitte overflaten med en øvre grense for relativ feil på mindre enn 0,5 % for alle testene i dette studiet. Det ble funnet at effekten av skipshastighet, vanndybde og skjærstrømprofil hadde signifikante effekter på trykkflekken. Dessuten er den beregnede trykkflekken veldig forskjellig fra den konstante trykkflekken som tradisjonelt brukes, hvor den tradisjonelt brukte flekken bare gir riktig skipsskrog når skipets hastighet er null. Imidlertid er det noen utfordringer med å få løsninger for det inverse problemet. Siden Greens funksjonen er divergerende for punktet $G(0)$ i tillegg til å være sakte konvergende for punktene nær null, vil Greens funksjonen ikke være fullt konvergert. I tillegg må strålingsbetingelsesparameteren ϵ settes til en relativt høy verdi for å unngå periodisk grense-problemer.

Likevel sikrer de resulterende trykkflekkene ønsket høyde på vannoverflaten. Imidlertid viser de store trykktaggene på baksiden av trykkflekken behovet for en mer fysisk ønsket overflatehøyde i skipets område. I den virkelige verden vil et skip bli forhøyet og rotert bakover ettersom trykk-kraften virker på skroget. Derfor er de beregnede trykkflekkene en rettferdig representasjon av skip i det unaturlige tilfellet av ikke-skiftende effektive skipsskrog.

Preface

This thesis is submitted for the degree of Master of Science at the Norwegian University of Science and Technology (NTNU). The work included in the thesis started in January 2019 at the Department of Energy and Process Engineering (EPT), NTNU, and was completed at the beginning of June 2019. The thesis was supervised by Associate Professor Simen Å Ellingsen.

I would like to thank Simen Å Ellingsen for all the invaluable information, wisdom, and enthusiasm throughout my time working on the project. This thesis truly would not be possible without his help. Yan Li has, as well, helped me countless hours with the project, and I thank her for her support and patience. I would also like to thank all the members of the Ellingsen's group for all the discussions and meetings we had throughout my time on the project. Finally, my greatest gratitude to my family and friends for supporting me and motivating me through thick and thin.

Trondheim, 2019-06-10

Einar Aalvik

Einar Aalvik

Contents

1 Introduction

Ship waves is a significant contribution to the dynamics of ships. While the waves themselves have been a curiosity for humans for a long time, the theory of ship waves has only been studied for a little over a century. A ship traveling through water will create the waves propagating away from the ship. Hence the creation of waves causes an energy loss in itself in addition to the viscous drag [1]. The presence of a shear current beneath the surface of the moving ship can greatly alter this energy loss in addition to some other attributes of ship waves [1, 2, 3, 4]. Special care is needed for calculating ship waves when introducing a shear current to the problem, hence a method called 'the pressure patch model' has been used in later years [2]. The pressure patch model uses an external pressure patch on the surface of the water to model a moving ship.

1.1 Background

Water waves occur all around the world and in many forms, from tsunamis and ocean tides to small waves created by ducks swimming in water. There is no doubt that water waves impact humans in both big and small ways. Ship waves are of special importance and can have huge impacts on the ship industry. Up to 30 % fuel consumption of ocean-going ships is from making waves [5]. Therefore, an understanding of water waves and ship wakes can have big implications in the design and maintenance of ships.

The theory of ship waves was first studied by the famous Lord Kelvin, which is why the ship wake is often called the Kelvin wake. One of his most important conclusions was that the wake line is offset from the path of the moving ship by an angle of $\arcsin(1/3) = 19.47°$ called the Kelvin angle. These results are to this day still used and referred to [6, 7, 2]. In [8], they observed that the Kelvin angle seemed to decrease for higher Froude numbers as a scale of Fr^{-1}. This was quickly resolved by Darmond and his colleagues [9] and Noblesse and his colleagues [7]. They show that the wave pattern seems narrower since the maximum amplitude of waves scales as Fr^{-1} for high Froude numbers, while the angle where no waves can be observed, remains constant and equal to the Kelvin angle of 19.47°.

Some decades after Kelvin's original treatment, T. H. Havelock developed a method for modeling ships with an external pressure applied to the water surface, thus creating a wave pattern behind the moving pressure [10]. This method is popularly called the pressure patch model, which have some benefits that recent studies have taken advantage of [6, 2, 3, 4]. This external pressure is only applied to the dynamic boundary condition and does not affect the equations of motion. Thus, it does not pose any restrictions on the flow vorticity like sources and sinks approaches. Hence, the pressure patch model allows ship wakes in the presence of a depth-dependent shear flow.

Introducing a depth-dependent shear current under the water surface of a moving ship, alter some key characteristics and implications of the waves. First of all in [2] Ellingsen uses a linear depth-dependent shear current to show that the combined Kelvin angle can now reach as much as 180°. Second of all, the functional relation between wave

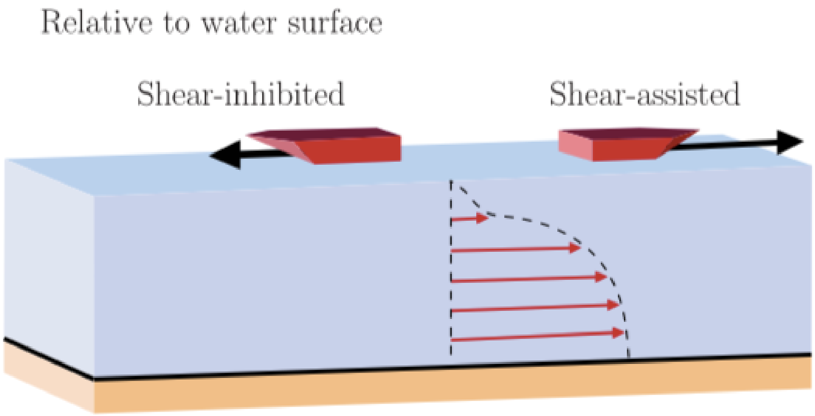


Figure 1: A Figure showing the definition of shear-inhibited and shear-assisted waves. The illustration is taken from [4].

frequency, wave number and propagation direction, the dispersion relation, is altered from the classical $\omega = \sqrt{(gk + \frac{\sigma k}{\rho})\tanh(kh)}$ for both the 2D and 3D case in presence of a shear current [1, 11]. Due to the difference in phase velocities for shear-inhibited and shear-assisted waves, mean kinetic energy is higher than mean potential energy for shear-assisted waves. Shear-inhibited and shear-assisted waves are defined as shown in Figure 1 as done in [4]. The opposite is true for shear-inhibited waves [1]. Hence, the principle of equipartition does not hold for waves traveling through the water with a shear current present. The shear current can, in a rough sense, alter the effective ship velocity when looking at the stationary wave pattern. Shear-inhibited waves will have a lower critical velocity, while shear-assisted waves will have a higher critical velocity. For instance, increasing the strength of the shear current in a shear-assisted system can cause a transition from subcritical waves to supercritical waves [3]. Also, an increase in side-on shear strength can cause a subcritical to supercritical transition [3]. Moreover, the shear current affects the wave resistance acting on the ship. Due to the asymmetrical wave pattern created by the shear, except for the cases where the shear current direction is directly upstream or downstream, there will be a lateral wave resistance component [3]. For a boat traveling on, for example, the Colombia river delta, the lateral wave resistance typically accounts for around 20% of the total wave resistance [4]. Furthermore, the velocity for maximum wave resistance is lowered in shear-inhibited waves and increased for shear-assisted waves. Besides, the maximum wave resistance in itself increases in shear-inhibited waves while decreasing in shear-assisted waves.

Arbitrary depth-dependent shear currents take a more complicated form than linearly depth-dependent shear. In [12], a direct integration method for the dispersion relation is developed. This is further used in [4] to develop a method for calculating transient ship waves and wave resistance. As for the stationary ship waves, the situation simplifies somewhat in comparison to the transient case. The implications of arbitrary shear profiles are similar to that of linear shear profiles, although the numbers change.

Similar to a shear current, the constraint of having a finite depth underneath the moving ship alter some key characteristics of the waves. In a finite depth, the dispersion relation will be changed being approximately $\omega = \sqrt{gh}$ for shallow waters. Thus, the critical velocity is lower for shallow waters, since the phase velocities are lower and transverse waves cannot keep up with the moving source for higher velocities [3]. The combination of finite depth and a depth-dependent shear current is not intuitive, and hence careful handling is required, and calculations need to be done on a case to case basis [3].

1.2 Thesis Motivation

As mentioned above, all the recent research concerning ship waves on depth-dependent shear flow in 3D are using the pressure patch model. Nonetheless, there are some downsides to using this model. The pressure patch has the same shape as the desired wetted ship hull. This gives a water surface depression equal to the hull when the ship is standing still. The model fails to recognize, however, that the actual surface depression underneath the patch will depend on the velocity of the ship, water depth, and the shear flow underneath.

A way to resolve this is to look at the *inverse problem*; given the wanted shape of the hull, what must the pressure patch be to produce this shape.

In general, an inverse problem is one where you use an observed effect to determine the nature of its cause, as opposed to a "forward" problem where you predict the effect when the cause is known. This particular inverse problem has not been treated before to the author's knowledge. Similar problems have been encountered in other areas, however. Binder, Blyth, and McCue developed a numerical method to calculate the inverse problem regarding free-surface flow past arbitrary topography [13]. In this case, the forward problem would be calculating the free-surface given the topography. Hence, the numerical tool determines the topography given the shape of the free surface. The method is used to calculate the topography giving wave-free solutions. The inverse problem to the pressure patch model is different, however, since the surface elevation is in this case caused by an external pressure patch, not the topography. Besides, the complexity of the problem is greatly enhanced by doing the calculations in 3D, while the topography inverse problem is only done in 2D.

Today, using the pressure patch model in combination with the linearized Euler equations is the only cheap way to simulate ship waves with vorticity. Thus, solving the inverse problem could be interesting for design purposes of ships and boats. Experiments are expensive and time-consuming and only done in the final stages of design. Solving the full Navier-Stokes equations using computational fluid dynamics, is far too computationally expensive for design purposes. Also, existing Green's function methods are not applicable in 3D with vorticity present. The results of studying ship waves in depth dependent shear flows can therefore possibly be of great use. Furthermore, the results of these studies could be used in operating decisions for ocean-going vessels. For example, are waves of small boats in the Colombia river delta heavily dependent on the shear current and ship velocity [4]. Hence being considerate of the velocity in regards to shear current can affect the wave resistance significantly.

1.3 Thesis Objective

This thesis aims to develop a method whereby the inverse pressure-patch problem may be solved numerically. That is, the aim is to devise a method to calculate an applied pressure distribution on the water surface which produces a desired wetted hull shape, given its velocity, a vertically varying horizontal shear current and a constant water depth. The problem will be addressed in settings of increasing complexity: first 2D then 3D geometry, first uniform, then linearly varying and finally arbitrary shear current. Furthermore, the thesis aims to develop a robust and user-friendly numerical library for use by Ellingsen's group and general publication.

2 Theoretical Framework

This section focuses on the theoretical framework required for solving the inverse problem as well as the forward problem. The forward problem is, in this thesis, referring to the classic problem of calculating the wave pattern given the external pressure distribution.

Initially, system definitions and geometry are defined. Then the linear surface waves theory is laid out for both 2D and 3D in the most general case of arbitrary shear and finite depth. The special cases like deep water and linear shear can be obtained from the general cases. More specifically, the governing equations, boundary conditions, and dispersion relation for linear surface waves will be described. These relations are the basis for both the forward and inverse problem.

2.1 2D

2.1.1 System Definitions and Geometry

Following the work of the Ellingsen group [2, 3, 4], a combination of different definitions is used for the 2D system. A ship moves with a steady velocity of V. A Cartesian coordinate system is used where the origin is put at the surface of the water. Hence, $z = 0$ defines the free water surface when everything is at rest, and the surface elevation is defined as $z = \hat{\zeta}(x, t)$, where x is the position on the surface and t is the time. The shear current profile is defined as U and the water depth as h. The system can be visualized, as done in Figure 2.

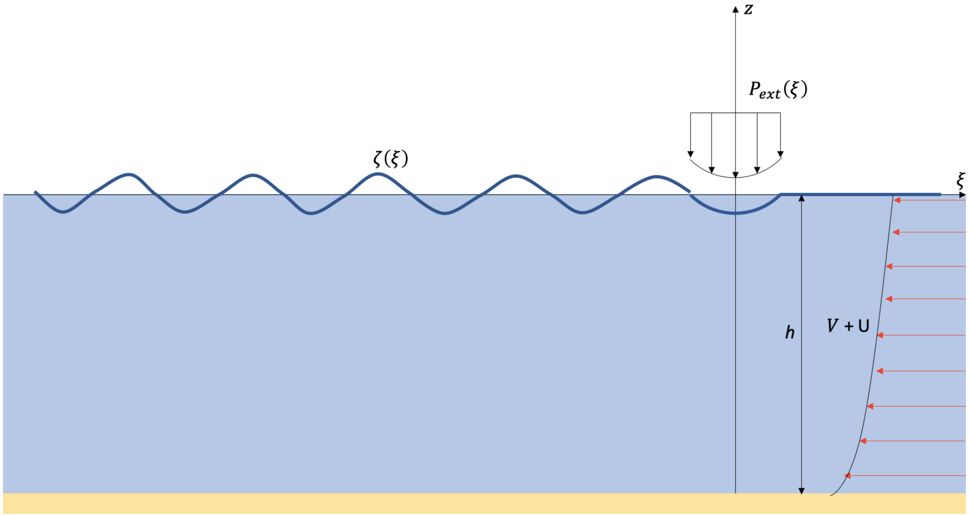


Figure 2: A figure illustrating the geometry of the problem. The figure shows steady state gravity waves generated from applying an external pressure source $\hat{p}_{ext}$. The ship moves at a constant velocity of V and the depth dependent arbitrary shear flow is U. $\hat{\zeta}$ represents the surface elevation and h the water depth.

The total velocity field can be written as

$$\boldsymbol{v}(x,z,t) = (U(z) + \hat{u}(x,z,t), \hat{w}(x,z,t)) \tag{1}$$

Furthermore, let the pressure field be $P(x,z,t)$ and $\hat{p}$ be the dynamic pressure defined as $\hat{p} = P + \rho g z$, where g is the gravitational acceleration and ρ is the water density. The flow is assumed to be incompressible and inviscid. All hatted quantities are assumed to be small, and the solutions will be linearized with respect to these. Since this thesis is only concerned with stationary waves, it is useful to introduce the variable ξ defined as $\xi = x - Vt$. Such a coordinate transformation will make the ship stand still with the stationary waves behind.

The prescribed surface elevation $\hat{\zeta}_s(\xi)$ is defined on $\xi \in [-l, l]$, such that $L = 2l$ is the length of the ship. Thus the surface elevation can be defined as

$$\hat{\zeta}(\xi) = \begin{cases} \hat{\zeta}_s, & \text{if } \xi \in [-l, l] \\ \hat{\zeta}[\hat{p}_{ext}(\xi)], & \text{otherwise} \end{cases} \tag{2}$$

where $\hat{p}_{ext}(\xi)$ is the unknown applied pressure patch. In this thesis, the external pressure patch will only have non-zero values in the region of the ship, $\xi \in [-l, l]$.

Underneath are a few definitions for later reference and convenience

$$U_0 = U(0) \ , \ U_0' = U'(0) \ , \ Fr = \frac{V}{\sqrt{gL}} \ , \ Fr_h = \frac{V}{\sqrt{gh}} \ , \ Fr_s = \frac{VU_0'}{g}$$

Fr, Fr_h, Fr_s will be referred to as the Froude number, the height Froude number, and the shear Froude number, respectively.

2.1.2 Governing Equations

Since waves are of a periodic nature, it is useful to make the ansatz, as done by Li on page 21 in [14], that all physical solutions depend on an overall oscillating factor $\mathrm{e}^{-i\omega t}$ where ω is the frequency of a propagating plane wave. The Fourier transform is therefore taken for all of the physical quantities in the x-direction.

$$\begin{aligned}&[\hat{u}(x,z,t), \hat{w}(x,z,t), \hat{p}(x,z,t), \hat{p}_{ext}(x,t), \hat{\zeta}(x,t)] = \\ &\int_{-\infty}^{\infty} \frac{dk}{2\pi}[u(k,z), w(k,z), p(k,z), p_{ext}(k), \zeta(k)]\mathrm{e}^{i(kx-\omega t)}\end{aligned} \tag{3}$$

Here k is the wave number.

As water in this problem is assumed to be inviscid and incompressible, the equations for linear surface waves are determined from continuity and the Euler equations; as presented in Equation (4a) and (4b).

$$\nabla \cdot \boldsymbol{v} = 0 \tag{4a}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \nabla)\boldsymbol{v} = -\frac{1}{\rho}\nabla P + \boldsymbol{g} \tag{4b}$$

Following the procedure of Ellingsen in [2] by linearizing and taking the Fourier transformation, the Euler equations and continuity can be expressed as

$$iku + \frac{\partial w}{\partial z} = 0 \tag{5a}$$

$$u(kU - \omega) - iwU' = -k\frac{p}{\rho} \tag{5b}$$

$$iw(kU - \omega) = -\frac{1}{\rho}\frac{\partial p}{\partial z} \tag{5c}$$

Eliminating u and p, gives the Rayleigh equation

$$\frac{\partial^2 w}{\partial z^2} - k^2 w = \frac{kU''}{kU - \omega} w \tag{6}$$

which is in accordance to the Rayleigh equation on page 390 in [15]. k can have both negative and positive values.

2.1.3 Boundary Conditions

To obtain the equation for linear surface waves, the boundary conditions need to be specified. Specifically, the boundary conditions of interest are the no-penetration condition, the kinematic boundary condition, and the dynamic boundary condition. The no-penetration condition says that the vertical velocity $\hat{w}$ needs to be zero at the bottom. Hence the no-penetration condition can be written as

$$w = 0 \ , \ \text{at } z = -h \tag{7}$$

The kinematic boundary condition states that if fluid particles have reached the free surface of the water, the fluid particle will always stay at the surface. The condition can be defined as

$$\frac{D\hat{\zeta}}{Dt} = \hat{w}\big|_{z=0} \tag{8}$$

This can be written as

$$w = i(kU - \omega)\zeta \ , \ \text{at } z = 0 \tag{9}$$

The dynamic boundary condition states that the pressure right above the free surface needs to equal the pressure right below. The ship waves in this problem fall under the category of gravity waves; hence, it is reasonable to neglect surface tension effects in the dynamic boundary condition [1]. Therefore, the condition can be stated as follows

$$P(\xi, \hat{\zeta}) = \hat{p}_{ext} \tag{10}$$

Applying the dynamic boundary condition to the expressions above gives

$$k^2 \frac{p_{ext}}{\rho} = -i\frac{\partial w}{\partial z}(kU - \omega) + iwkU' - gk^2\zeta \ , \ \text{at } z = 0 \tag{11}$$

For convenience, w_0 is defined as $w(k, 0)$ and $\frac{\partial w_0}{\partial z}$ as $\frac{\partial w(k,0)}{\partial z}$. Now, combining the kinematic and dynamic boundary condition

$$i(kU_0 - \omega)k^2 \frac{p_{ext}}{\rho} = (kU_0 - \omega)^2 \frac{\partial w_0}{\partial z} - (gk^2 + kU_0'(kU_0 - \omega))w_0 \tag{12}$$

The Rayleigh equation (6) does not, in general, have an analytic solution. One of the exceptions where the equation does have a solution is the case where the shear profile is linear, $U'' = 0$. A solution to $\frac{\partial^2 w}{\partial z^2} - k^2 w = 0$ is $w_h(k, z) = \frac{\sinh(k(z+h))}{\sinh(kh)}$. Using this result and the no-penetration condition as done in [4] gives

$$\begin{aligned}
&\int_{-h}^{0} dz[w_h(k,z)(\frac{\partial^2 w(k,z)}{\partial z^2} - k^2 w(k,z)) + w(k,z)(\frac{\partial^2 w_h(k,z)}{\partial z^2} - k^2 w_h(k,z))] \\
&= w_h(k,0)\frac{\partial w(k,0)}{\partial z} - w(k,0)\frac{\partial w_h(k,0)}{\partial z} \\
&= \int_{-h}^{0} dz \frac{kU'' w(k,z) w_h(k,z)}{kU - \omega}
\end{aligned} \tag{13}$$

Inserting the combination of the kinematic and dynamic boundary condition (12) into (13)

$$w_0(k)\Delta_R(k, \omega(k)) = ik(kU_0 - \omega)\tanh(kh)\frac{p_{ext}}{\rho} \tag{14}$$

where

$$\Delta_R(k,\omega(k)) = (1+I_g)(\omega - kU_0)^2 + \frac{(\omega - kU_0)(kU_0')\tanh(kh)}{k} - gk\tanh(kh)$$
$$I_g = \int_{-h}^{0} dz \frac{U''\sinh(k(z+h))}{(kU-\omega)\cosh(kh)} \frac{w(k,z)}{w(k,0)}$$

Inserting this into (9) gives

$$\zeta(k) = \frac{p_{ext} k\tanh(kh)}{\rho\Delta_R(k,\omega(k))} \tag{15}$$

Taking the fourier transform and using the kinematic pressure, $\frac{p_{ext}}{\rho} \to p_{ext}$, gives

$$\hat{\zeta}(\xi) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \frac{p_{ext} k\tanh(kh)\mathrm{e}^{ik\xi}}{\Delta_R(k,\omega(k))} \tag{16}$$

However, this integral is not defined due to singularities in the integrand. In order to define the integral, a radiation condition can be used.

The radiation condition replaces ω with $\omega + i\epsilon$ where ϵ is an infinitesimal positive variable which preferably would be set to zero in the end. By adding this extra term, the poles are moved off the real axis and integration path, and thus making it possible to evaluate the integral using the residue theorem or numerical integrations like fast Fourier transform. Another way to look at it is that the radiation condition adds an additional boundary condition infinitely far away. Physically, there will be waves propagating from both the source and from infinitely far away. The radiation condition constrains the steady-state solution to have the waves propagating only from the source. It is the same as letting the pressure be turned on gradually from $t = -\infty$ [16]. L.D. Landau and E.M. Lifshitz gives in paragraph 25 in [17] an approximate relation for ϵ as $\epsilon = 2\nu k^2$, where ν is the kinematic viscosity. The reason why it is desired to have ϵ go to zero in the end, is that viscous effects are neglected in this study, as is typically done for ship waves.

By replacing ω with $\omega + i\epsilon$ and neglecting second order terms of ϵ, the general solution now reads,

$$\hat{\zeta}(x,t) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \frac{p_{ext} k\tanh(kh)\mathrm{e}^{i(kx-\omega t)}}{\Delta_R(k,\omega(k)+i\epsilon)} \tag{17}$$

Furthermore, the phase velocity needs to be $c = V$ for stationary waves. Hence the angular frequency is $\omega = kV$ for 2D stationary waves. Now using the variable ξ defined as $\xi = x - Vt$,

$$\hat{\zeta}(\xi) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \frac{p_{ext} k\tanh(kh)\mathrm{e}^{ik\xi}}{\Delta_R(k,kV+i\epsilon)} \tag{18}$$

where

$$\Delta_R(k, kV + i\epsilon) = (1 + I_g)(kV)^2 + (VkU_0' - gk)\tanh(kh) + i\epsilon\Phi$$
$$\Phi = 2Vk(1 + Ig) + U_0'\tanh(kh)$$
$$I_g = \int_{-h}^{0} dz \frac{U''\sinh(k(z+h))}{k(U - V)\cosh(kh)} \frac{w(k, z)}{w(k, 0)}$$

For brevity, the surface velocity is set to zero, $U_0 = 0$. This is the general expression for linear surface waves on depth dependent flow and more special cases can be obtained from this equation. By setting the shear current to be linear, $U''(z) = 0$, I_g is zero, and the known equation for surface waves on linear shear current is obtained as in [1]. Likewise, the expression for no shear surface waves can be obtained from setting $U_0' = 0$, and deep water surface waves by letting $h \to \infty$.

2.1.4 The Inverse Problem

Equation (18) is the equation for the surface elevation of linear surface waves, and solving for $\hat{\zeta}$ is the forward problem. The problem of interest, the inverse problem, is finding the external pressure distribution $\hat{p}_{ext}$ corresponding to the given hull form $\hat{\zeta}_s$. This has never been done before to the author's knowledge. An approach for solving this pressure ship problem is to define a Green's function as follows

$$G(k) = \frac{k\tanh(kh)}{\Delta_R(k, kV + i\epsilon)} \tag{19a}$$
$$G(\xi) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} \frac{k\tanh(kh)\mathrm{e}^{ik\xi}}{\Delta_R(k, kV + i\epsilon)} \tag{19b}$$

where $G(k)$ is the Green's function in Fourier space, and $G(\xi)$ is the Green's function in real space. A Green's function can be seen as an impulse response and a propagator, and will in this case contain information about the propagation of particles at the surface.

The equation for surface elevation can then be expressed as

$$\hat{\zeta}(\xi) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} p_{ext}(k)G(k)\mathrm{e}^{ik\xi} \tag{20}$$

The convolution theorem gives

$$\hat{\zeta}(\xi) = \hat{p}_{ext}(\xi) * G(\xi) \tag{21a}$$
$$\hat{\zeta}(\xi) = \int_{-\infty}^{\infty} d\xi' \hat{p}_{ext}(\xi')G(\xi - \xi') \tag{21b}$$

Since the applied pressure is only non-zero at $\xi' \in [-l, l]$

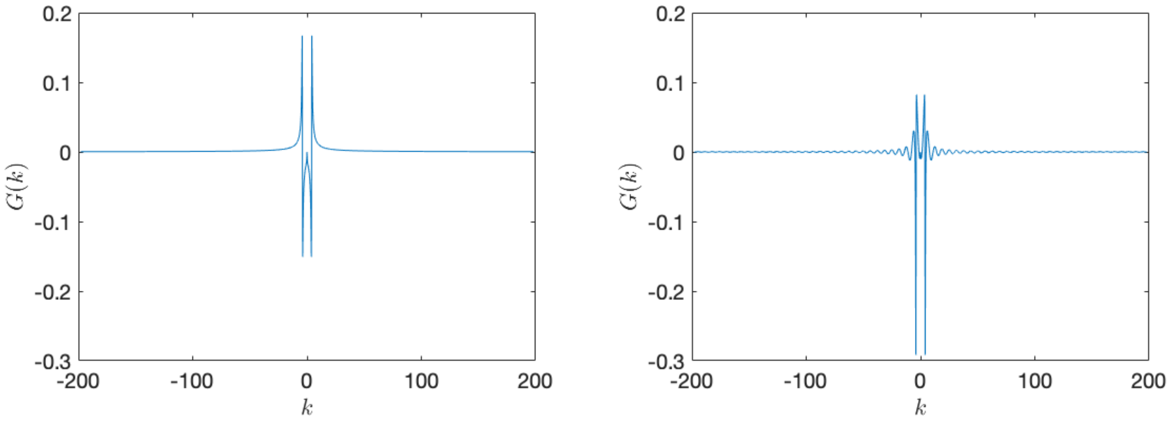
$$\hat{\zeta}(\xi) = \int_{-l}^{l} d\xi' \hat{p}_{ext}(\xi')G(\xi - \xi') \tag{22}$$

Moreover, the surface elevation in the region of the ship, $\xi \in [-l, l]$, is known. Thus, the equation (22) can be expressed as

$$\hat{\zeta}_s(\xi) = \int_{-l}^{l} d\xi' \hat{p}_{ext}(\xi') G(\xi - \xi') \quad , \quad \xi \in [-l, l] \tag{23}$$

Now, using equation (23), it is possible to solve for the external pressure patch $\hat{p}_{ext}$. When discretized, equation (23) makes a linear equation system where $\hat{\zeta}_s$ and G is known.

However, finding the Green's function can be troublesome. Although the radiation condition moves the poles of the integration path, the Green's function is still divergent at the point $\xi = 0$ due to lack of oscillations in the integrand. Furthermore, the integral is slowly convergent in points close to $\xi = 0$, meaning the numerical integration needs to be over a wider range of k-values in order to converge to a solution. In addition, the spacing between the discretized k-values needs to be sufficiently small to resolve the more frequent oscillations in the integrand occurring in points further away from $\xi = 0$. Plots showing an oscillating and non-oscillating integrand are shown in Figure (3).



(a) **(b)**

Figure 3: Plots of the Green's function integrand $G(k)$ for (a) $\xi = 0$ (non-oscillating integrand) (b) $\xi = 1$ (oscillating integrand). The plots are done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$, $Fr_s = 0$, $N_{ship} = 2^6$, $N_x = 2^{15}$, and $\epsilon = 2$

Furthermore, equation (23) is a Fredholm integral equation of type one and is known to be ill-conditioned and sensitive to error when using numerical methods [18]

2.2 3D

By transitioning into 3D, the physics of the problem changes significantly. The waves will now move in an extra dimension, hence dissipating quicker than for the 2D cases. Furthermore, the wave pattern will look different and more realistic in comparison to 2D. The procedure for obtaining the governing equations are, however, similar to the procedure for 2D.

2.2.1 System Definitions and Geometry

The extension to 3D is quite trivial for system definitions and geometry. The ship now moves with a velocity of $\boldsymbol{V}$ with a shear current profile $\boldsymbol{U}(z) = (U_x, U_y)$, water depth h, and surface elevation $\hat{\zeta}(\boldsymbol{x}, t)$. $\boldsymbol{x}$ is defined as $\boldsymbol{x} = (x, y)$ and the origin of the Cartesian coordinate system is still put at the water surface.

The total velocity field in 3D is

$$\boldsymbol{v}(x, y, z, t) = (U_x(z) + \hat{u}(x, y, z, t), U_y(z) + \hat{v}(x, y, z, t), \hat{w}(x, y, z, t)) \tag{24}$$

Furthermore, the dynamic pressure can now be written as

$$\hat{p}(x, y, z, t) = P(x, y, z, t) - \rho g z \tag{25}$$

and

$$\boldsymbol{\xi} = \boldsymbol{x} - \boldsymbol{V}t \tag{26}$$

The surface elevation can now be defined as

$$\hat{\zeta}(\boldsymbol{\xi}) = \begin{cases} \hat{\zeta}_s(\boldsymbol{\xi}), & \text{if } \boldsymbol{\xi} \in A \\ \hat{\zeta}[\hat{p}_{ext}(\boldsymbol{\xi})], & \text{otherwise} \end{cases} \tag{27}$$

where A is the region of the ship. Also in 3D, it is assumed that the external pressure will only have non-zero values in the region of the ship A.

Underneath are a few definitions for later reference and convenience

$$\boldsymbol{U}_0 = \boldsymbol{U}(0)\ ,\ \boldsymbol{U}_0' = \boldsymbol{U}'(0)\ ,\ Fr = \frac{|\boldsymbol{V}|}{\sqrt{gL}}\ ,\ Fr_h = \frac{|\boldsymbol{V}|}{\sqrt{gh}}\ ,\ Fr_s = \frac{|\boldsymbol{V}||\boldsymbol{U}_0'|}{g}$$

As for 2D, Fr, Fr_h, Fr_s will be referred to as the Froude number, the height Froude number, and the shear Froude number, respectively.

2.2.2 Governing Equations

Using the same ansatz as done in 2D and on page 21 in [14] by Li

$$\begin{aligned} &\hat{u}(\boldsymbol{x}, z, t), \hat{v}(\boldsymbol{x}, z, t), \hat{w}(\boldsymbol{x}, z, t), \hat{p}(\boldsymbol{x}, z, t), \hat{p}_{ext}(\boldsymbol{x}, t), \hat{\zeta}(\boldsymbol{x}, t)] = \\ &\int \frac{d^2k}{(2\pi)^2}[u(\boldsymbol{k}, z), v(\boldsymbol{k}, z), w(\boldsymbol{k}, z), p(\boldsymbol{k}, z), p_{ext}(\boldsymbol{k}), \zeta(\boldsymbol{k})]\mathrm{e}^{i(\boldsymbol{k}\cdot\boldsymbol{x}-\omega t)} \end{aligned} \tag{28}$$

The continuity equation (4a) and linearized Euler equation (4b) can now be expressed as

$$ik_x u + ik_y v + \frac{dw}{dz} = 0 \tag{29a}$$

$$u(\boldsymbol{k} \cdot \boldsymbol{U} - \omega) - iw\frac{dU_x}{dz} = -k_x \frac{p}{\rho} \tag{29b}$$

$$v(\boldsymbol{k} \cdot \boldsymbol{U} - \omega) - iw\frac{dU_y}{dz} = -k_y \frac{p}{\rho} \tag{29c}$$

$$iw(\boldsymbol{k} \cdot \boldsymbol{U} - \omega) = -\frac{1}{\rho}\frac{dp}{dz} \tag{29d}$$

Eliminating u, v and p, gives the Rayleigh equation

$$\frac{\partial^2 w}{\partial z^2} - k^2 w = \frac{\boldsymbol{k} \cdot \boldsymbol{U}''}{\boldsymbol{k} \cdot \boldsymbol{U} - \omega} w \tag{30}$$

where $\boldsymbol{U}'' = \frac{d^2\boldsymbol{U}}{dz^2}$.

2.2.3 Boundary Conditions

The same boundary conditions are needed in 3D as in 2D. The no-penetration condition is exactly the same and can be written as

$$w = 0\ ,\ \text{at } z = -h \tag{31}$$

The kinematic boundary condition, $\frac{D\hat{\zeta}}{Dt} = \hat{w}\big|_{z=0}$, can be expressed as follows in 3D

$$w = i(\boldsymbol{k} \cdot \boldsymbol{U} - \omega)\zeta\ ,\ \text{at } z = 0 \tag{32}$$

and the 3D dynamic boundary condition reads

$$k^2 \frac{p_{ext}}{\rho} = -i\frac{\partial w}{\partial z}(\boldsymbol{k} \cdot \boldsymbol{U} - \omega) + iw(\boldsymbol{k} \cdot \boldsymbol{U}') - gk^2\zeta\ ,\ \text{at } z = 0 \tag{33}$$

The same definitions as in 2D are made regarding the vertical velocity on the water surface. w_0 is defined as $w(k, 0)$ and $\frac{\partial w_0}{\partial z}$ as $\frac{\partial w(k,0)}{\partial z}$. Combining the dynamic and kinematic boundary condition then yields the relation

$$i(\boldsymbol{k} \cdot \boldsymbol{U}_0 - \omega)k^2 \frac{p_{ext}}{\rho} = (\boldsymbol{k} \cdot \boldsymbol{U}_0 - \omega)^2 \frac{\partial w_0}{\partial z} - (gk^2 + \boldsymbol{k} \cdot \boldsymbol{U}_0'(\boldsymbol{k} \cdot \boldsymbol{U}_0 - \omega))w_0 \tag{34}$$

The Rayleigh equation (30) does, in general, not have a analytic solution. One of the exceptions where the equation does have a solution is the case where the shear profile is linear, $\boldsymbol{U}'' = \boldsymbol{0}$. A solution to $\frac{\partial^2 w}{\partial z^2} - k^2 w = 0$ is $w_h(\boldsymbol{k}, z) = \frac{\sinh(k(z+h))}{\sinh(kh)}$. Using this result no-penetration condition, $w(\boldsymbol{k}, -h) = 0$, as done in [4] gives

$$\begin{aligned}
&\int_{-h}^{0} dz[w_h(\boldsymbol{k}, z)(\frac{\partial^2 w(\boldsymbol{k}, z)}{\partial z^2} - k^2 w(\boldsymbol{k}, z)) + w(\boldsymbol{k}, z)(\frac{\partial^2 w_h(\boldsymbol{k}, z)}{\partial z^2} - k^2 w_h(\boldsymbol{k}, z))] \\
&= w_h(\boldsymbol{k}, 0)\frac{\partial w(\boldsymbol{k}, 0)}{\partial z} - w(\boldsymbol{k}, 0)\frac{\partial w_h(\boldsymbol{k}, 0)}{\partial z} \\
&= \int_{-h}^{0} dz \frac{(\boldsymbol{k} \cdot \boldsymbol{U}'')w(\boldsymbol{k}, z)w_h(\boldsymbol{k}, z)}{\boldsymbol{k} \cdot \boldsymbol{U} - \omega}
\end{aligned} \tag{35}$$

Inserting the combination of the kinematic and dynamic boundary condition (34) into (35)

$$w_0(\boldsymbol{k})\Delta_R(\boldsymbol{k}, \omega(\boldsymbol{k})) = i(\boldsymbol{k} \cdot \boldsymbol{U}_0 - \omega)k\tanh(kh)\frac{p_{ext}}{\rho} \tag{36}$$

where

$$\begin{aligned}
\Delta_R(\boldsymbol{k}, \omega(\boldsymbol{k})) &= (1 + I_g)(\omega - \boldsymbol{k} \cdot \boldsymbol{U}_0)^2 \\
&+ \frac{(\omega - \boldsymbol{k} \cdot \boldsymbol{U}_0)(\boldsymbol{k} \cdot \boldsymbol{U}_0')\tanh(kh)}{k} - gk\tanh(kh) \\
I_g &= \int_{-h}^{0} dz \frac{(\boldsymbol{k} \cdot \boldsymbol{U}'')w(\boldsymbol{k}, z)\sinh(k(z+h))}{k(\boldsymbol{k} \cdot \boldsymbol{U} - \omega)w(\boldsymbol{k}, 0)\cosh(kh)}
\end{aligned}$$

Inserting this into (32) gives

$$\zeta(\boldsymbol{k}) = \frac{p_{ext}k\tanh(kh)}{\rho\Delta_R(\boldsymbol{k}, \omega(\boldsymbol{k}))} \tag{37}$$

When using the kinematic pressure $\frac{p_{ext}}{\rho} \rightarrow p_{ext}$ and doing the fourier transform, the equation for the surface elevation reads

$$\hat{\zeta}(\boldsymbol{x}, t) = \int \frac{d^2k}{(2\pi)^2} \frac{p_{ext}k\tanh(kh)\mathrm{e}^{i(\boldsymbol{k} \cdot \boldsymbol{x} - \omega t)}}{\Delta_R(\boldsymbol{k}, \omega)} \tag{38}$$

For the 3D case, the angular frequency ω has to be $\omega = \boldsymbol{k} \cdot \boldsymbol{V}$ in order to obtain stationary waves behind ships traveling at a constant speed of $\boldsymbol{V}$. Hence, by applying the radiation condition, the equation for surface waves is

$$\hat{\zeta}(\boldsymbol{x}, t) = \int \frac{d^2k}{(2\pi)^2} \frac{p_{ext}k\tanh(kh)\mathrm{e}^{i(\boldsymbol{k} \cdot \boldsymbol{x} - (\boldsymbol{k} \cdot \boldsymbol{V})t)}}{\Delta_R(\boldsymbol{k}, \boldsymbol{k} \cdot \boldsymbol{V} + i\epsilon)} \tag{39}$$

Now using the variable $\boldsymbol{\xi}$ defined as $\boldsymbol{\xi} = \boldsymbol{x} - \boldsymbol{V}t$, the final equation for the stationary surface elevation is

$$\hat{\zeta}(\boldsymbol{\xi}) = \int \frac{d^2k}{(2\pi)^2} \frac{p_{ext}k\tanh(kh)\mathrm{e}^{i(\boldsymbol{k}\cdot\boldsymbol{\xi})}}{\Delta_R(\boldsymbol{k}, \boldsymbol{k}\cdot\boldsymbol{V} + i\epsilon)} \tag{40}$$

where

$$\begin{aligned}
\Delta_R(\boldsymbol{k}, \omega(\boldsymbol{k})) &= (1 + I_g)(\boldsymbol{k}\cdot\boldsymbol{V})^2 + \left(\frac{(\boldsymbol{k}\cdot\boldsymbol{V})(\boldsymbol{k}\cdot\boldsymbol{U}_0')}{k} - gk\right)\tanh(kh) + i\epsilon\Phi \\
\Phi &= 2(\boldsymbol{k}\cdot\boldsymbol{V})(1 + I_g) + \boldsymbol{U}_0'\tanh(kh) \\
I_g &= \int_{-h}^{0} dz \frac{(\boldsymbol{k}\cdot\boldsymbol{U}'')\sinh(k(z+h))}{k(\boldsymbol{k}\cdot\boldsymbol{U} - \boldsymbol{k}\cdot\boldsymbol{V})\cosh(kh)} \frac{w(\boldsymbol{k}, z)}{w(\boldsymbol{k}, 0)}
\end{aligned}$$

As for 2D, the surface velocity is set to zero, $\boldsymbol{U}_0 = 0$. The different special cases can be obtained from this equation. By setting the shear current to be linear, $\boldsymbol{U}''(z) = 0$, I_g is zero and the known equation for surface waves on linear shear current is obtained as in [3]. Likewise, the expression for no shear surface waves can be obtained from setting $\boldsymbol{U}_0' = 0$, and deep water surface waves by letting $h \to \infty$.

2.2.4 The Inverse Problem

Equation (40) is the equation for the surface elevation of linear surface waves, and solving for $\hat{\zeta}$ is the forward problem. The problem of interest, the inverse problem, is finding the external pressure distribution $\hat{p}_{ext}$ corresponding to the given hull form $\hat{\zeta}_s$. This has never been done before. An approach for solving this pressure ship problem is to define a Green's function as follows

$$G(\boldsymbol{k}) = \frac{k\tanh(kh)}{\Delta_R(\boldsymbol{k}, \boldsymbol{k}\cdot\boldsymbol{V} + i\epsilon)} \tag{41a}$$

$$G(\boldsymbol{\xi}) = \int \frac{d^2k}{(2\pi)^2} \frac{k\tanh(kh)\mathrm{e}^{i(\boldsymbol{k}\cdot\boldsymbol{\xi})}}{\Delta_R(\boldsymbol{k}, \boldsymbol{k}\cdot\boldsymbol{V} + i\epsilon)} \tag{41b}$$

where $G(\boldsymbol{k})$ is the Green's function in Fourier space, and $G(\boldsymbol{\xi})$ is the Green's function in real space. A Green's function can be seen as an impulse response and a propagator, and will in this case contain information about the propagation of particles at the surface.

The equation for surface elevation can then be expressed as

$$\hat{\zeta}(\boldsymbol{\xi}) = \int \frac{d^2k}{(2\pi)^2} p_{ext}(\boldsymbol{k}) G(\boldsymbol{k}) \mathrm{e}^{i(\boldsymbol{k}\cdot\boldsymbol{\xi})} \tag{42}$$

The convolution theorem then gives

$$\hat{\zeta}(\boldsymbol{\xi}) = \hat{p}_{ext}(\boldsymbol{\xi}) * G(\boldsymbol{\xi}) \tag{43a}$$

$$\hat{\zeta}(\boldsymbol{\xi}) = \int d^2\xi' \hat{p}_{ext}(\boldsymbol{\xi}') G(\boldsymbol{\xi} - \boldsymbol{\xi}') \tag{43b}$$

Moreover, since the applied pressure is only non-zero at $\boldsymbol{\xi} \in \boldsymbol{A}$

$$\hat{\zeta}(\xi) = \int_{\boldsymbol{A}} d^2\xi' \hat{p}_{ext}(\boldsymbol{\xi}') G(\boldsymbol{\xi} - \boldsymbol{\xi}') \tag{44}$$

As in the 2D case, finding the Green's function can be troublesome. Although the radiation condition moves the poles of the integration path, the 3D Green's function is also divergent at the point $\boldsymbol{\xi} = \mathbf{0}$ due to lack of oscillations in the integrand. Similarly, the integral is slowly convergent in points close to $\boldsymbol{\xi} = \mathbf{0}$, meaning the numerical integration needs to be over a wider range of $\boldsymbol{k}$-values in order to converge to a solution. In addition, the spacing between the discretized $\boldsymbol{k}$-values needs to be sufficiently small in order to resolve the more frequent oscillations in the integrand occurring in points further away from $\boldsymbol{\xi} = \mathbf{0}$. Plots showing an oscillating and non-oscillating integrand in the 3D case are shown in Figure (4).

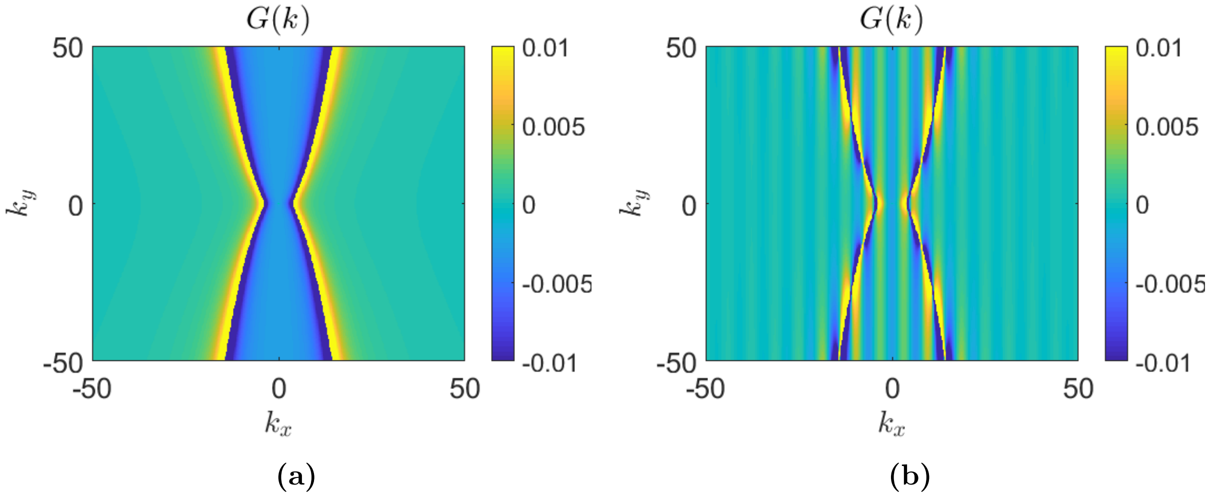


Figure 4: Plots of the Green's function integrand $G(K)$ for (a) $\boldsymbol{\xi} = \mathbf{0}$ (non-oscillating integrand) (b) $\xi_x = 1$, $\xi_y = 0$ (oscillating integrand). The plots are done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$, $Fr_s = 0$, $N_{x,ship} = 2^6$, $N_x = 2^{13}$, and $\epsilon = 2$

3 Methodology

The numerical method for solving the inverse problem was created in MATLAB and uses a direct integration method. Also the verification, like solving the forward problem, is done in MATLAB. SI units are used for all dimensional variables and parameters.

3.1 The code

3.1.1 Initializing

Initially, the code takes in some user-defined parameters. Due to the generality of the code, there are many degrees of freedom when defining a system to calculate. The tables (1) and (2) shows an overview of the different parameters that needs to be defined. The numerical methods for solving the inverse problem can be found in Appendix B.

Table 1: Table showing an overview of the different mandatory user defined parameters for 2D calculations

z	A vector of the discretized variable z. This parameter is used to get the water-depth in addition to being used in the equations.
L	The length of the ship in longitudinal direction in meters.
N_{ship}	Number of points on the ship in x-direction.
N_x	Number of points in the domain in x-direction.
ϵ	The parameter used in the radiation condition.
V	The velocity of the ship.
U	The depth dependent shear current velocity as seen from the surface of the water. The vector should be discretized to match the water depth z
U_0'	The derivative of the shear current velocity evaluated at the surface.
U''	The double derivative of the depth dependent shear current velocity as seen from the surface of the water. The vector should be discretized to match the water depth z
$\hat{\zeta}_s$	The wanted surface elevation in the area of the ship. Typically this means the mean wet hull of the ship.

Table 2: Table showing an overview of the different mandatory user defined parameters for 3D calculations

z	A vector of the discretized variable z. This parameter is used to get the waterdepth in addition to being used in the equations.
L	The length of the ship in longitudinal direction in meters.
$N_{x,ship}$	Number of points on the ship in x-direction.
N_x	Number of points in the domain in x-direction.
AR	The aspect ratio of the ship defined as $AR = W/L$ where W is the width of the ship. The aspect ratio is also used to define the number of points in y-direction of both the ship and the domain, $N_y = N_x \cdot AR$.
ϵ	The parameter used in the radiation condition.
V_x	The velocity of the ship in x-direction.
V_y	The velocity of the ship in y-direction.
U_x	The depth dependent shear current velocity in x-direction as seen from the surface of the water. The vector should be discretized to match the water depth z
U_y	The depth dependent shear current velocity in y-direction as seen from the surface of the water. The vector should be discretized to match the water depth z
$U'_{0,x}$	The derivative of the shear current velocity in x-direction evaluated at the surface.
$U'_{0,y}$	The derivative of the shear current velocity in y-direction evaluated at the surface.
U''_x	The double derivative of the depth dependent shear current velocity in x-direction. The vector should be discretized to match the water depth z
U''_y	The double derivative of the depth dependent shear current velocity in y-direction. The vector should be discretized to match the water depth z
$\hat{\zeta}_s$	The wanted surface elevation in the area of the ship. Typically this means the mean wet hull of the ship.

3.1.2 Finding the vertical velocity

Due to the complexity and generality of the problem, a numerical method is used for calculating the vertical velocity in Fourier space. More specifically, a central difference scheme of second order is used to calculate w in the Rayleigh equation (6) and (30). Hence, the scheme can be formulated as follows,

$$w_{i+1} + w_{i-1} - \left(2 + \Delta z^2 k^2 + \Delta z \frac{U''}{U - V}\right) w_i = 0 \tag{45a}$$

$$w_{i+1} + w_{i-1} - \left(2 + \Delta z^2 k^2 + \Delta z \frac{\boldsymbol{k} \cdot \boldsymbol{U}''}{\boldsymbol{k} \cdot \boldsymbol{U} - \boldsymbol{k} \cdot \boldsymbol{V}}\right) w_i = 0 \tag{45b}$$

where w is discretized in z-direction and is still a function of k. There are some problems with finding the vertical velocity however. First of all, the scheme cannot solve the problem when $U - V = 0$ for the 2D case and $\boldsymbol{k} \cdot \boldsymbol{U} - \boldsymbol{k} \cdot \boldsymbol{V} = 0$ for the 3D case. This is what is called a critical layer and needs careful handling [14]. The critical layers happens at a specific water depth z_c. In 2D, it is only a problem if the shear current velocity is greater than the speed of the moving ship. In 3D however, the problem can only be avoided if the shear current is parallel to the moving ship for the whole water depth. Second of all, the 3D case is not defined for $\boldsymbol{k} = \boldsymbol{0}$. Both the critical layer and the $\boldsymbol{k} = \boldsymbol{0}$ problem is ignored in this thesis by setting $\Delta z \frac{U''}{U-V} = 0$ and $\Delta z \frac{\boldsymbol{k} \cdot \boldsymbol{U}''}{\boldsymbol{k} \cdot \boldsymbol{U} - \boldsymbol{k} \cdot \boldsymbol{V}} = 0$ for the 2D and 3D case respectively.

3.1.3 Finding I_g

The next critical step in the numerical calculations is finding the function I_g. As I_g is a function of k and a big k-domain is needed for accurate calculations, the simple but quick trapezoid method is used for the integration. Also here, critical layers are a problem. Here the critical layers appear as poles in the integrand when $z = z_c$. Similar to finding the vertical velocity, the problem is ignored by setting the integrand equal to zero at the critical depths z_c. As I_g is a function of k, this has to be done for all the different k-values where a critical layer exists. Furthermore, the integrand is not defined when $k = 0$. However, since the Green's function integrand approaches zero in the limit of $k \to 0$, I_g is set to zero at this point.

3.1.4 The Inverse Problem

Finding the Green's function matrix is a vital part of solving the inverse problem. Also, the integral is divergent for $G(0)$ and slowly convergent for other values. Hence, careful handling of the Green's function is needed. To treat the integral of the Green's function similar to the integral in the forward problem, the same k-mesh is used as in the forward problem. Since the forward problem uses a Fast Fourier Transform scheme, the k-mesh is restricted to equidistant points in a finite domain. This causes some issues for the integration as a high integration limit difference is needed for convergence while small distances in the area of the poles are necessary for an accurate result.

The integrand $G(\boldsymbol{k})$ also causes some problems in MATLAB when $\boldsymbol{k} = 0$. Hence the limit $\lim\limits_{k \to 0} G(\boldsymbol{k}) = 0$ is done analytically and set manually in the code. As for I_g, the simple but quick trapezoid scheme is used for the integration due to the size of the k-domain. The double integral is done for each point $\boldsymbol{\xi} - \boldsymbol{\xi}'$

Now, the external pressure distribution or pressure patch can be calculated using the Green's function matrix and the prescribed surface elevation. The discretized relation reads as follows.

$$\zeta_i^{(ps)} = \sum_j^{N_x} \hat{p}_{ext}(\xi_i) G(\xi_i - \xi_j) \Delta\xi_x \tag{46a}$$

$$\zeta_{i,j}^{(ps)} = \sum_k^{N_y} \sum_t^{N_x} \hat{p}_{ext}(\xi_{i,j}) G(\xi_{i,j} - \xi_{k,t}) \Delta\xi_x \Delta\xi_y \tag{46b}$$

where $\zeta_i^{(ps)}$ and $\zeta_{i,j}^{(ps)}$ are the discretized prescribed surface elevation for the 2D case and 3D case respectively. This is a linear equation system that can be solved by using backslash in MATLAB.

3.2 Cases

In order to test the pressure patch's dependence on the velocity of the ship, water depth, and shear current profile, some test cases have been constructed.

3.2.1 Overview

The resulting choice of parameters for the different cases are illustrated in the tables (3) and (4).

Table 3: Table showing the different 2D test cases.

Fr	$\mathrm{Fr_h}$	$\mathrm{Fr_s}$	Shear Direction	Shear Profile
0.3	0.1	0	-	-
0.5	0.1	0	-	-
0.5	0.9	0	-	-
0.5	0.1	0.5	Assisted	Linear
0.5	0.1	0.5	Inhibited	Linear
0.5	0.1	0.5	Assisted	Exponential
0.5	0.1	0.5	Inhibited	Exponential

Table 4: Table showing the different 3D test cases.

Fr	$\mathrm{Fr_h}$	$\mathrm{Fr_s}$	Shear Direction	Shear Profile
0.3	0.1	0	-	-
0.5	0.1	0	-	-
0.5	0.9	0	-	-
0.5	0.1	0.5	Assisted	Linear
0.5	0.1	0.5	Inhibited	Linear
0.5	0.1	0.5	Side-On	Linear
0.5	0.1	0.5	Assisted	Exponential
0.5	0.1	0.5	inhibited	Exponential
0.5	0.1	0.5	Side-On	Exponential

3.2.2 2D

For simplicity, a Gaussian distribution with length $L = 1$ is used for the prescribed surface elevation as shown in Figure (5). It is described by equation (47).

$$\hat{\zeta}_s = -\frac{1}{15}\mathrm{e}^{-(\frac{\pi}{l}\xi)^2} \tag{47}$$

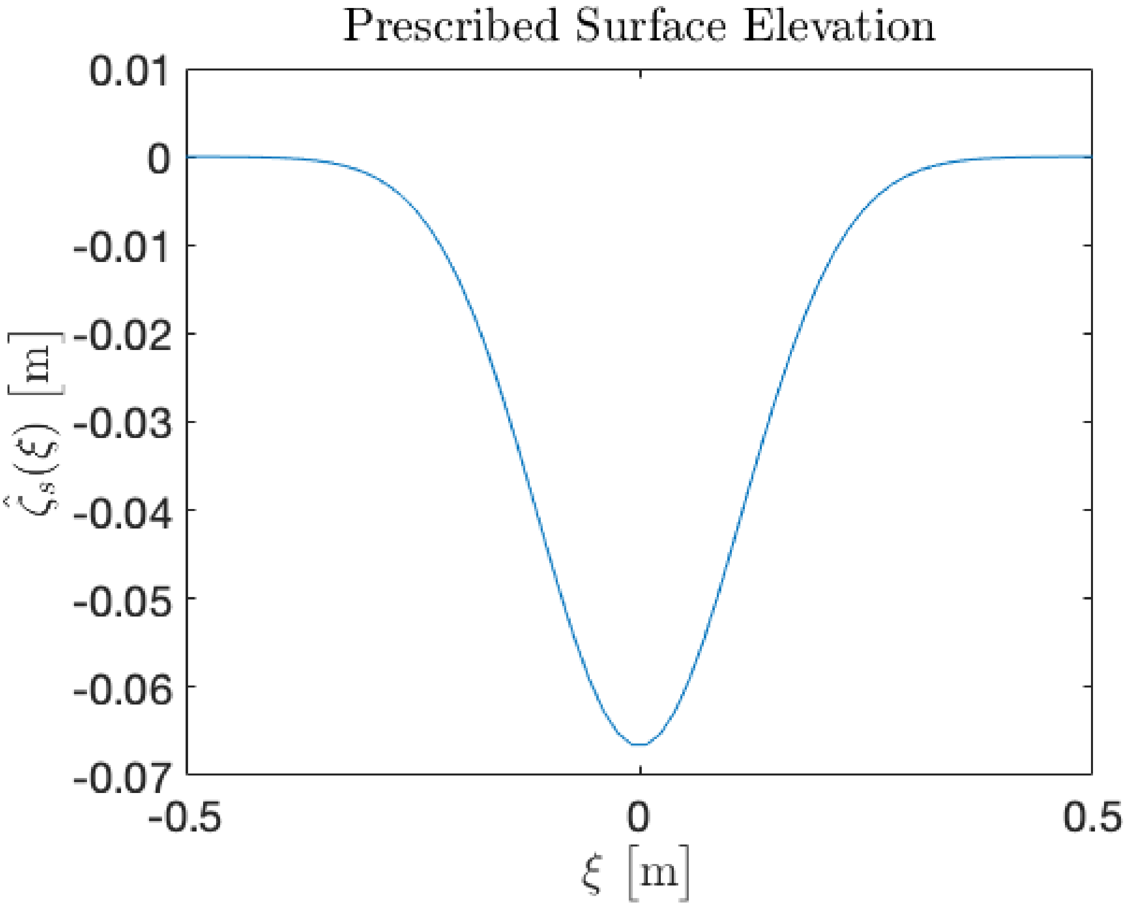


Figure 5: Prescribed surface elevation $\hat{\zeta}_s$ for the 2D cases.

Hence, the velocity of the ship V can be determined using the Froude number $Fr = \frac{V}{\sqrt{gL}}$. Furthermore, the z-vector is initialized using the height Froude number $Fr_h = \frac{V}{\sqrt{gh}}$ and by the number of points in z-direction which is set to be $N_z = 200$ for all the cases.

To illustrate that the effects of a non-linear depth-dependent shear flow, an exponential shear current profile, (48), is used in addition to the linear profile.

$$U(z) = \frac{gFr_s}{3V}(1 - e^{3z}) \tag{48}$$

Hence, the derivative evaluated at the surface is $U_0' = -\frac{gFr_s}{V}$, which is the correct definition of the shear Froude number Fr_s. The depth-dependent shear profile is illustrated in Figure (6). By choosing such a shear profile, the problem of critical layers is avoided, since the shear current velocity U never reaches the velocity of the moving sheep V. The exponential shear current is illustrated for a shear-assisted system in Figure (6).

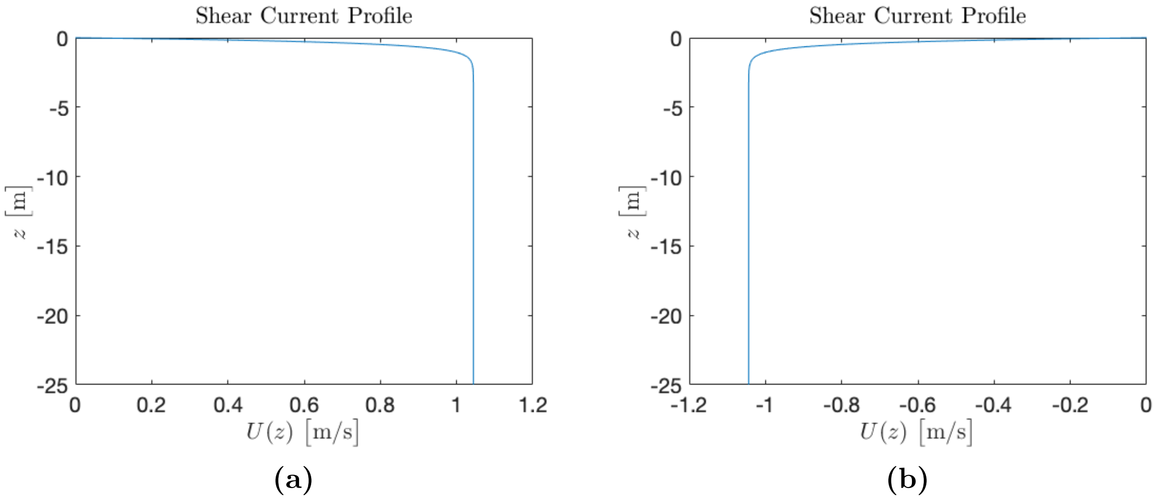


Figure 6: Shear current profile for the 2D case as seen from the water surface when $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ in a exponential shear profile in (a) a assisted system, (b) a inhibited system.

The radiation condition parameter ϵ is set to the relatively high $\epsilon = 2$. The reason for using a high ϵ is to make the problem feasible to calculate without the use of supercomputers. By choosing a high ϵ, the integral in the Green's function converges faster and the physical domain N_x don't have to be unreasonably big to avoid periodic boundary condition issues. However, such a big ϵ is not physically correct and will alter some wave characteristics. These issues will be discussed further in section (5).

3.2.3 3D

Similar to the 2D case, a elliptical Gaussian distribution is used with a longitudinal length $L = 1$ and a width $W = L \cdot AR$. An aspect ratio of $AR = 0.25$ is used for all 3D cases done in this thesis. Thus, the resulting prescribed surface elevation is as shown in Figure 7. It is described by equation (49).

$$\hat{\zeta}_s = -\frac{1}{15}\mathrm{e}^{-((\frac{\pi}{l}\xi_x)^2+(\frac{\pi}{w}\xi_y)^2)} \tag{49}$$

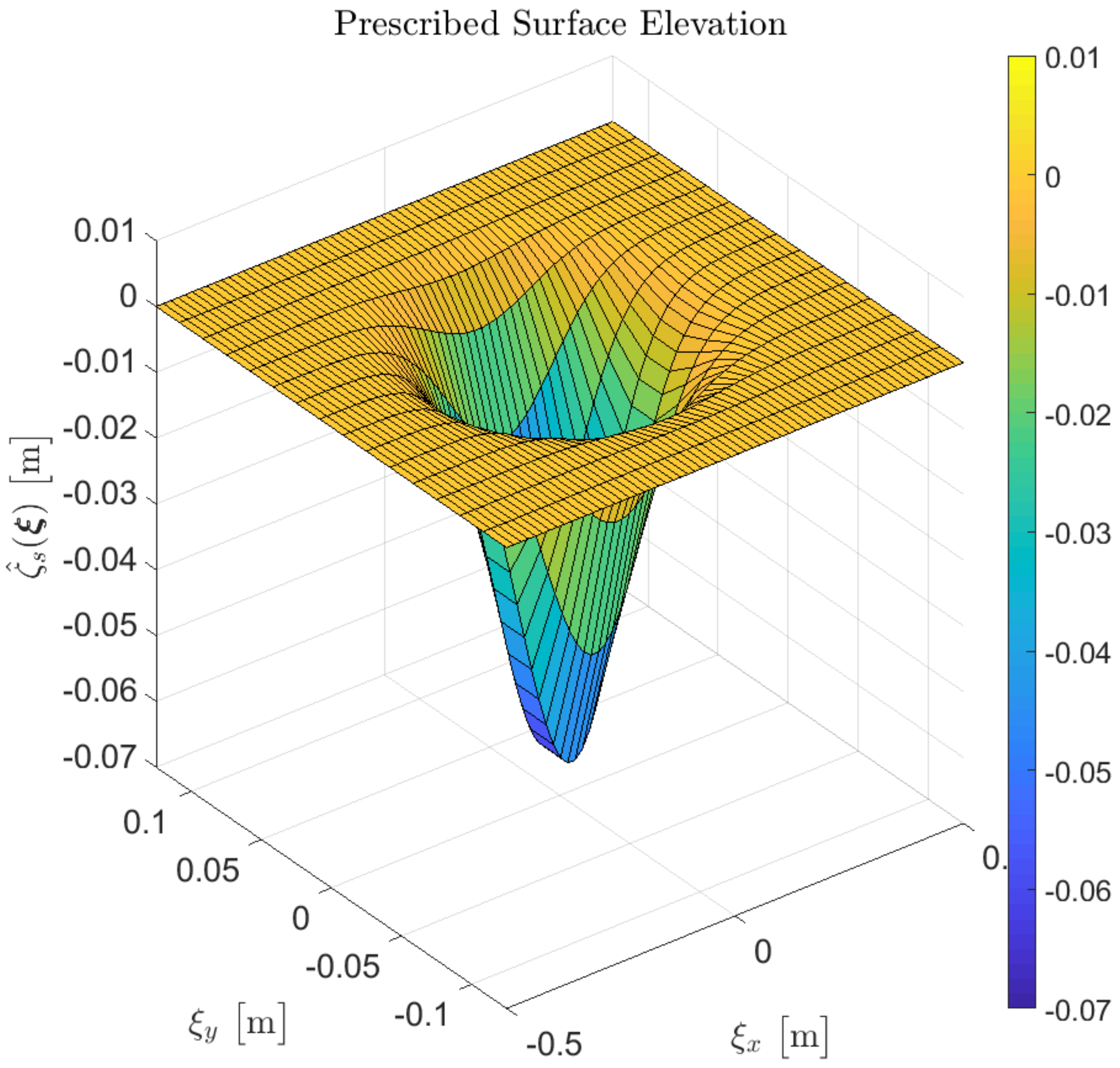


Figure 7: Prescribed surface elevation $\hat{\zeta}_s$ for the 3D cases.

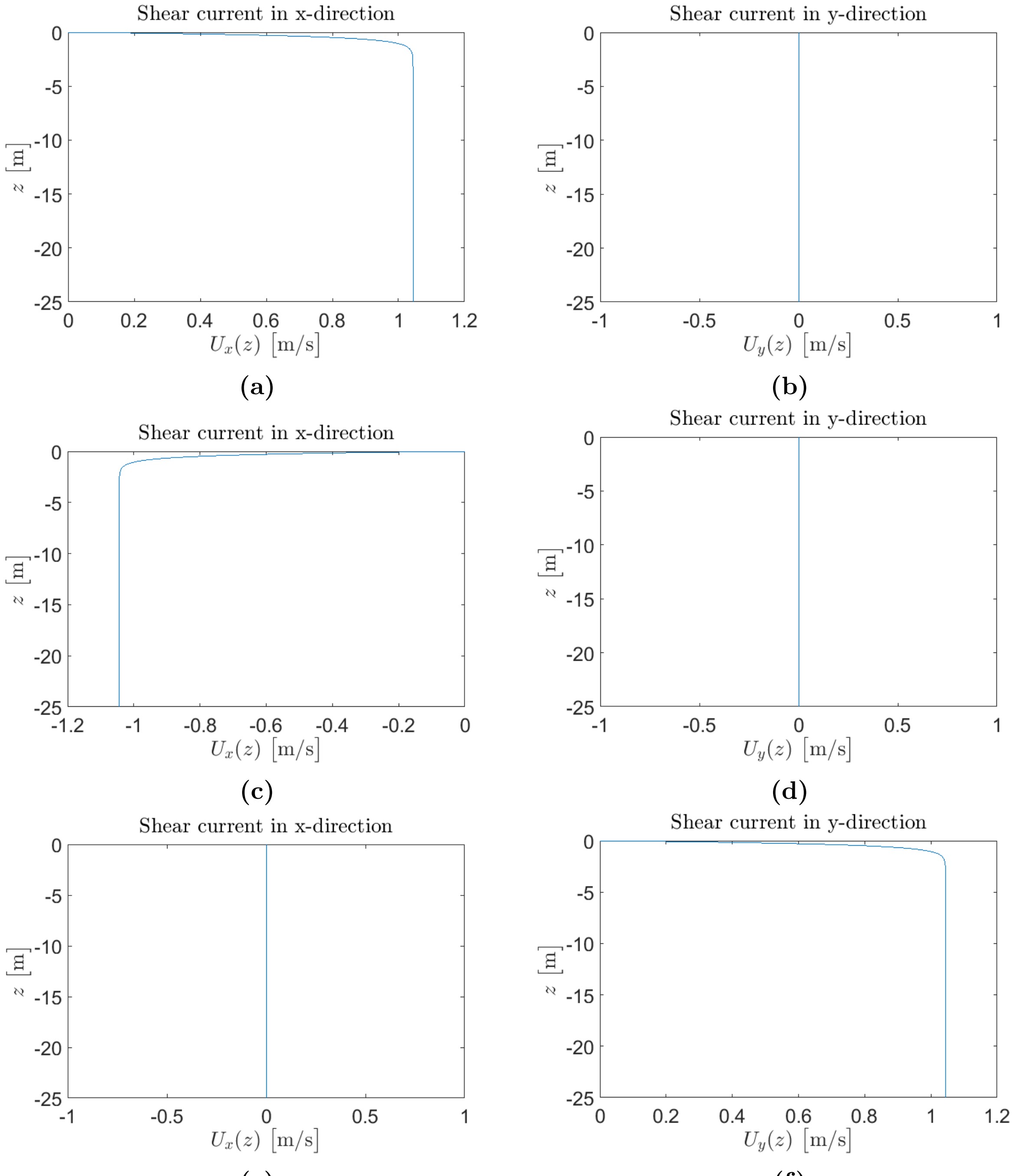
The velocity of the ship $|\boldsymbol{V}|$ is determined from the Froude number $Fr = \frac{|\boldsymbol{V}|}{\sqrt{gL}}$. As for the 2D case, the z-vector is initialized using the height Froude number $Fr_h = \frac{|\boldsymbol{V}|}{\sqrt{gh}}$ and by setting $N_z = 200$.

The same shear current profile is used for the 3D case as the 2D case is used, only the direction of the shear current is chosen to be normal to the direction of the moving ship. If the ship moves in the x-direction, the shear current can be described by the following equations

$$U_x(z) = 0 \tag{50a}$$

$$U_y(z) = \frac{gFr_s}{3V}(1 - e^{3z}) \tag{50b}$$

Furthermore, this shear current profile maintains the relationship $Fr_s = \frac{|\boldsymbol{U}_0||\boldsymbol{V}|}{g}$. The shear current profile is illustrated in Figure (8).



(a) **(b)** **(c)** **(d)** **(e)** **(f)**

Figure 8: Shear current profile for the 3D case as seen from the water surface when $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ in a exponential shear profile in (a,b) a assisted system, (c,d) a inhibited system, (e,f) a side-on system.

As for the 2D case, the radiation condition parameter ϵ is set to be $\epsilon = 2$. The argument that these calculations are computationally expensive is especially important in 3D. It would not be possible to use this numerical tool for low ϵ and get accurate results without the use of supercomputers.

3.3 Verification

3.3.1 The Forward Problem

A reasonable test for checking the resulting pressure patch is to use it in the forward problem and look at the resulting wave pattern and ship result. The discrete forward problem is solved using a fast Fourier transform (FFT) method, which is a relatively quick method for calculating a discrete Fourier transform. Initially, the FFT scheme is used on the external pressure $\hat{\tilde{p}}$ to find the Fourier space external pressure $\tilde{p}$. Then an inverse FFT method is used to calculate the surface elevation. More specifically, the MATLAB functions fft() in the 2D case, and fft2() in the 3D case, are used for discrete Fourier transform of the pressure. While the ifft(), and ifft2() are used for inverse discrete Fourier transform to find the surface elevation.

The discrete forward problem in 2D can be formulated as a discrete Fourier transform.

$$\hat{\zeta}_k = \frac{1}{N}\sum_{n=1}^{N}\frac{\tilde{p}_n k_n \mathrm{tanh}(k_n h)\mathrm{e}^{\frac{2\pi i}{N}(n-1)(k-1)}}{(\Delta_R)_n} \tag{51}$$

where

$$\begin{aligned}
(\Delta_R)_n &= (1+(I_g)_n)(k_n V)^2 + (V k_n U_0' - g k_n)\mathrm{tanh}(k_n h) + i\epsilon\Phi_n \\
\Phi_n &= 2V k_n(1+(I_g)_n) + U_0'\mathrm{tanh}(k_n h) \\
(I_g)_n &= \int_{-h}^{0} dz \frac{U''\mathrm{sinh}(k_n(z+h))}{k_n(U-V)\mathrm{cosh}(k_n h)}\frac{w_n}{(w_0)_n}
\end{aligned}$$

The 3D discrete forward problem can be formulated as

$$\hat{\zeta}_{t,k} = \frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{n=1}^{N}\frac{\tilde{p}_{m,n}k_{m,n}\mathrm{tanh}(k_{m,n}h)\mathrm{e}^{\frac{2\pi i}{M}(m-1)(t-1)}\mathrm{e}^{\frac{2\pi i}{N}(n-1)(k-1)}}{(\Delta_R)_{m,n}} \tag{52}$$

where

$$\begin{aligned}
(\Delta_R)_{m,n} &= (1+(I_g)_{m,n})((k_x)_{m,n}V_x + (k_y)_{m,n}V_y)^2 \\
&\quad + \left(\frac{((k_x)_{m,n}V_x + (k_y)_{m,n}V_y)((k_x)_{m,n}U_{x,0}')}{k_{m,n}} - g k_{m,n}\right)\mathrm{tanh}(k_{m,n}h) + i\epsilon\Phi_{m,n} \\
\Phi_{m,n} &= 2((k_x)_{m,n}V_x + (k_y)_{m,n}V_y)(1+(I_g)_{m,n}) + \boldsymbol{U}_0'\mathrm{tanh}(k_{m,n}h) \\
(I_g)_{m,n} &= \int_{-h}^{0} dz \frac{(\boldsymbol{k}_{m,n}\cdot\boldsymbol{U}'')\mathrm{sinh}(k_{m,n}(z+h))}{k_{m,n}(\boldsymbol{k}_{m,n}\cdot\boldsymbol{U} - \boldsymbol{k}_{m,n}\cdot\boldsymbol{V})\mathrm{cosh}(k_{m,n}h)}\frac{w(\boldsymbol{k}_{m,n},z)}{w(\boldsymbol{k}_{m,n},0)}
\end{aligned}$$

The difference between the wanted surface elevation in the area of the ship and the actual surface elevation is a natural way to verify the pressure patches. Moreover, the mean relative error defined as

$$\epsilon_s = \frac{\max(|\hat{\zeta} - \hat{\zeta}_s|)}{\max(|\hat{\zeta}_s|)} \tag{53}$$

will be used as verification for the different cases in this thesis.

3.3.2 Green's function

As the pressure patch is highly dependent on the Green's function, it is important to investigate the function in order to verify that the resulting pressure patch is in fact correct. More specifically, a convergence test concerning both the integral limits and the Fourier spacing Δk is conducted. Since the k-values used for the integration in the Green's function are the same as the ones used in a FFT scheme, the only way to increase the integration limits is to decrease the distance between points in real space, $\Delta\xi_x$ and $\Delta\xi_y$. Hence, the convergence of the function regarding the integral limits is tested by increasing the number of points in the region of the ship N_{ship}, making the spacing in real space smaller. The convergence is measured as the difference in the solution for the current N_{ship} and some presumed correct solution with a relatively high N_{ship}. For 2D, the presumed correct solution was the value obtained when setting $N_{ship} = 2^{10}$, In 3D, $N_{ship} = 2^7$ was used as the reference point. The solution obtained for both the 2D and 3D case is not, however, expected to be fully converged for all the points. But due to the computational expensiveness of the problem, these parameters are reasonable to get an estimate of the convergence. The Green's function is as mentioned in section (2.1.4) and (2.2.4), not convergent in the point $\xi = 0$, and slowly convergent close to this point.

The integrand of the Green's function is highly oscillating for high values of ξ, as seen in figure (3) and (4). Therefore, it is important to have sufficiently small spacing in Fourier space making Δk small. This can be done by increasing the number of points in the domain. Similar to the integral limits, the convergence test regarding Δk is measured by the difference in value between the current number of points in the domain N_x and some presumed correct solution with a relatively high N_x.

3.3.3 Verifying I_g

I_g is used in both the forward problem and the inverse problem when an arbitrary depth-dependent shear profile is present. Hence, the solution of this integral plays an important role in the results. As the parameter N_z is not used for other calculations other than for I_g, the verification of I_g is important when determining N_z. Similar to the convergence test of the Green's function, a reference value presumed to be correct was used to measure the convergence of the integral. However, since I_g is less computationally expensive than the Green's function, the error is defined as the mean difference of all the points in I_g.

$$\epsilon_{I_g} = \text{mean}(|I_g - I_{g,final}|) \tag{54}$$

The different size parameters will be set to a suitable value determined by the convergence test of the Green's function. Furthermore, the reference value used as the presumed correct solution $I_{g,final}$ is solution obtained when setting $N_z = 300$ for both the 2D and 3D case.

3.3.4 Wavelength

An additional test for the 2D results is to check to measure the resulting wavelength and compare it to the theoretical wavelength. Since the result of the inverse problem are, in most cases considered in combination with solving the forward problem, some validation for the forward problem is useful. The wavelength is measured at the points indicated in Figure 9 for all the 2D cases.

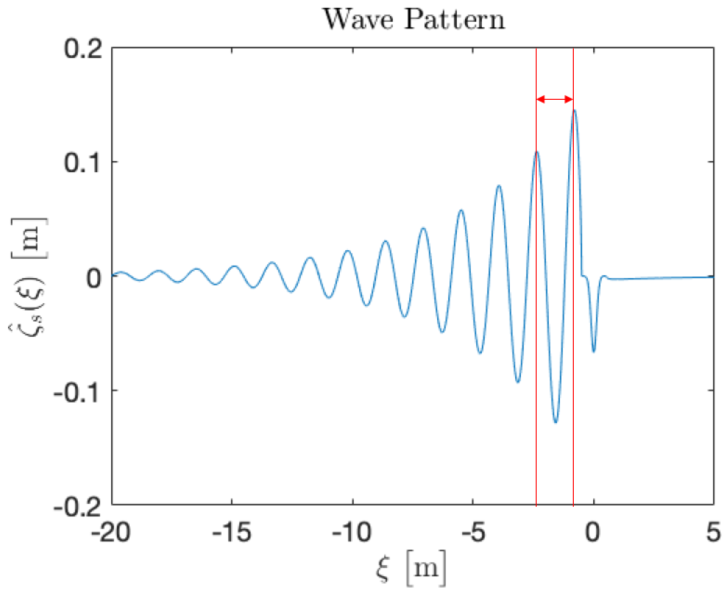
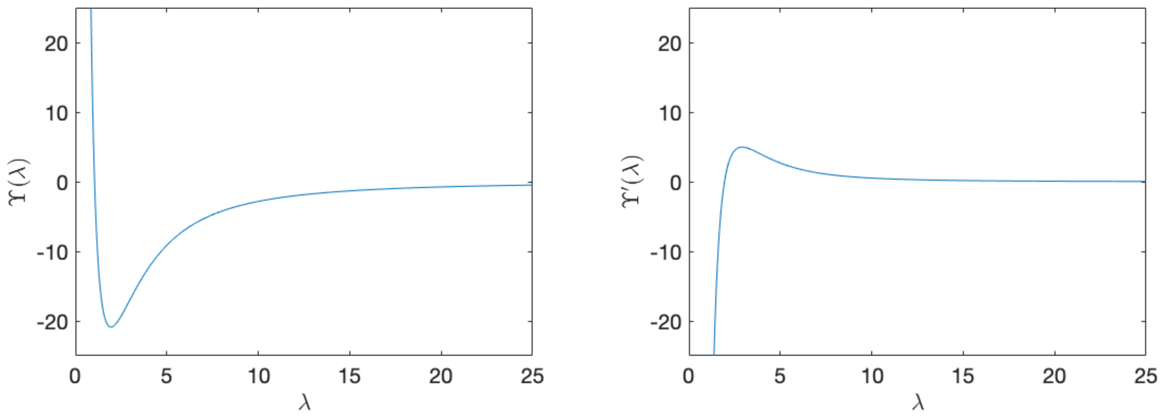


Figure 9: A figure showing how the wavelength is measured.

The theoretical wavelength is found using the implicit dispersion relation $\Delta_R = 0$. Moreover, the numerical method of Newton-Raphson is used to obtain the wavelength. The scheme can be expressed as follows.

$$\lambda^{(n+1)} = \lambda^{(n)} - \frac{\Delta_R(\lambda^{(n)})}{\Delta_R'(\lambda^{(n)})} \tag{55}$$



(a) **(b)**

Figure 10: Plots showing the implicit dispersion relation $\Upsilon(\lambda) = \Delta_R(\lambda)$ and its derivative $\Upsilon'(\lambda) = \Delta'_R(\lambda)$ in the case of $Fr = 0.5$, $Fr_h = 0.1$, and $Fr_s = 0.5$.

Due to the sloping of the implicit dispersion relation $\Delta_R(\lambda)$, a high initial guess for the wavelength λ will make the solution diverge. Hence, a low initial guess of $\lambda = 0.01$ is made for all the different cases. As seen in Figure (10), this low initial guess will make the solution converge to the right solution.

3.3.5 The inverse problem

A way to test the inverse problem solver is to use the numerical method to find a pressure patch for a surface depression where the pressure patch is already known. More specifically, given a Gaussian pressure $\hat{p}_{ext}$, the surface elevation in the region of the ship will be one of the plots in section 4.1. Thus, if one of these plots are given as the wanted surface elevation in the area of the ship, the resulting pressure should be precisely the Gaussian pressure.

4 Results

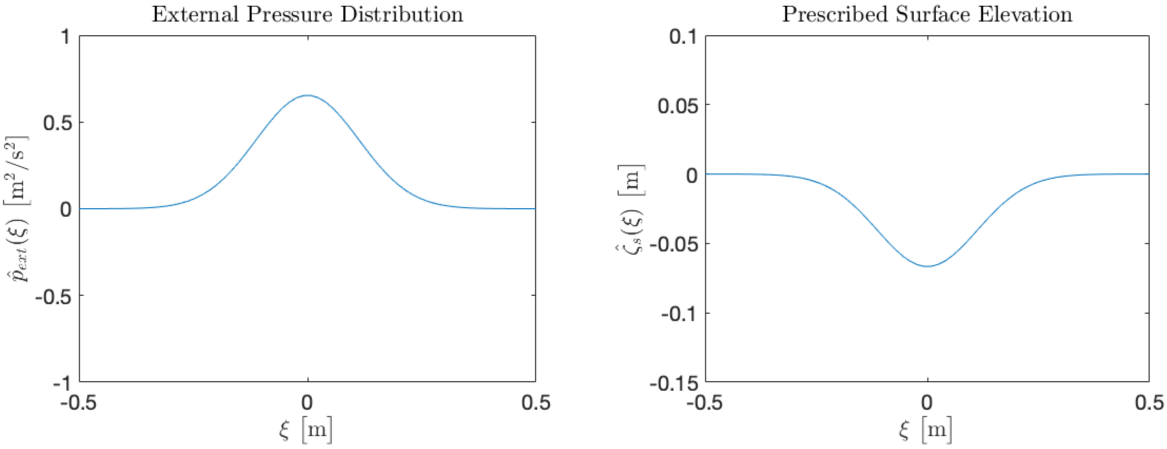
4.1 Constant Pressure Source

Part of the motivation for this thesis is that using the same pressure source for different conditions won't produce the same effective ship hull on the water surface. Hence, this section is included to show the effect of changing the conditions.

4.1.1 2D

A regular Gaussian pressure distribution is used as the pressure patch, as shown in Figure 11a. This gives the right surface elevation in the area of the ship, as shown in Figure

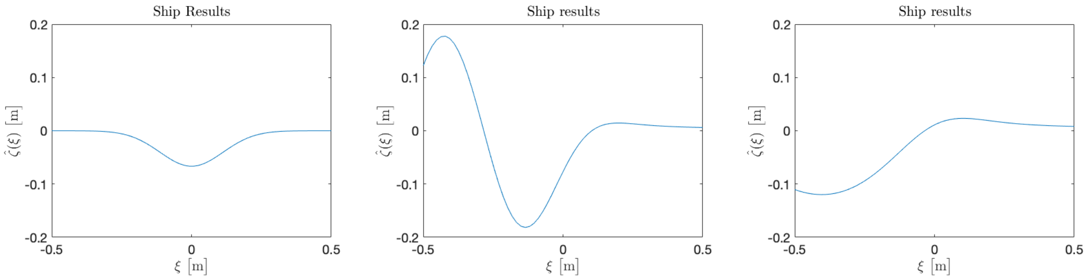
11b, when $Fr = 0$.



(a) **(b)**

Figure 11: Plots showing (a) the regular Gaussian pressure patch used, $\hat{p}_{ext} = -g\hat{\zeta}_s$ (b) the wanted surface elevation in the area of the ship in 2D $\hat{\zeta}_s = -\frac{1}{15}e^{-(\frac{\pi}{l}\xi)^2}$.

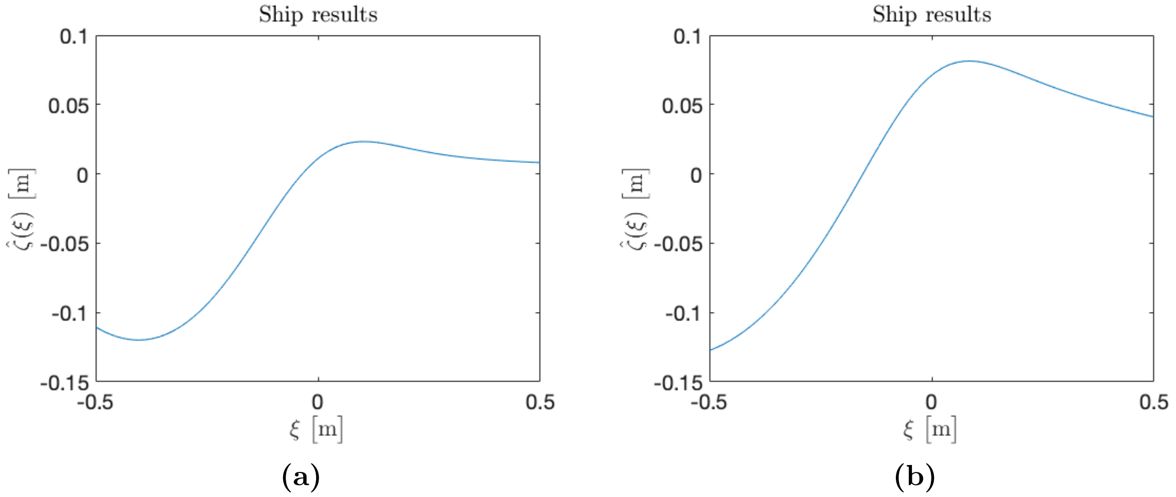
The effect of increasing the velocity is shown in Figure 12. As seen, the ship velocity changes the surface elevation in the region of the ship significantly.



(a) **(b)** **(c)**

Figure 12: Plots showing the effect of increasing the velocity for a constant pressure source. The plots are taken for $Fr_h = 0.05$, $Fr_s = 0$, and (a) $Fr = 0$, (b) $Fr = 0.3$, (c) $Fr = 0.5$.

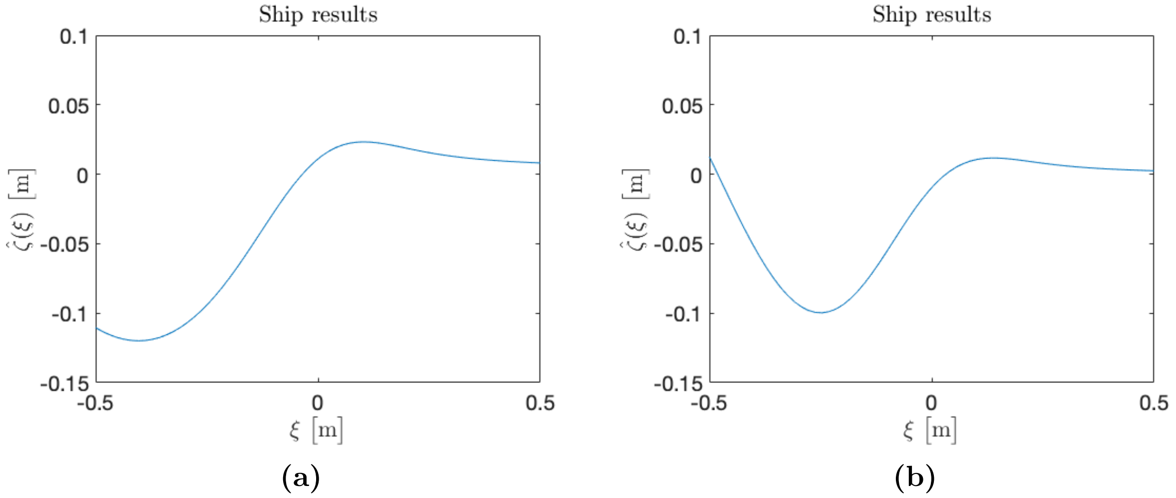
Also, changing the water depth effects the resulting effective ship hull. The effect of purely decreasing the water depth is shown in Figure 13.



(a) **(b)**

Figure 13: Plots showing the effect of decreasing the water depth for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_s = 0$, and (a) $Fr_h = 0.05$, (b) $Fr_h = 0.9$.

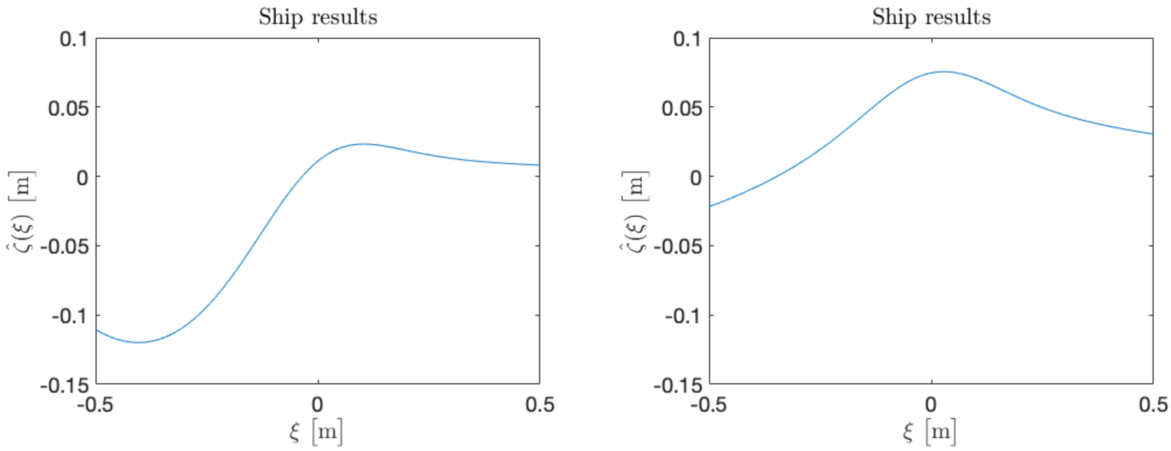
In a shear-assisted system, the wavelength is shortened. This also affects the effective ship hull for a constant pressure source. The effect of increasing the shear strength in a shear-assisted system is shown in Figure 14.



(a) **(b)**

Figure 14: Plots showing the effect of increasing the shear strength in a shear-assisted system for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_h = 0.1$, and (a) $Fr_s = 0$, (b) exponential shear profile with $Fr_s = 0.5$.

Likewise, the wavelength is lengthened in a shear-inhibited system. In addition, the shear current profile seems to raise the surface elevation in the region of the ship. The effect of increasing the shear strength in a shear-inhibited system is shown in Figure 15.

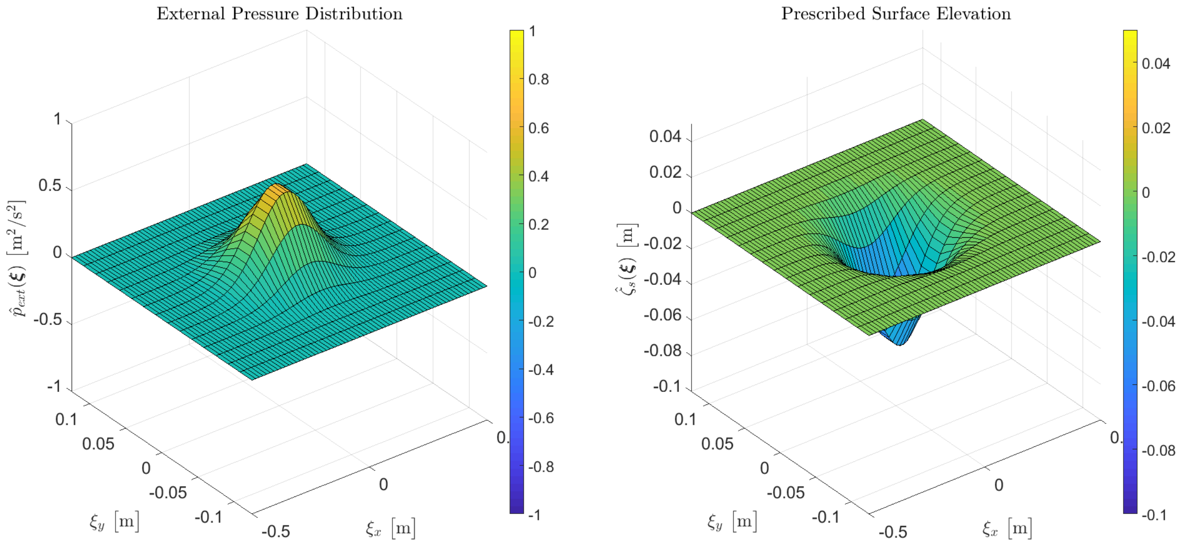


(a) **(b)**

Figure 15: Plots showing the effect of increasing the shear strength in a shear-inhibited system for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_h = 0.1$, and (a) $Fr_s = 0$, (b) exponential shear profile with $Fr_s = 0.5$.

4.1.2 3D

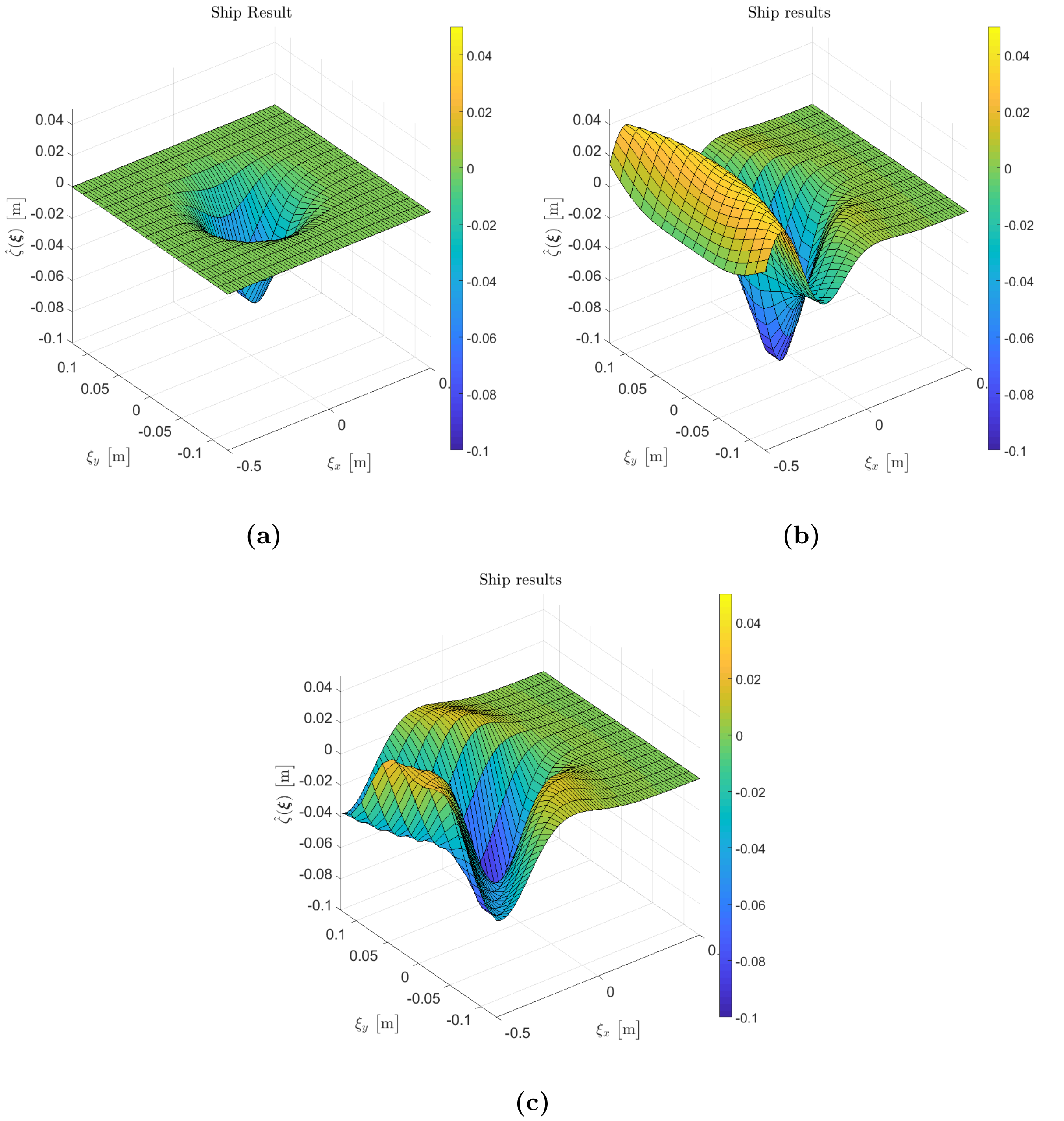
A regular Gaussian pressure distribution is used as the pressure patch, as shown in Figure 16a. This gives the right surface elevation in the area of the ship, as shown in Figure 16b, when $Fr = 0$.



(a) **(b)**

Figure 16: Plots showing (a) the regular Gaussian pressure patch used, $\hat{p}_{ext} = -g\hat{\zeta}_s$ (b) the wanted surface elevation in the area of the ship in 3D $\hat{\zeta}_s = -\frac{1}{15}\mathrm{e}^{-((\frac{\pi}{l}\xi_x)^2+(\frac{\pi}{w}\xi_y)^2)}$.

The effect of increasing the velocity is shown in Figure 17. As for the 2D case, the ship velocity has a significant effect on the surface elevation in the region of the ship.



(a)

(b)

(c)

Figure 17: Plots showing the effect of increasing the velocity for a constant pressure source. The plots are taken for $Fr_h = 0.1$, $Fr_s = 0$, and (a) $Fr = 0$, (b) $Fr = 0.3$, (c) $Fr = 0.5$.

While decreasing the water depth in 2D had a noticeable effect on the surface elevation in the region of the ship, the effect of decreasing the water depth in 3D is almost negligible. This can be seen in Figure 18.

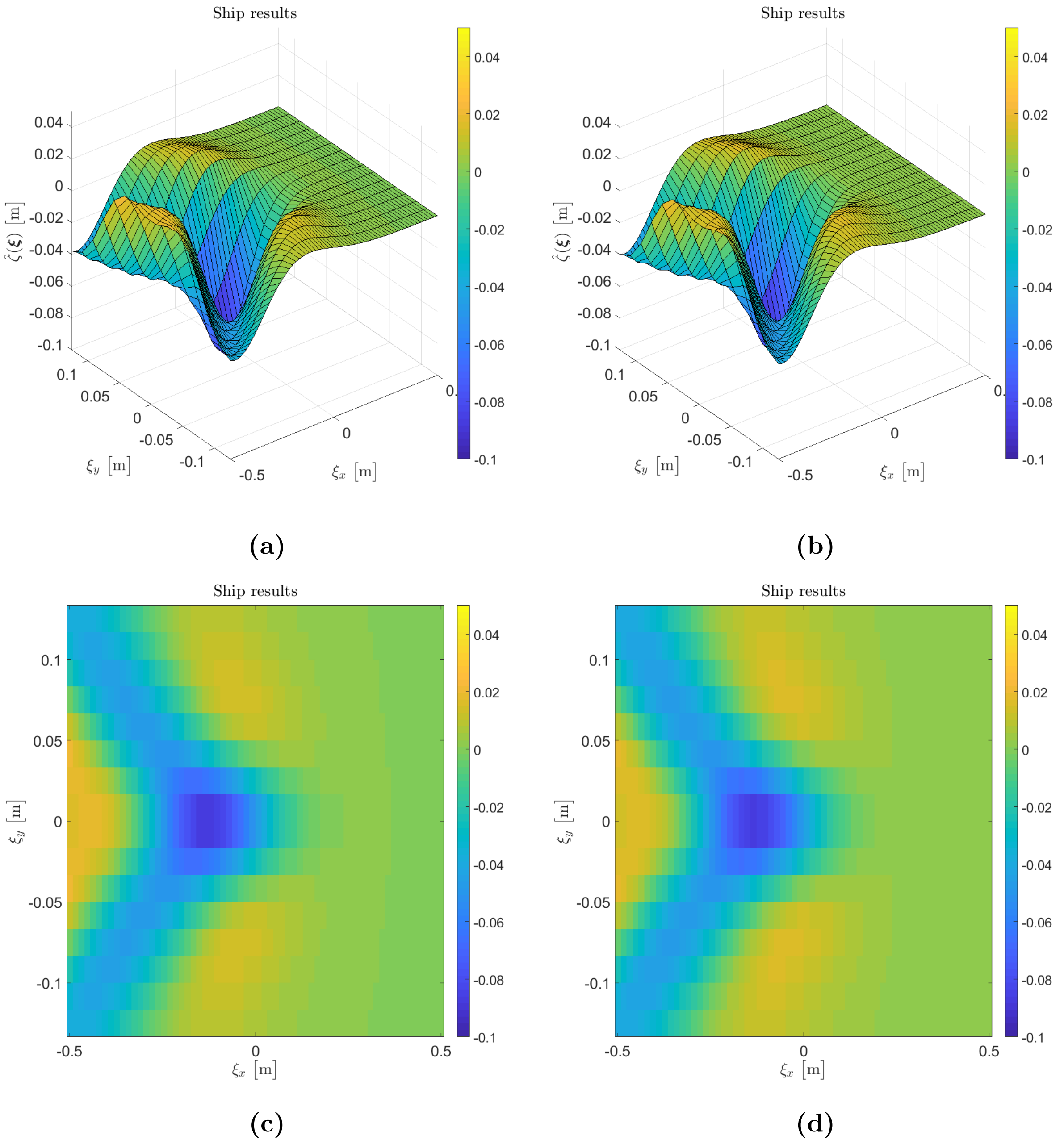
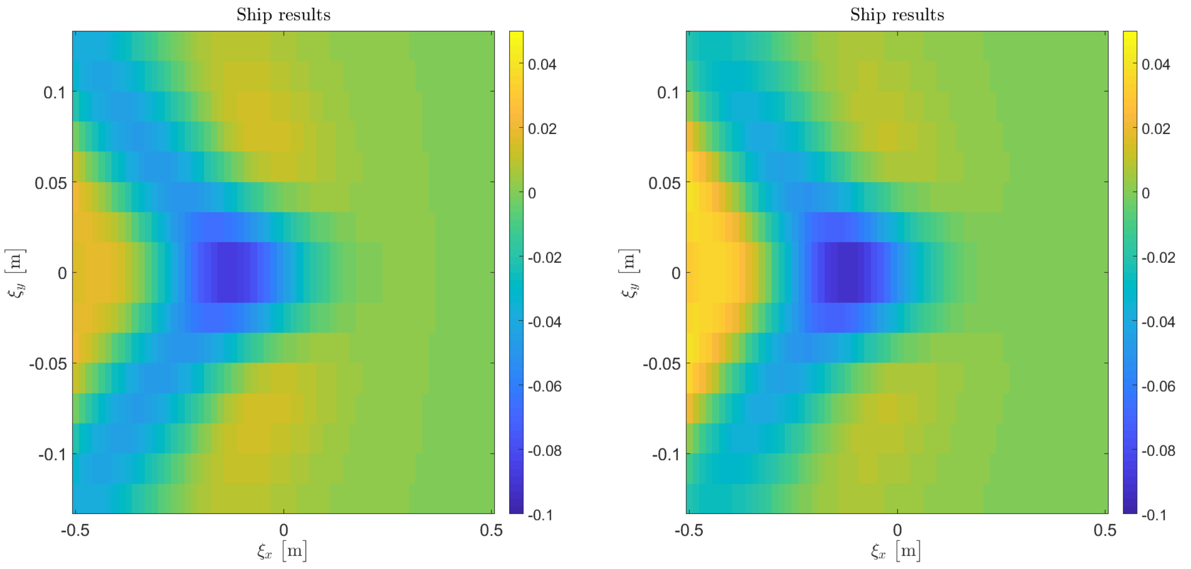


Figure 18: Plots showing the ship results from different water depths when a constant pressure source is used. (c) and (d) are scaled color images looking at the surface elevation from above The plots are taken for $Fr = 0.5$, $Fr_s = 0$, and (a,c) $Fr_h = 0.1$, (b,d) $Fr_h = 0.9$.

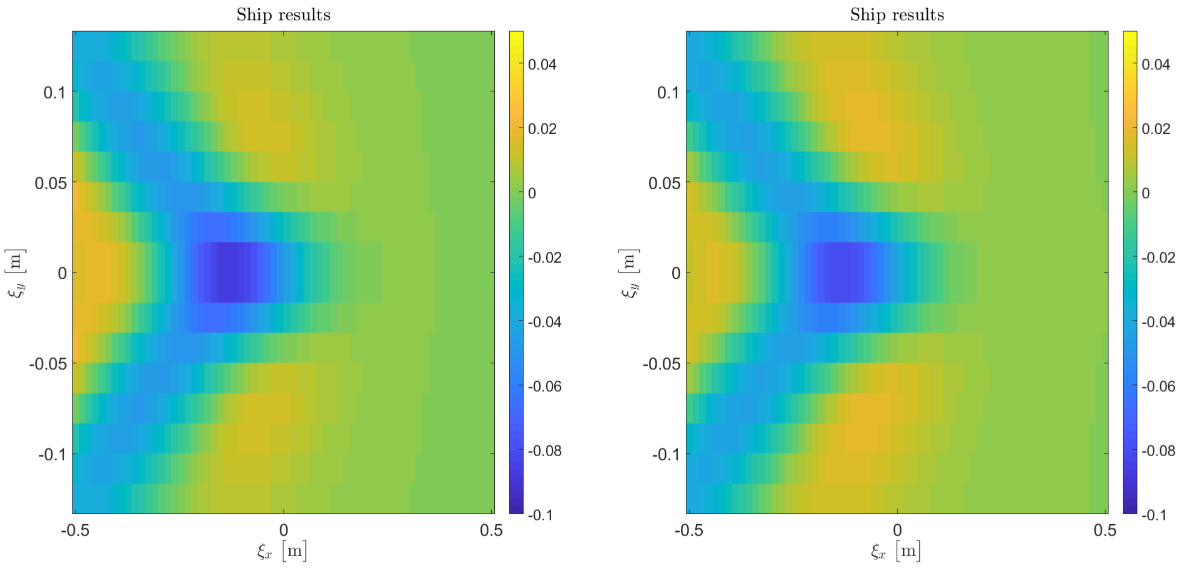
The effect of increasing the shear strength in a shear-assisted system is shown in Figure 19. Similar to 2D, the effect of introducing a shear-assisted profile is to shift the wave pattern towards the right. Thus the surface elevation in the region of the ship has higher a wave amplitude closer to the center of the pressure patch.



(a) **(b)**

Figure 19: Plots showing the effect of increasing the shear strength in a shear-assisted system for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_h = 0.1$, and (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential shear profile.

While a shear-assisted system increases the amplitude close to the center of the pressure patch, a shear-inhibited system decreases it. The effect of increasing the shear strength in a shear-inhibited system is shown in Figure 20.



(a) **(b)**

Figure 20: Plots showing the effect of increasing the shear strength in a shear-inhibited system for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_h = 0.1$, and (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential shear profile.

The effect of increasing the shear strength in a side-on shear system is shown in Figure 21. As seen, the resulting surface elevation in the region of the ship is shifted upwards in the direction of the shear current. Hence, an asymmetric ship result is created.

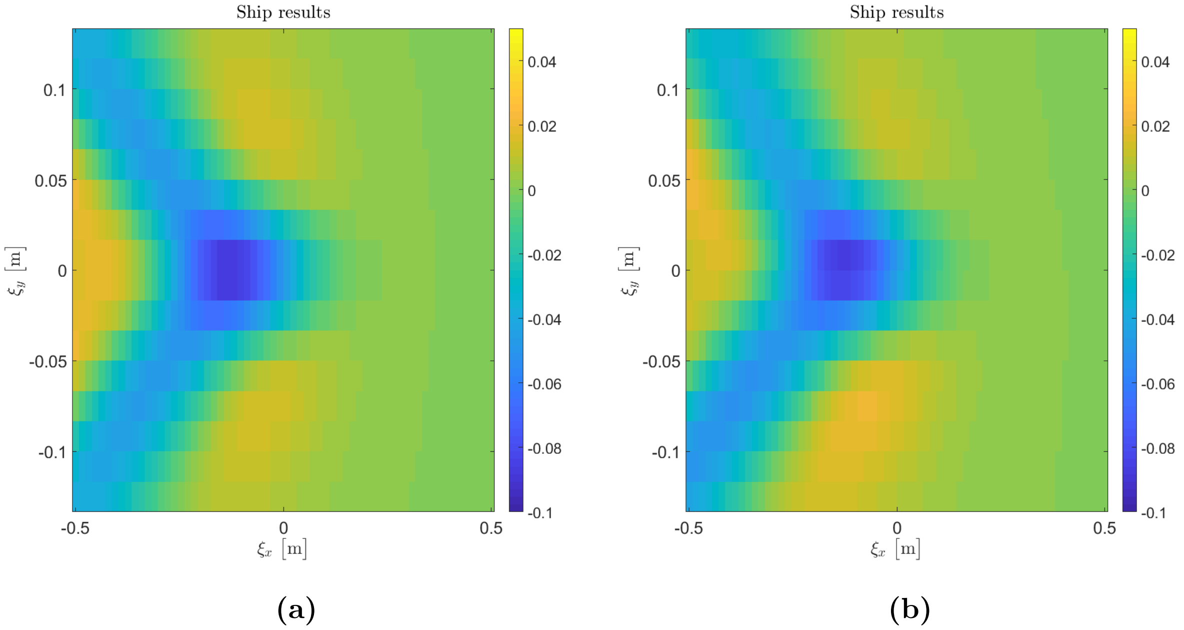


Figure 21: Plots showing the effect of increasing the shear strength in a side-on shear system for a constant pressure source. The plots are taken for $Fr = 0.5$, $Fr_h = 0.1$, and (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential shear profile.

4.2 Variable Pressure Source

The ship velocity, water depth, and shear current profile have a significant effect on the surface elevation in the area of the ship. Hence, these conditions are expected to affect the calculated pressure patch from the inverse problem. This section studies the effect of ship velocity, water depth, and shear profile on the resulting pressure patch.

First of all, when using the calculated pressure patch, the difference between the prescribed surface elevation and the resulting surface elevation in the area of the ship is almost imperceptible. For more information about the error of the result, see section 4.5. The resulting surface elevation in the region of the ship for both 2D and 3D are shown in Figure 22.

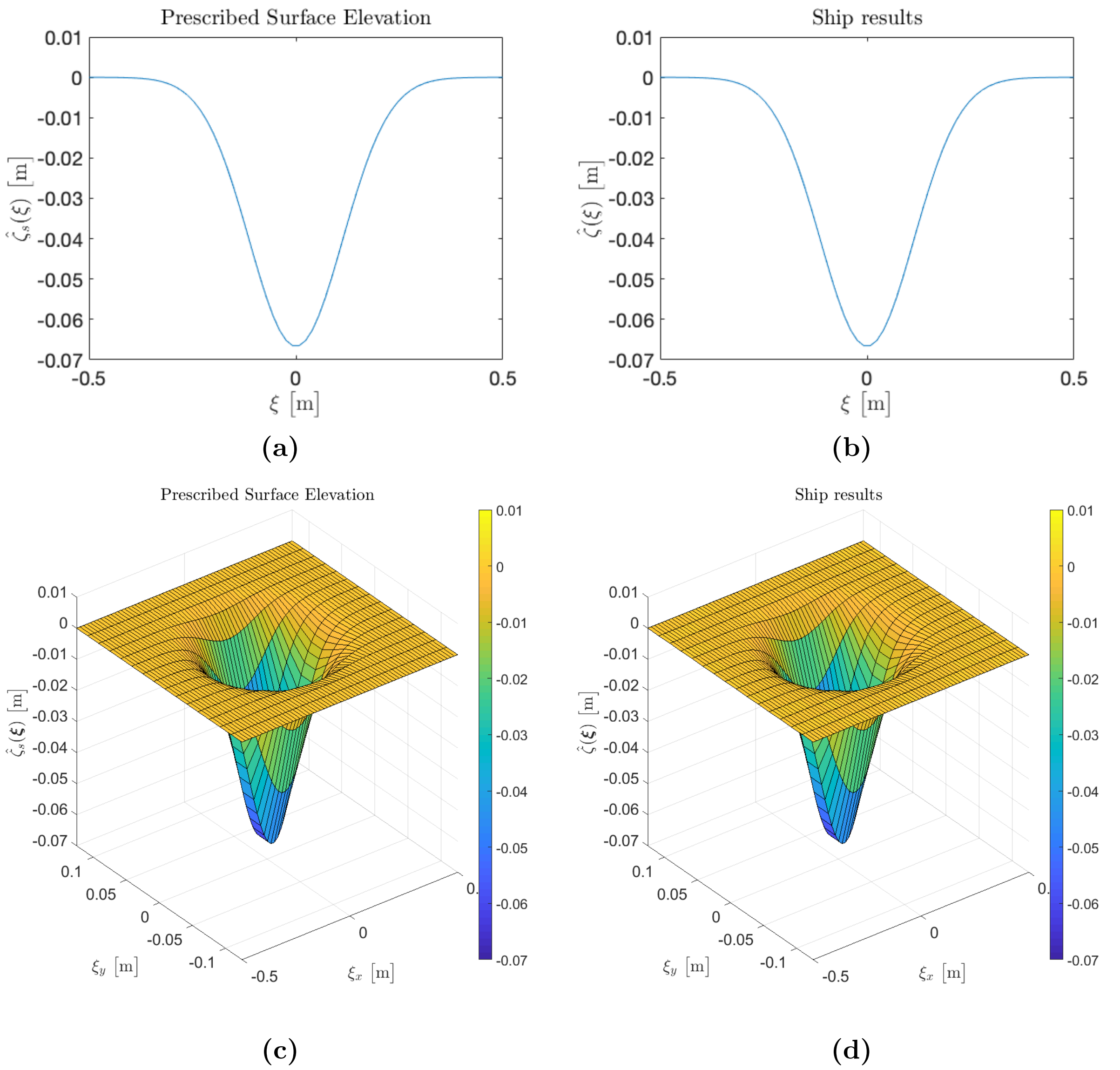
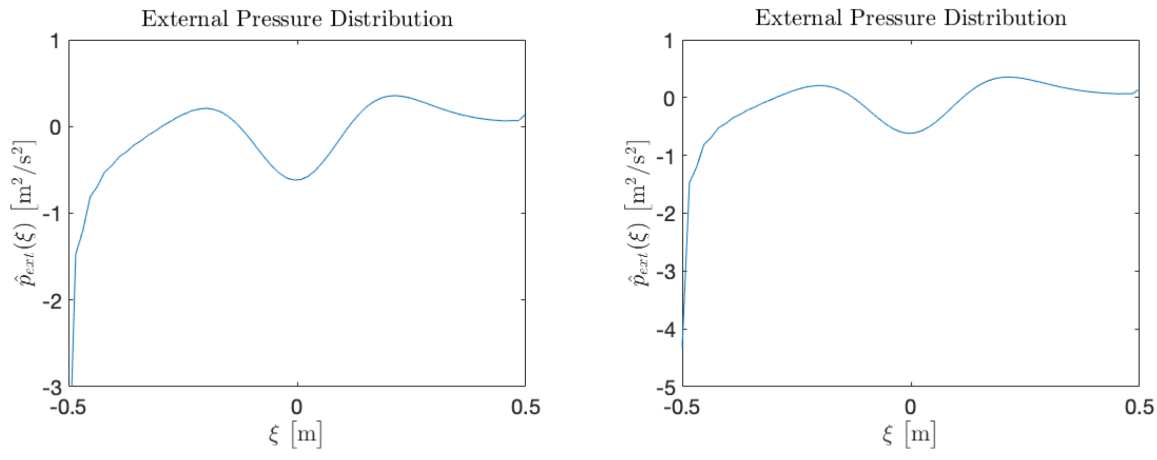


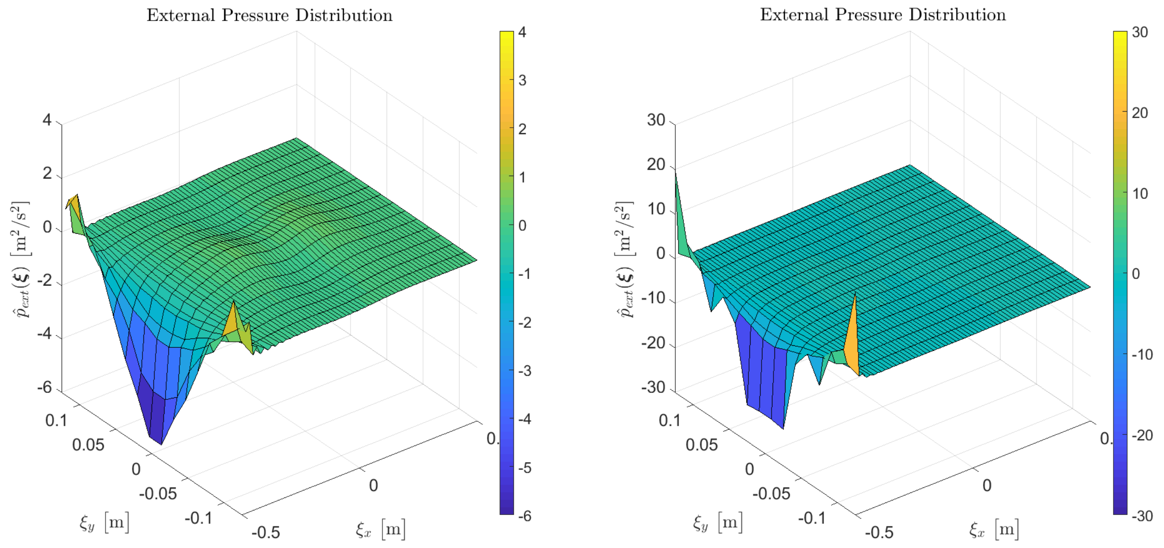
Figure 22: Plots showing the prescribed surface elevation for the (a) 2D case, (c) 3D case, and the ship result for (b) 2D case where $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$, (d) 3D case where $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$.

Second of all, the resulting pressure distribution is significantly different from the regular Gaussian pressure distribution. As seen in Figure 23 and 24, there are much bigger pressure variations away from the center of the patch. This can be seen as the pressure compensating for the slower reaction time of the water surface when the ship is moving at a certain velocity. Also, some pressure spikes are happening at the back of the pressure patches for both the 2D and 3D case, although the 3D pressure patches tend to have bigger spikes than the 2D patches. These pressure spikes cause some trouble when looking at the wave pattern created. This will be discussed further in section 5.



(a) **(b)**

Figure 23: Plots showing the resulting pressure patch when $Fr = 0.5$ $Fr_h = 0.1$, $Fr_s = 0$, (a) Regularized scaling of plot (b) Full plot.



(a) **(b)**

Figure 24: Plots showing the resulting pressure patch when $Fr = 0.5$ $Fr_h = 0.1$, $Fr_s = 0$. (a) Regularized scaling of plot (b) Full plot

Furthermore, the different conditions have similar effects on the calculated pressure patches as the ship results for a constant pressure source. The biggest effect, as for the ship result of with constant pressure patch, is the increase of ship velocity. Here, the increase in velocity causes bigger pressure variations throughout the pressure patch. Physically, this makes sense as more power is needed to compensate for shorter impact time. The effect can be seen in Figure 25 and 26. As the pressure patch values in the middle of the patch are of interest as well as the back, a regularized set of plot limits are used when illustrating the pressure patches. The same plot limits as in Figure 23a and 24a are used.

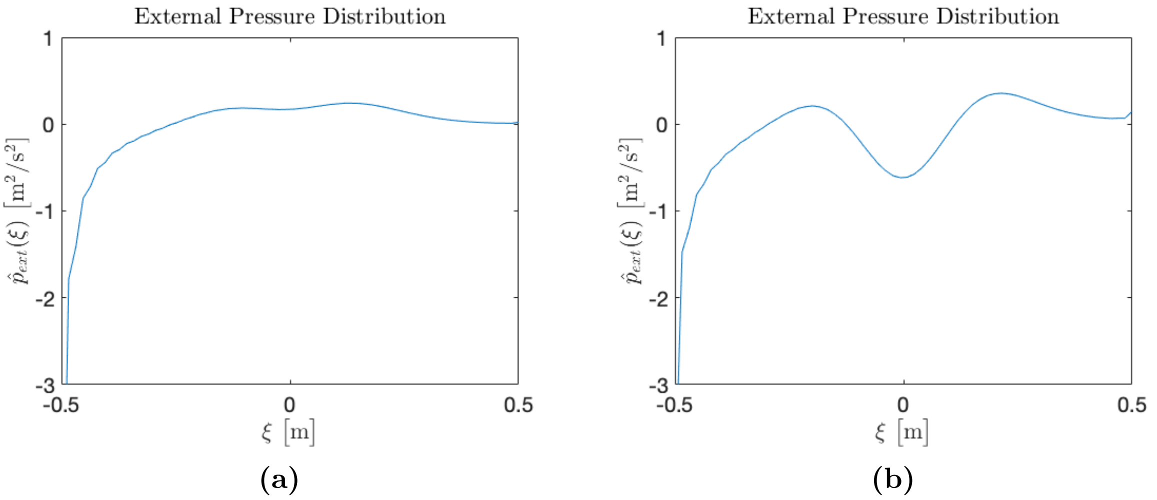


Figure 25: Plots showing the resulting pressure patch when $Fr_h = 0.1$, $Fr_s = 0$, (a) $Fr = 0.3$ (b) $Fr = 0.5$

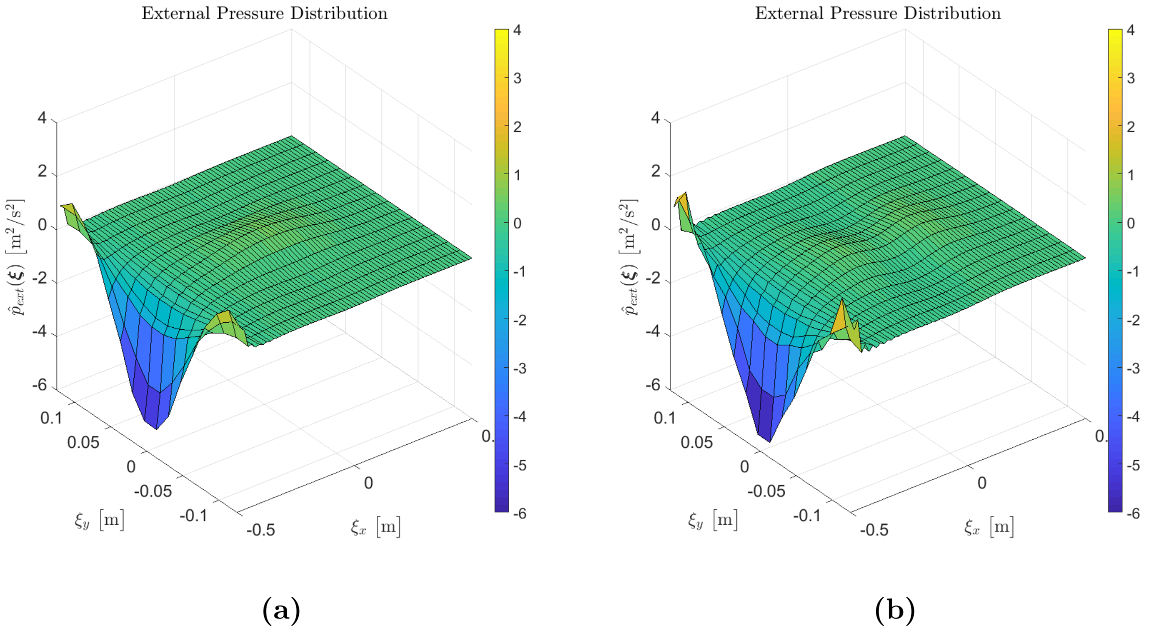
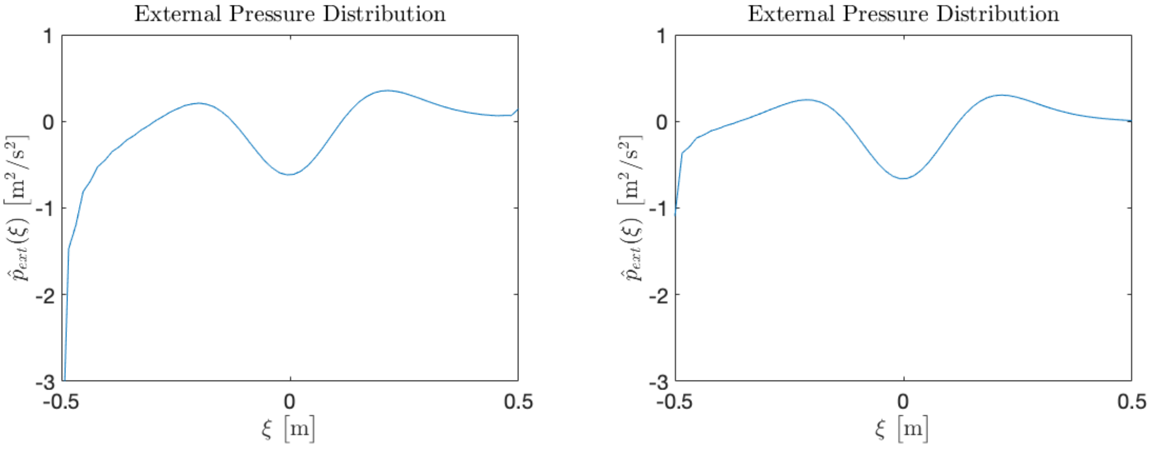


Figure 26: Plots showing the resulting pressure patch when $Fr_h = 0.1$, $Fr_s = 0$, (a) $Fr = 0.3$ (b) $Fr = 0.5$

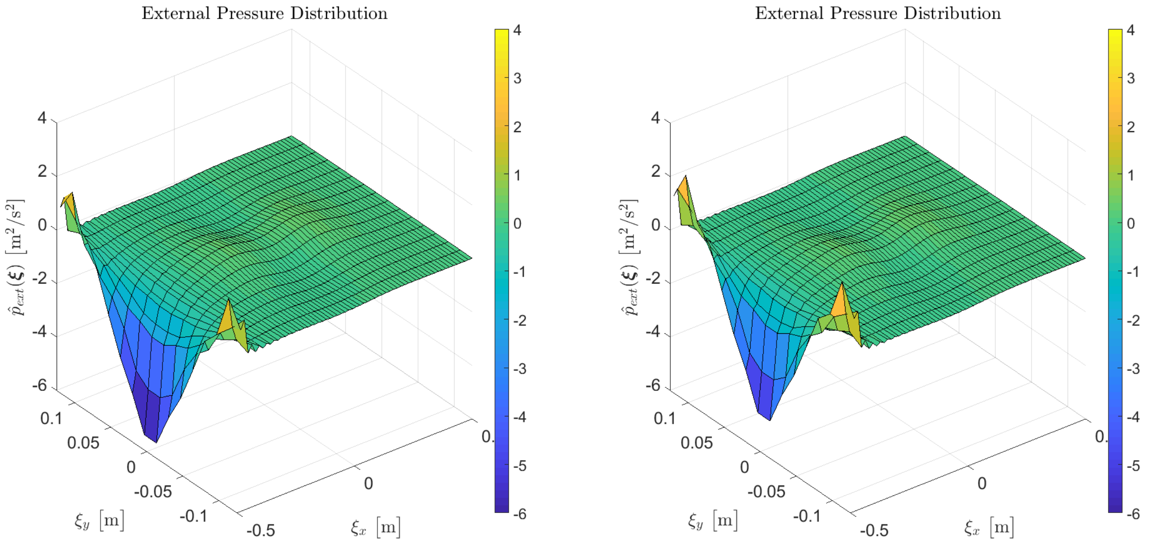
Similar to the ship result of a constant pressure patch, the effect of decreasing the water depth has a bigger impact on the 2D pressure patch than the 3D pressure patch. In 2D, the pressure patch is more symmetrical with a slightly raised left side, causing the pressure spike at the back of the patch to have a smaller magnitude. In 3D, the impact is imperceptible from looking at the regular external pressure distribution surface plot.

However, when looking at the full pressure patch, it is possible to see that also in 3D, the decrease of water depth dampens the pressure spikes. The regular external pressure distribution plots are shown in Figure 27 and 28, while the full pressure patch in the 3D case are shown in Figure 29.



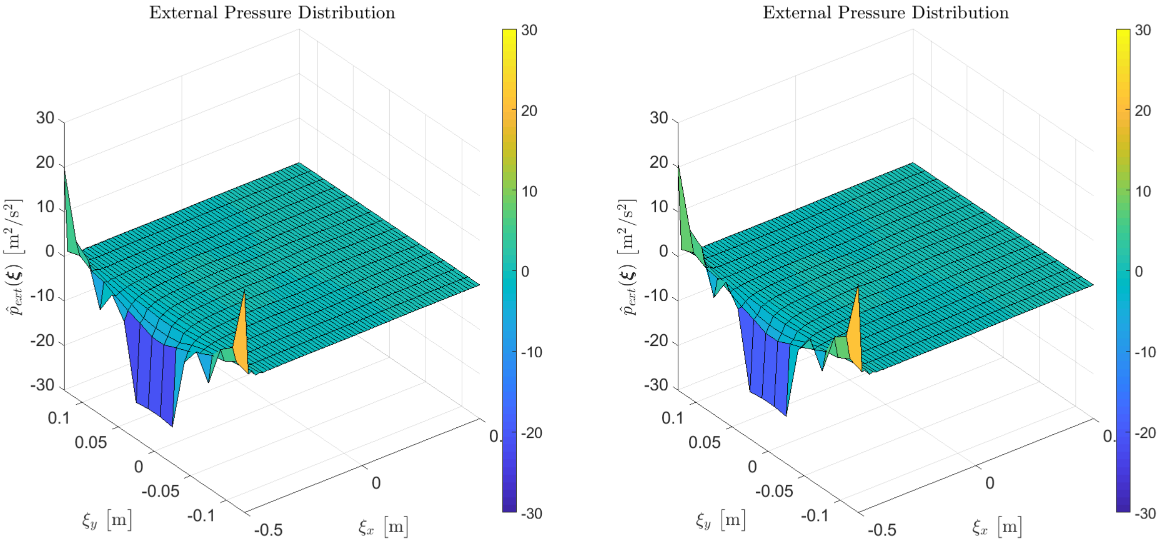
(a) **(b)**

Figure 27: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_s = 0$, (a) $Fr_h = 0.1$ (b) $Fr_h = 0.9$



(a) **(b)**

Figure 28: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_s = 0$, (a) $Fr_h = 0.1$ (b) $Fr_h = 0.9$

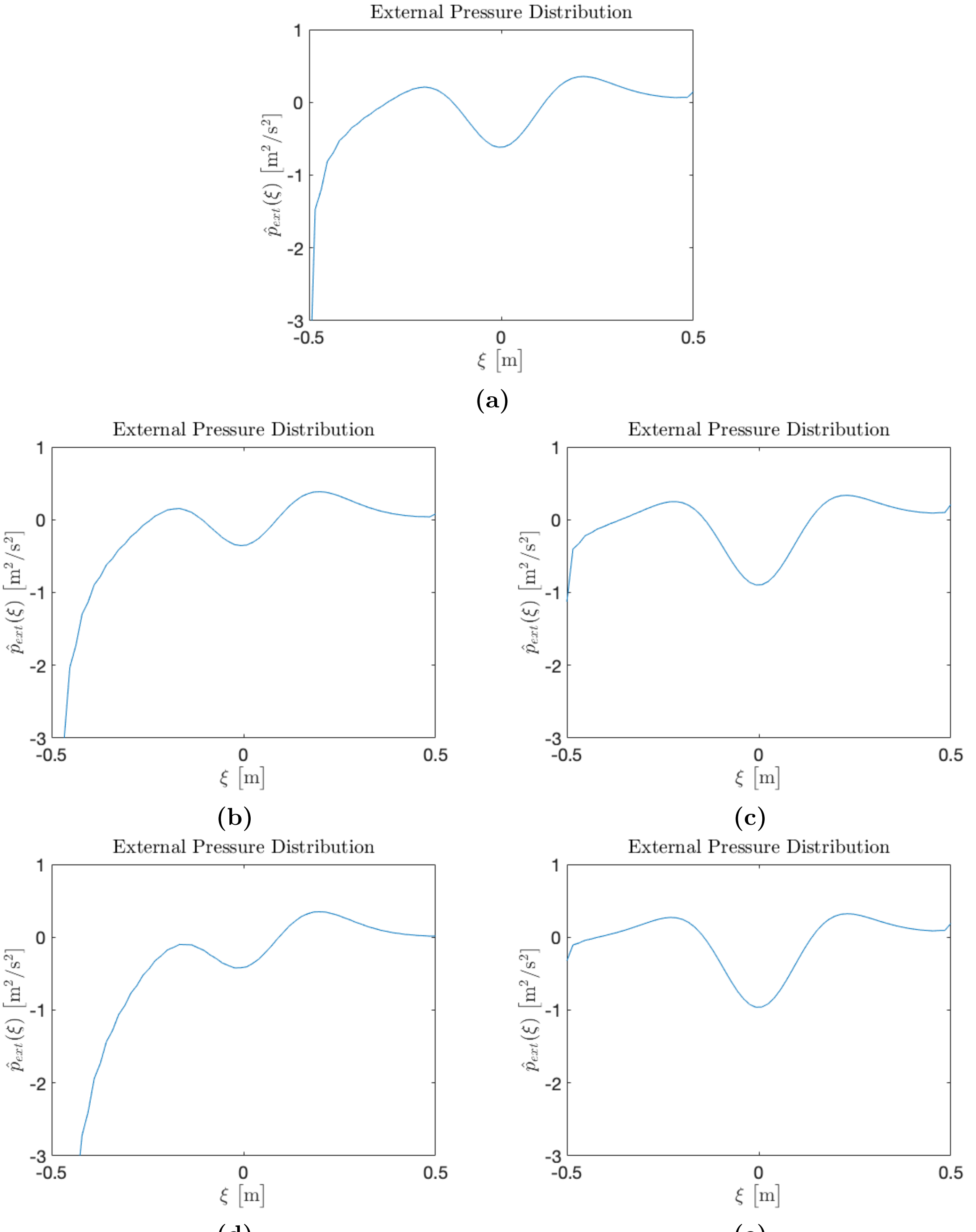


(a)

(b)

Figure 29: Plots showing the full resulting pressure patch when $Fr = 0.5$, $Fr_s = 0$, (a) $Fr_h = 0.1$ (b) $Fr_h = 0.9$

The effect of increasing the shear flow strength in 2D for both a shear-inhibited and shear-assisted system are shown in Figure 30. Moreover, the figure shows the effect of both a linear and an exponential shear current profile. As can be seen, the shear-assisted system causes the pressure spikes to be bigger, in addition to lowering the pressure values at the left side of the patch in general. The opposite is true for the shear-inhibited system, where the pressure spikes are smaller and the left side of the patch is in general raised, making the pressure distribution more symmetrical. It should also be noted that the exponential shear current profile seems to have a stronger effect on the pressure distribution compared to the linear profile.



(a)

(b)

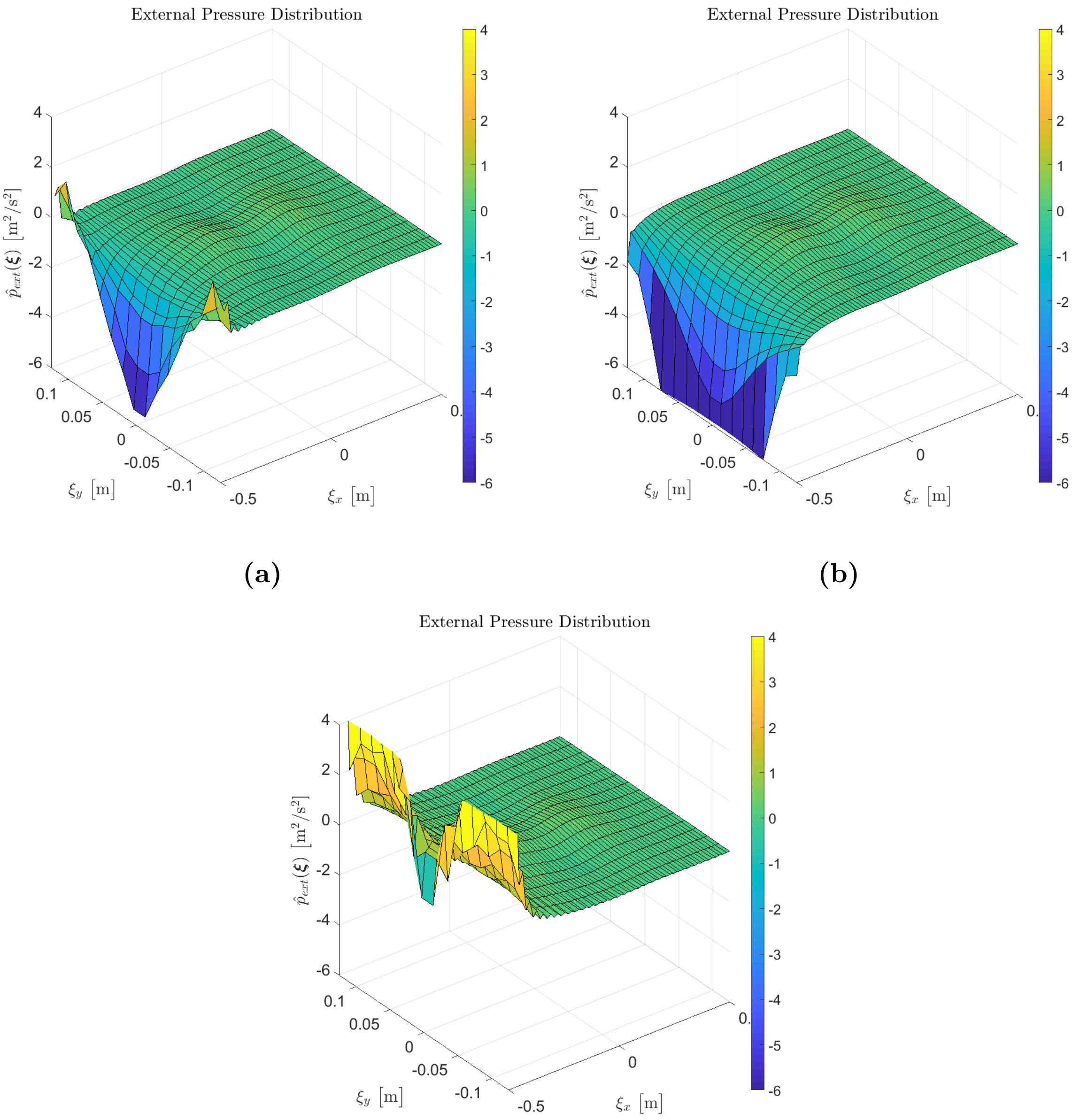
(c)

(d)

(e)

Figure 30: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ in a linear shear-assisted system, (c) $Fr_s = 0.5$ in a linear shear-inhibited system, (d) $Fr_s = 0.5$ in a exponential shear-assisted system, (e) $Fr_s = 0.5$ in a exponential shear-inhibited system.

The effects that the shear current profile had on the 2D pressure patches can be observed in 3D by looking at Figure 31 and 32. For a shear-assisted profile, the back side of the pressure patch is lowered, while for a shear-inhibited profile, the back side of the pressure patch is raised. The exception to this rule is the external pressure distribution for an exponential shear-assisted profile. Here, the back side of the pressure patch is raised. This anomaly seems to be connected to the problem of critical layers. A further discussion about critical layers can be found in section 5. Increasing the shear strength in the case of side-on shear profile causes an asymmetry in the pressure distribution. A high pressure is needed at the back right corner of the ship, upstream of the shear current, whereas a low pressure is needed at the back left corner of the ship, downstream of the shear current. This is shown in Figure 33.

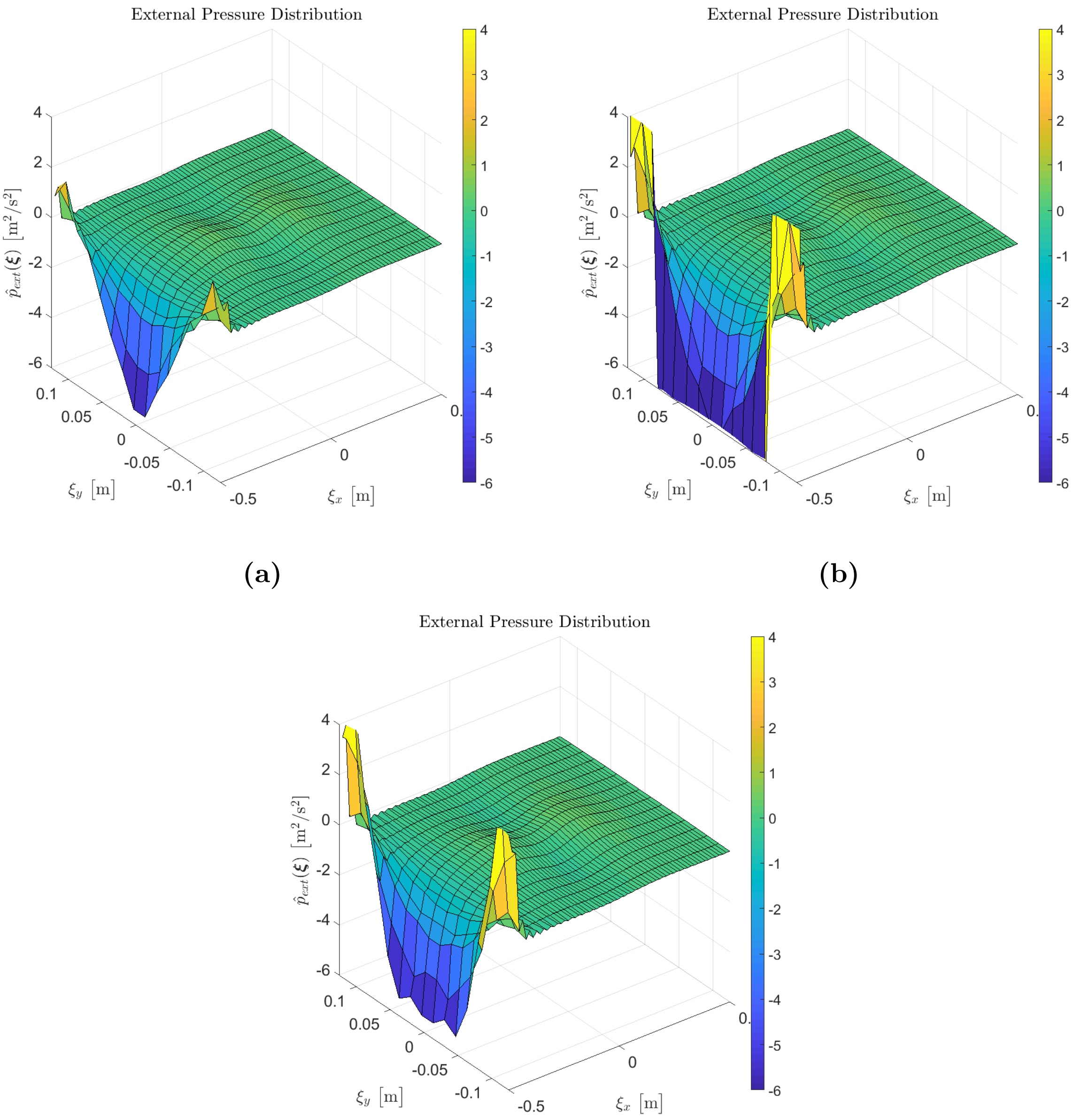


(a)

(b)

(c)

Figure 31: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr = 0.5$ with a linear shear-assisted profile, (b) $Fr = 0.5$ with a exponential shear-assisted profile.

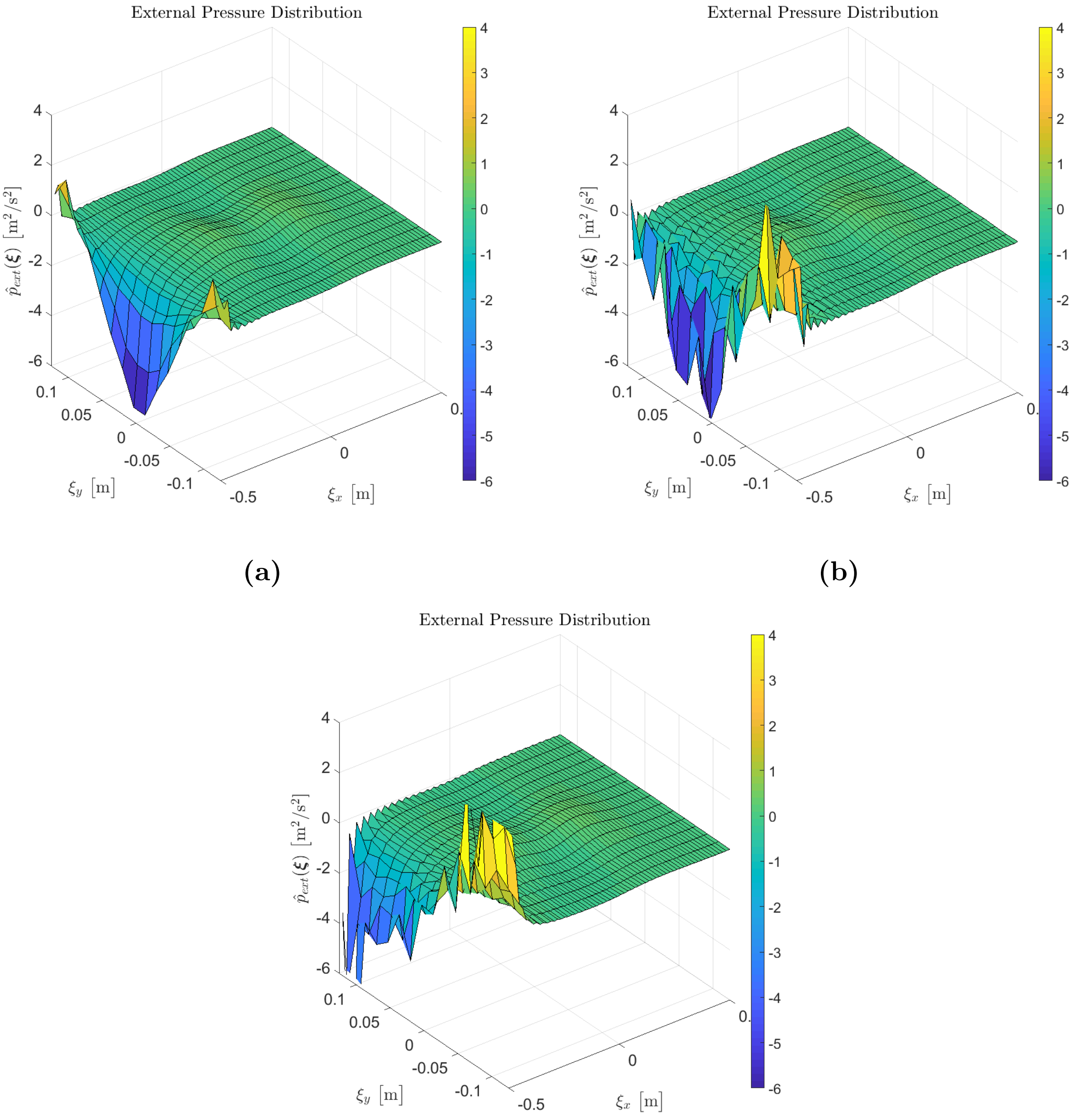


(a)

(b)

(c)

Figure 32: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr = 0.5$ with a linear shear-inhibited profile, (b) $Fr = 0.5$ with a exponential shear-inhibited profile.



(a)

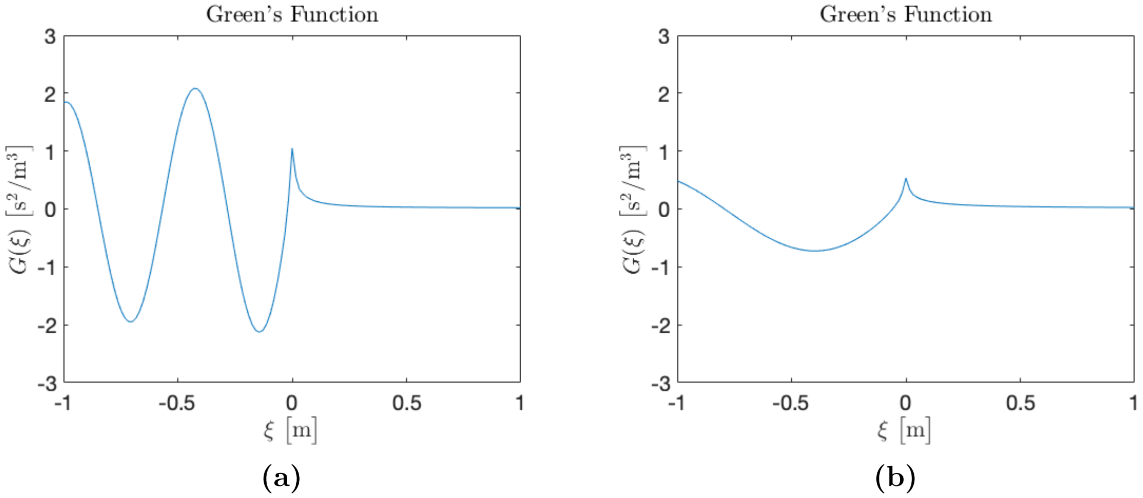
(b)

(c)

Figure 33: Plots showing the resulting pressure patch when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr = 0.5$ with a linear side-on shear profile, (b) $Fr = 0.5$ with a exponential side-on shear profile.

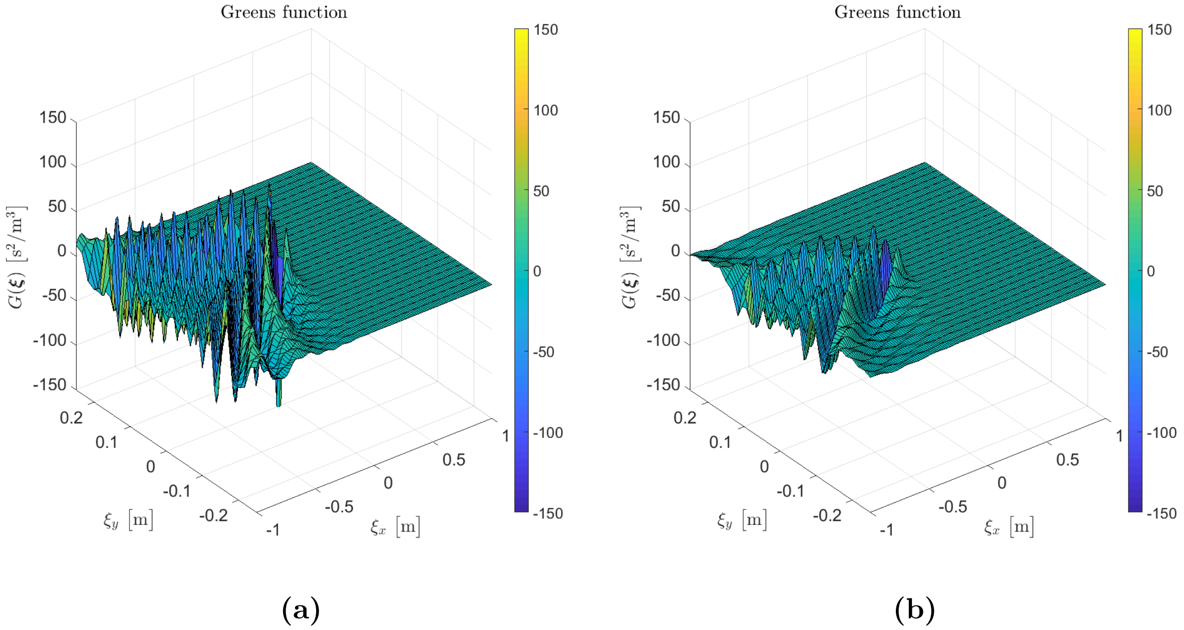
4.3 Green's function

Since the solution of the inverse problem is highly dependent on the Green's function, some insight into the nature of the resulting pressure patches can be gained from looking at the effect of different conditions on the Green's function. As shown in Figure 34, the effect of increasing the velocity is an increase in wavelength and decrease in amplitude for both the 2D and 3D case. This implies the response from applying a pressure patch on the water surface is dampened for higher velocities and higher wavelengths will occur.



(a) (b)

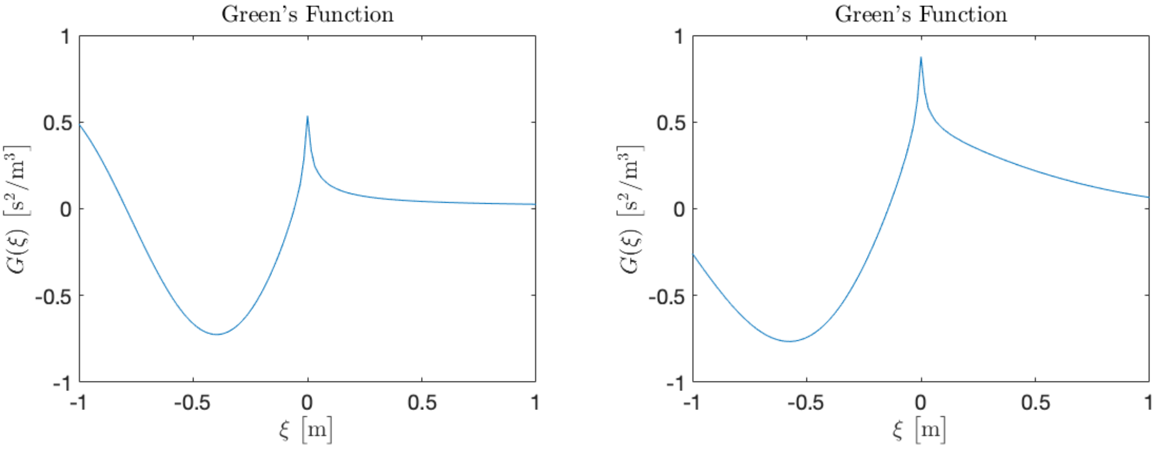
Figure 34: Plots showing the Green's function when $Fr_h = 0.1$, $Fr_s = 0$, (a) $Fr = 0.3$, (b) $Fr = 0.5$.



(a) (b)

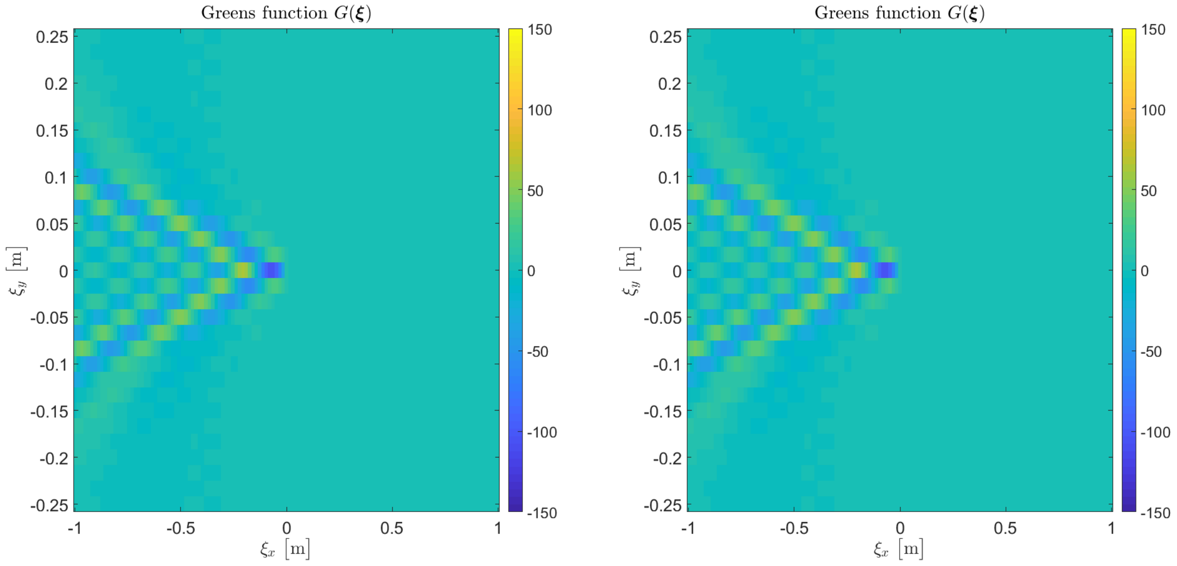
Figure 35: Plots showing the Green's function when $Fr_h = 0.1$, $Fr_s = 0$, (a) $Fr = 0.3$, (b) $Fr = 0.5$.

As for the external pressure, the effect of decreasing the water depth is only noticeable in 2D. When the ship is traveling through shallow waters in 2D, the Green's function have higher amplitudes upstream of the point pressure and the amplitudes are in general higher around the point pressure compared to a deep water case. Hence, the surface elevation of the point in question will be more affected by surrounding points in shallow water than in deep water. In addition, the wavelength of the pressure point response is longer in shallow water than in deep water. Figure 36 shows the effect of decreasing the water depth in 2D, while Figure 37 shows how similar the Green's function is at different water depths in 3D.



(a) **(b)**

Figure 36: Plots showing the Green's function when $Fr = 0.5$, $Fr_s = 0$, (a) $Fr_h = 0.1$ (b) $Fr_h = 0.9$.



(a) **(b)**

Figure 37: Plots showing the Green's function when $Fr = 0.5$, $Fr_s = 0$, (a) $Fr_h = 0.1$ (b) $Fr_h = 0.9$.

The effect of introducing a shear current profile to the problem in 2D is shown in Figure 38. Increasing the shear strength in a shear-assisted system will decrease the wavelength in the Green's function while in a shear-inhibited system the wavelength will increase.

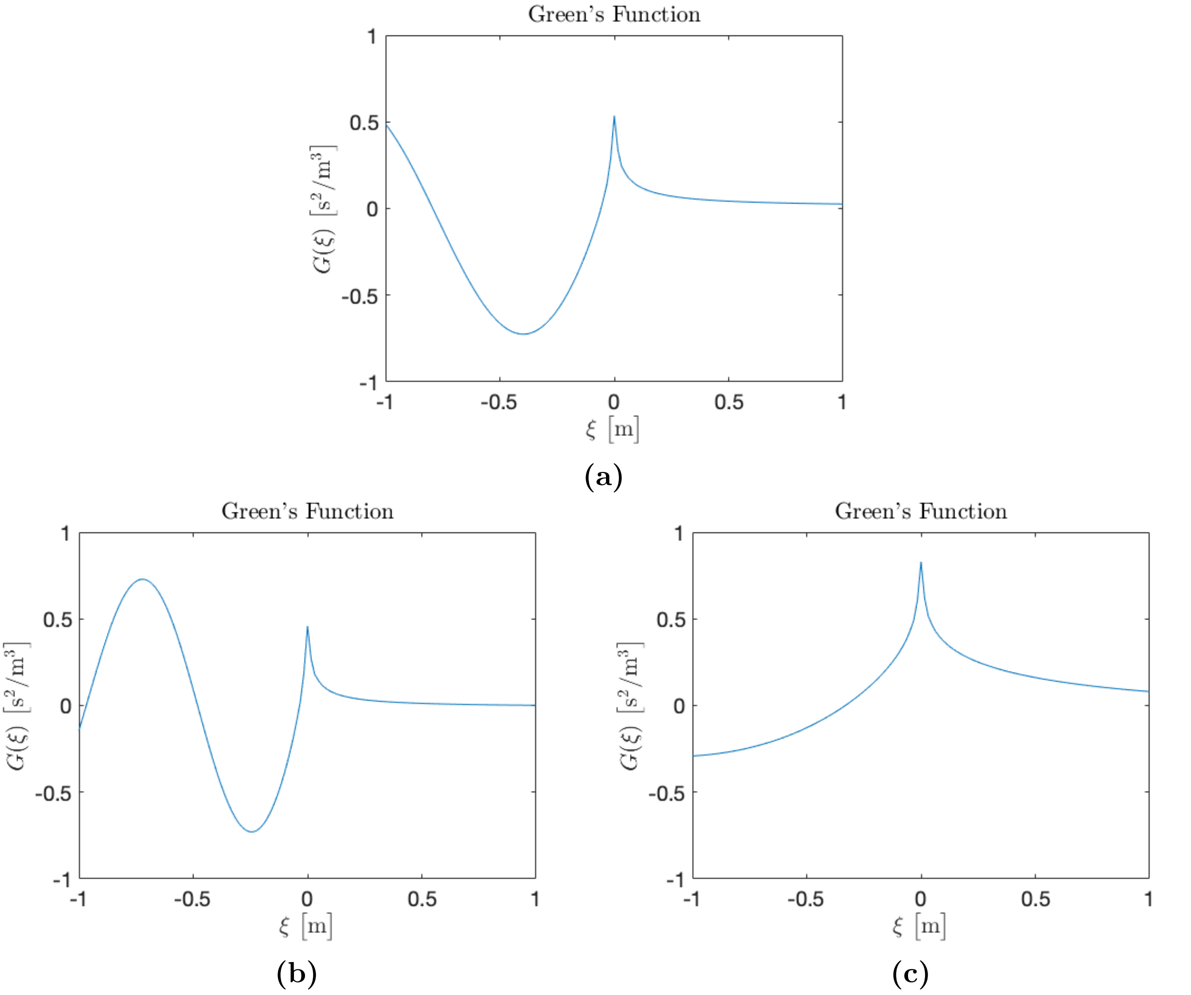
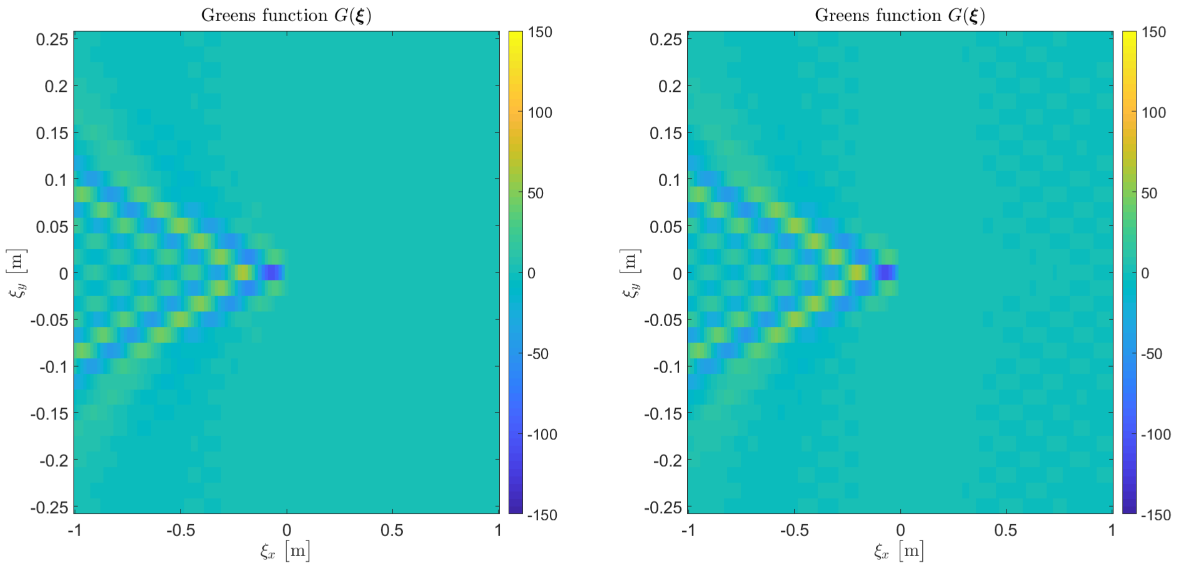


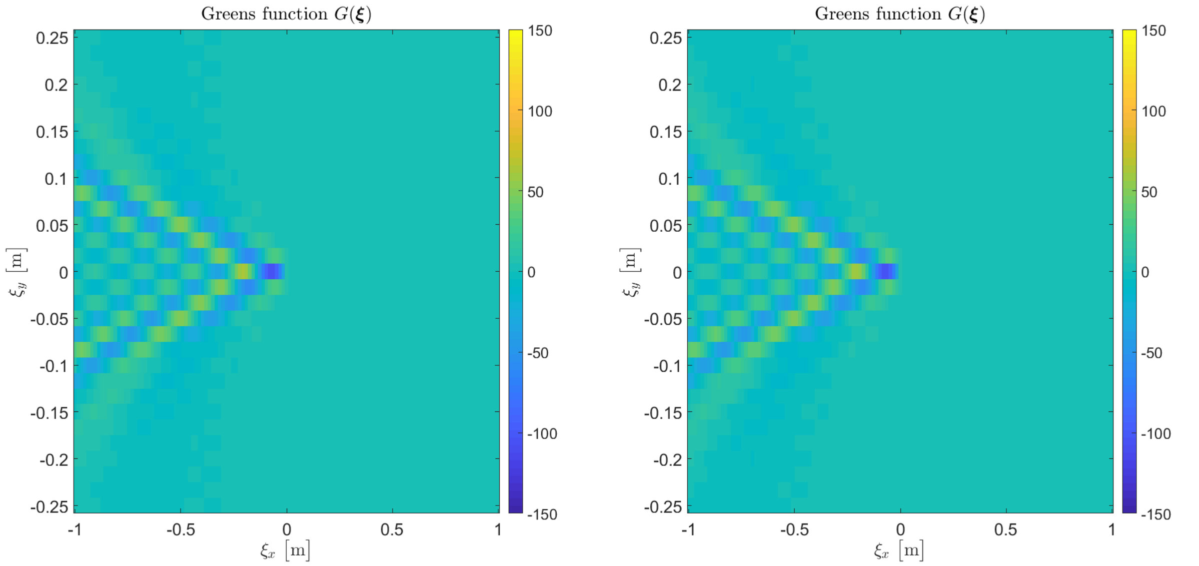
Figure 38: Plots showing the Green's function when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ in a exponential shear-assisted system, (c) $Fr_s = 0.5$ in a exponential shear-inhibited system.

The same effect can be seen in Figure 39 and 40 for the 3D case. For a side on shear profile, the Green's function will become asymmetrical and the V-shape will rotate slightly in the direction of the shear current as shown in Figure 41.



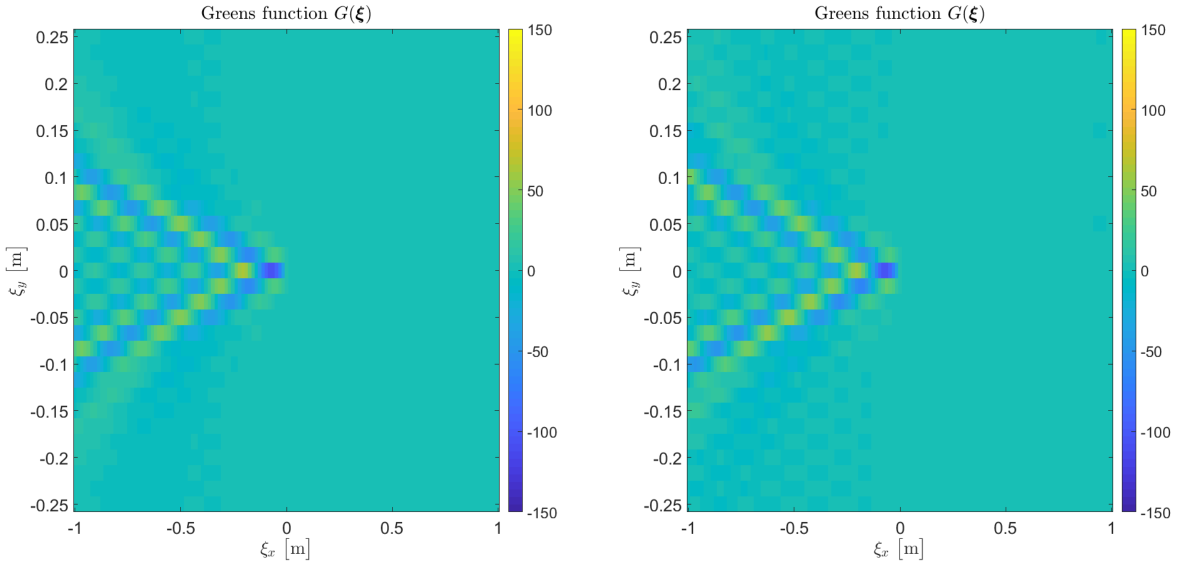
(a) **(b)**

Figure 39: Plots showing the Green's function when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential shear-assisted profile.



(a) **(b)**

Figure 40: Plots showing the Green's function when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential shear-inhibited profile.



(a) **(b)**

Figure 41: Plots showing the Green's function when $Fr = 0.5$, $Fr_h = 0.1$, (a) $Fr_s = 0$, (b) $Fr_s = 0.5$ with a exponential side-on shear profile.

4.4 Wave Pattern

It has been shown that using the calculated pressure patch, the resulting pressure, as well as the ship result, changes drastically. A change in the external pressure distribution does not, however, mean a big difference in wave pattern created by the moving ship. This section compares the wave pattern created by a calculated pressure patch to the constant Gaussian pressure patch used in section 4.1. All the different wave patterns for the different cases are shown in Appendix A.

4.4.1 2D

In 2D, there are two important differences in the wave pattern from the constant pressure source and the calculated pressure patch. Firstly, the surface elevation close to the ship, the near field, is significantly different. The calculated pressure patch forces the surface elevation in the region of the ship to shape of the effective wetted hull, and waves propagate accordingly from the source. While the near field is different for the constant and calculated pressure, the far field does not change significantly. This effect is best seen in the shallow water case illustrated in Figure 42.

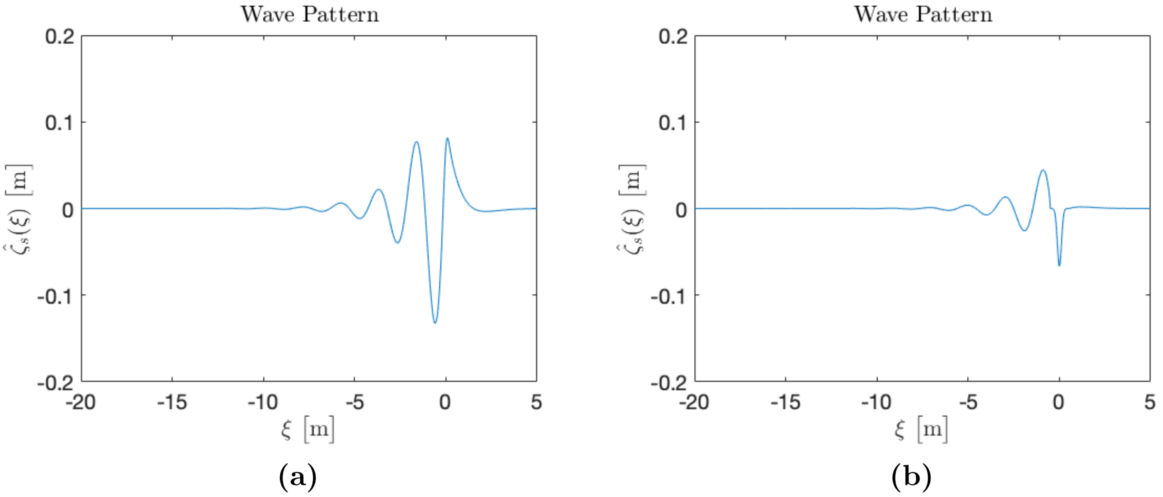


Figure 42: Plots showing the wave pattern for $Fr = 0.5$, $Fr_h = 0.9$, $Fr_s = 0$ for (a) a Gaussian pressure source, (b) the calculated pressure source.

Secondly, the amplitude of the waves is bigger for the calculated pressure patch compared to the constant Gaussian pressure patch. The reason for this is the spikes at the back of the pressure patch. The difference in amplitude is best shown with an exponential shear-assisted profile as illustrated in Figure 43. A list of all the differences in max amplitude is shown in Table 5.

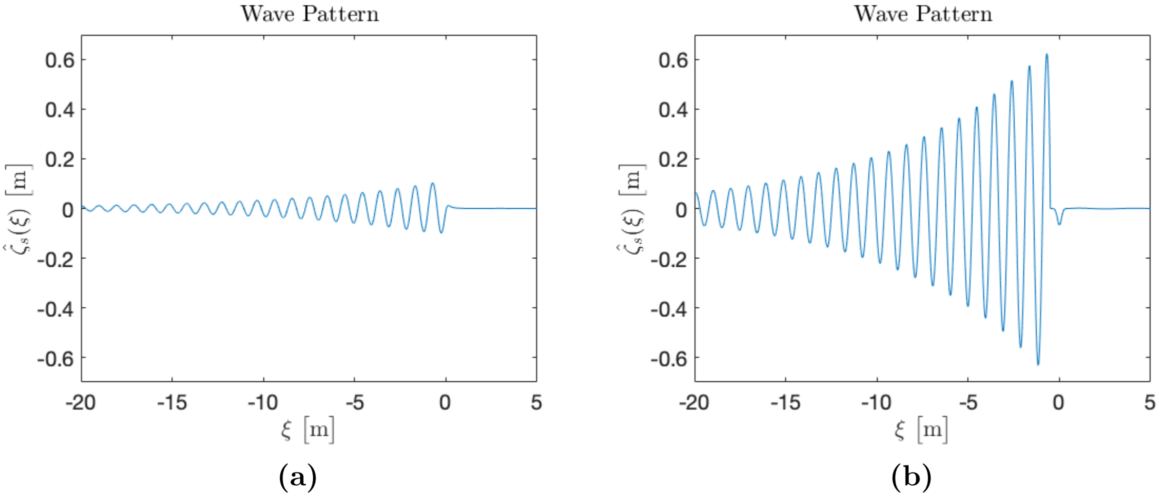


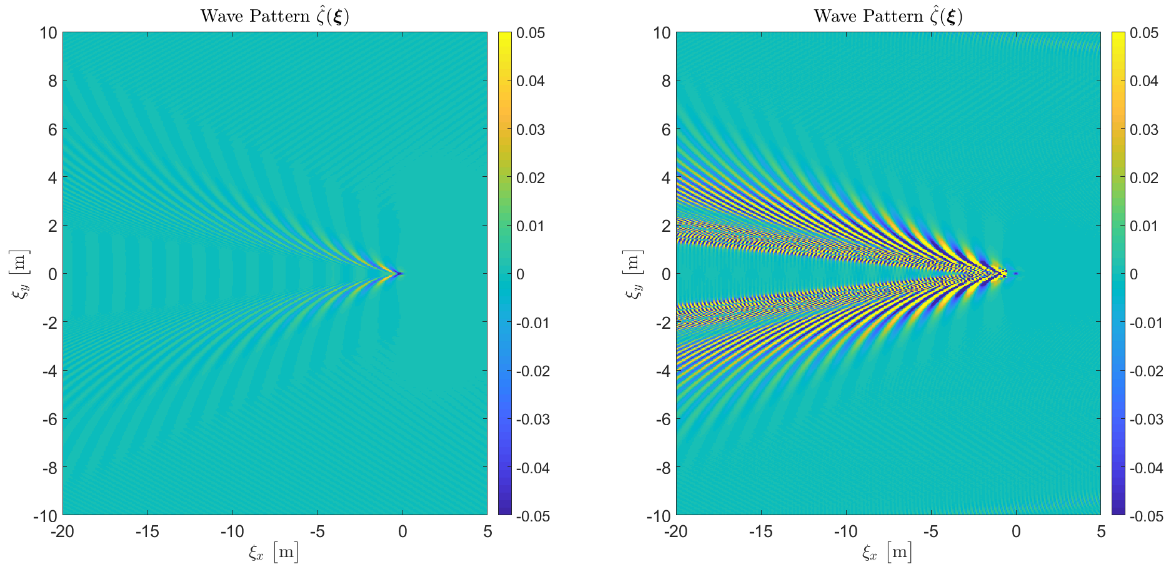
Figure 43: Plots showing the wave pattern of a shear-assisted system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for a exponential shear profile, where (a) is a Gaussian pressure source, (b) is the calculated pressure source.

Table 5: Table maximum amplitudes for the different cases. Amplitude C is the maximum amplitude of the waves created by a calculated pressure patch, while Amplitude G is the maximum amplitude of the waves created by a Gaussian pressure patch.

Fr	$\mathbf{Fr}_h$	$\mathbf{Fr}_s$	Shear Direction	Shear Profile	Amplitude C	Amplitude G
0.5	0.1	0	-	-	0.37708	0.17752
0.5	0.1	0	-	-	0.14488	0.10687
0.5	0.9	0	-	-	0.04449	0.081377
0.5	0.1	0.5	Assisted	Linear	0.33293	0.10906
0.5	0.1	0.5	Inhibited	Linear	0.051175	0.056612
0.5	0.1	0.5	Assisted	Exponential	0.12008	0.10325
0.5	0.1	0.5	Inhibited	Exponential	0.01528	0.075527

4.4.2 3D

The same differences in wave pattern between the constant and calculated pressure patch can be seen in 3D as in 2D. While the surface elevation in the near field is changed significantly, the wave pattern in the far field has the same form. In 3D, however, the differences in wave pattern are bigger than for 2D. The amplitude difference is amplified in 3D due to the bigger pressure spikes happening at the back of the pressure patch. Besides, the 3D calculated pressure patch creates some noise happening inside the transverse waves. This is due to the positive pressure spikes happening at the back corners of the pressure patch. All these differences can be seen in Figure 44. Furthermore, the increase in amplitude difference for 3D cases can be seen in Table 6.



(a) **(b)**

Figure 44: Plots showing the wave pattern for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$, where (a) is a Gaussian pressure source, and (b) is the calculated pressure source.

Table 6: Table maximum amplitudes for the different cases. Amplitude C is the maximum amplitude of the waves created by a calculated pressure patch, while Amplitude G is the maximum amplitude of the waves created by a Gaussian pressure patch.

Fr	**Fr**$_h$	**Fr**$_s$	**Shear Direction**	**Shear Profile**	**Amplitude C**	**Amplitude G**
0.3	0.1	0	-	-	1.2482	0.038428
0.5	0.1	0	-	-	0.62602	0.042634
0.5	0.9	0	-	-	0.59291	0.038362
0.5	0.1	0.5	Assisted	Linear	0.55171	0.047824
0.5	0.1	0.5	Inhibited	Linear	0.86136	0.034200
0.5	0.1	0.5	Side-on	Linear	2.1118	0.043671
0.5	0.1	0.5	Assisted	Exponential	2.0295	0.047916
0.5	0.1	0.5	Inhibited	Exponential	1.0006	0.033161
0.5	0.1	0.5	Side-on	Exponential	3.1245	0.042965

4.5 Verification

4.5.1 Forward Problem

The resulting error in ship results for the different cases can be seen in Table 7 and 8. The tables show that there is a clear difference in accuracy for the calculated pressure patch and the constant Gaussian pressure patch. While the error of the Gaussian pressure is in the order of magnitude of 10^{-1}, the calculated pressure patch's errors are typically in the order of magnitude of 10^{-6} for 2D and 10^{-4} for 3D.

Table 7: Table showing the mean relative errors for different cases in 2D. Fr is the Froude number $\mathrm{Fr} = \frac{V}{\sqrt{gL}}$ where L is the longitudinal length of the ship. $\mathrm{Fr_s}$ is the shear Froude Number defined as $\mathrm{Fr_s} = \frac{VU_0'}{g}$. $\mathrm{Fr_h}$ is the height Froude number $\mathrm{Fr_h} = \frac{V}{\sqrt{gh}}$. The cases are simulated using $\epsilon = 2$, $N_z = 200$, a 64 point ship, and a 32768 point domain. Error C is the mean relative error in the region of the ship for a calculated pressure patch, while Error G is the mean relative error in the region of the ship for a Gaussian pressure patch.

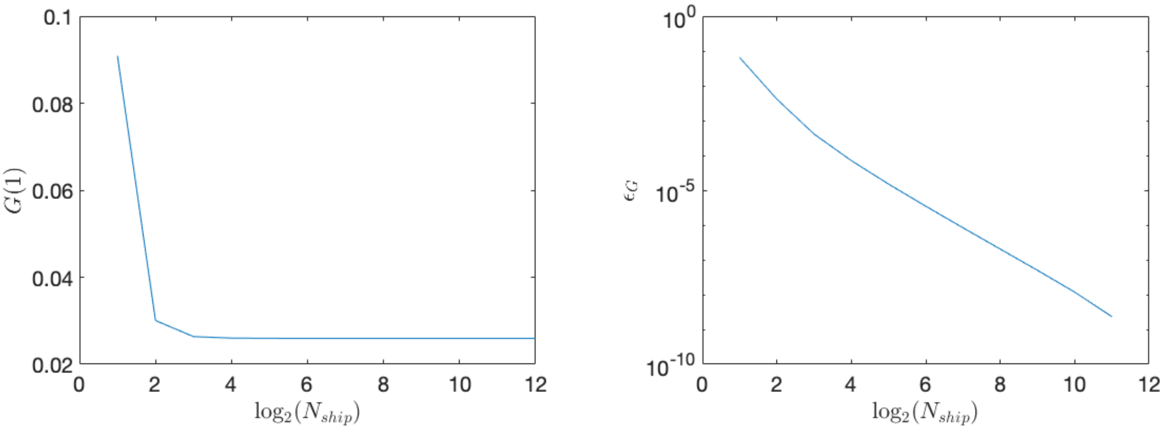
Fr	**Fr**$_\mathbf{h}$	**Fr**$_\mathbf{s}$	**Shear Direction**	**Shear Profile**	**Error C**	**Error G**
0.3	0.1	0	-	-	1.59×10^{-5}	0.97670
0.5	0.1	0	-	-	3.6773×10^{-6}	0.85404
0.5	0.9	0	-	-	9.1142×10^{-7}	1.2204
0.5	0.1	0.5	Assisted	Linear	1.01×10^{-5}	0.66793
0.5	0.1	0.5	inhibited	Linear	1.07×10^{-6}	0.89279
0.5	0.1	0.5	Assisted	Exponential	2.0326×10^{-6}	0.57895
0.5	0.1	0.5	inhibited	Exponential	4.0892×10^{-7}	0.89279

Table 8: Table showing the mean relative errors for different cases in 3D. Fr is the Froude number $\mathrm{Fr} = \frac{|\boldsymbol{V}|}{\sqrt{gL}}$ where L is the longitudinal length of the ship. $\mathrm{Fr_s}$ is the shear Froude Number defined as $\mathrm{Fr_s} = \frac{|\boldsymbol{V}||\boldsymbol{U}_0'|}{g}$. $\mathrm{Fr_h}$ is the height Froude number $\mathrm{Fr_h} = \frac{|\boldsymbol{V}|}{\sqrt{gh}}$. The cases are simulated using $\epsilon = 2$, $N_z = 200$, a 16 × 64 point ship, and a 2048 × 8192 point domain. Error C is the mean relative error in the region of the ship for a calculated pressure patch, while Error G is the mean relative error in the region of the ship for a Gaussian pressure patch.

Fr	$\mathbf{Fr_h}$	$\mathbf{Fr_s}$	**Shear Direction**	**Shear Profile**	**Error C**	**Error G**
0.3	0.1	0	-	-	1.0730×10^{-4}	0.19001
0.5	0.1	0	-	-	1.5353×10^{-4}	0.21462
0.5	0.9	0	-	-	1.5725×10^{-4}	0.22644
0.5	0.1	0.5	Assisted	Linear	3.0959×10^{-5}	0.20312
0.5	0.1	0.5	Inhibited	Linear	1.7962×10^{-4}	0.21558
0.5	0.1	0.5	Side-On	Linear	1.8779×10^{-4}	0.21386
0.5	0.1	0.5	Assisted	Exponential	2.1074×10^{-4}	0.19846
0.5	0.1	0.5	Inhibited	Exponential	9.7286×10^{-5}	0.21550
0.5	0.1	0.5	Side-On	Exponential	3.006×10^{-3}	0.21402

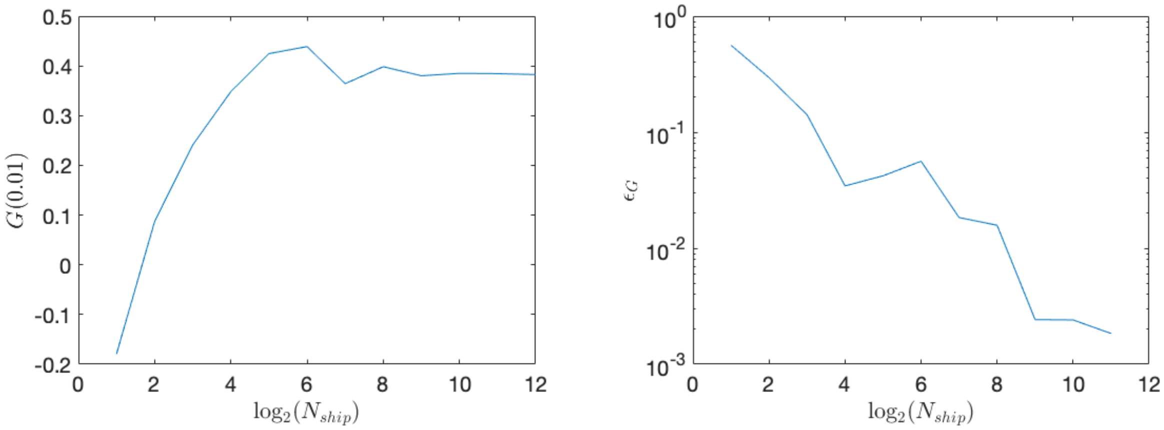
4.5.2 2D Green's function

Since the results are highly dependent on the Green's function, it is desirable to have a converged solution of the function. As mentioned in section ref(), the Green's function is divergent at the point $G(\xi = 0)$. Unfortunately, it is not possible to get accurate solutions for the points close to $\xi = 0$ as well. The reason for this is that the same k-values as in the FFT are used for the integration, making the k_{max}-value dependent on $\Delta\xi$. Trying to get higher k_{max}-values by making $\Delta\xi$ smaller will create points closer to $\xi = 0$, which in turn requires higher k_{max}-values. In addition, the calculations quickly become computationally expensive by decreasing $\Delta\xi$. Thus, this thesis is only concerned with obtaining reasonably accurate solutions of the Green's function for parts of the domain. As seen in Figure 45, the solution of the Green's function is quickly converged to a solution for points far away from $\xi = 0$. However, for points closer to $\xi = 0$, the Green's function is slower to converge. Figure 46 illustrates this.



(a) **(b)**

Figure 45: Plots showing the convergence of the Green's function at the fastest converging point, $G(\xi = 1)$. The test is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$.

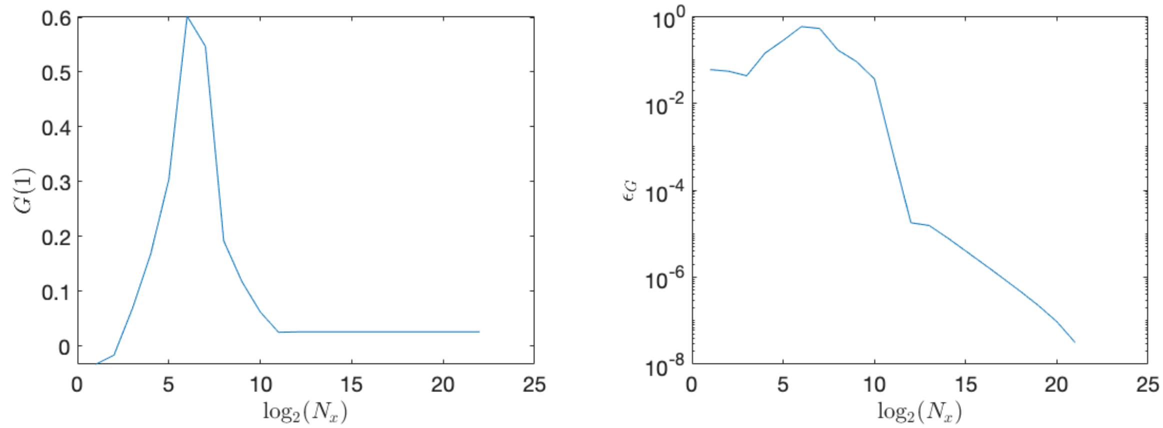


(a) **(b)**

Figure 46: Plots showing the convergence of the Green's function at a slow converging point, $G(\xi = 0.01)$. The test is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$.

As can be seen from Figure 46, the Green's function is not quite converged for $\xi = 0.01$ when using $N_{ship} = 2^6$. Nevertheless, having $N_{ship} = 2^6$ is a reasonable choice as this on the verge of making the calculations too computationally expensive for a normal computer. For points $\xi > 0.01$, the solution should be reasonably converged. It should be noted that the convergence of the Green's function is also affected by the shear current profile and the water depth. However, the conclusion remains the same. The Green's function will be well converged for points far away from $\xi = 0$, while being quite converged for points close to $\xi = 0$.

Now, given that $N_{ship} = 2^6$, the problem is to find the required Δk in order to solve the integrals correctly. The convergence of the integral regarding the number of points in the domain N_x is shown in Figure ref() for the different cases in the highest oscillating point $G(\xi = 1)$.



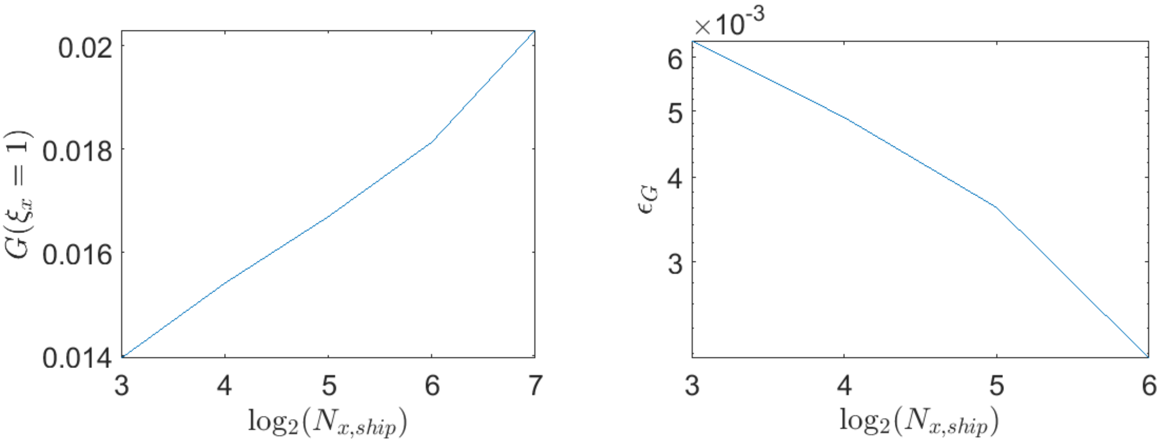
(a) **(b)**

Figure 47: Plots showing the convergence of the Green's function at the most oscillating point, $G(1)$. The error is defined as the difference between the value of the Green's function integral for the current domain N_x to the value of the integral when the domain is $N_x = 2^{22}$. The test is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$

By looking at the plots in Figure 47, it is clear that choosing a domain size of $N_x = 2^{15}$ should be sufficient for the 2D case when $\epsilon = 2$ and $N_{ship} = 2^6$. Also here, the convergence of the Green's function is affected by the shear current profile and the water depth. However, having $N_x = 2^{15}$ makes the integrals converged for all the different cases.

4.5.3 3D Green's function

The same issues regarding the convergence of the Green's function are encountered in 3D as in 2D, only now the computational effort is an even bigger issue. As seen in Figure 48, $G(\xi_x = 1)$ is in the 3D case not converged for $N_{x,ship} = 2^6$.



(a) **(b)**

Figure 48: Plots showing the convergence of the Green's function at the fastest converging point, $G(\xi_x = 1)$. The test is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$.

Choosing $N_{x,ship} = 2^6$ is the highest number of points in the region of the ship possible without the use of supercomputers for the current numerical method. Figure 49 shows that by choosing $N_x = 2^{13}$ should be sufficient for making the integrals converge with respect to the domain size.

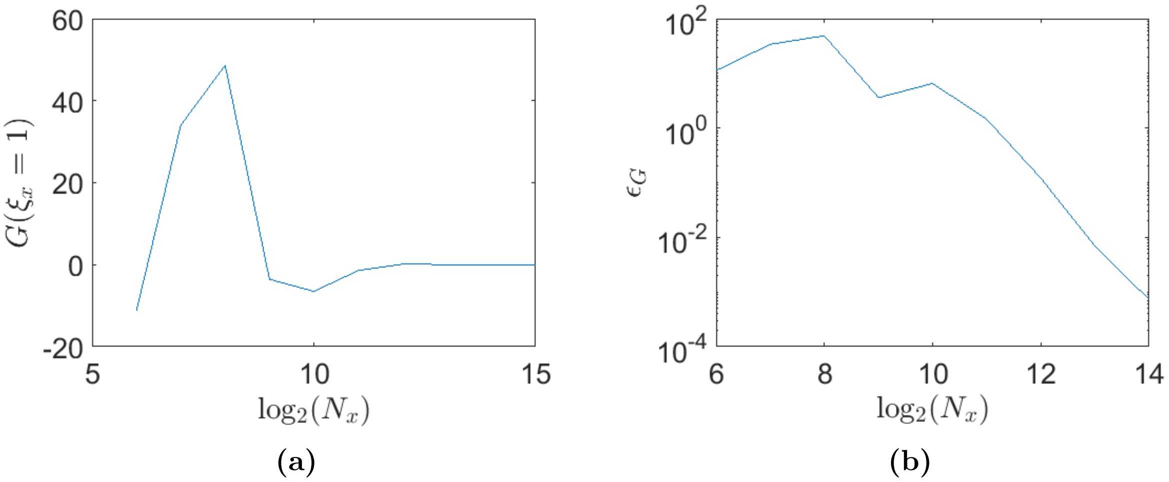
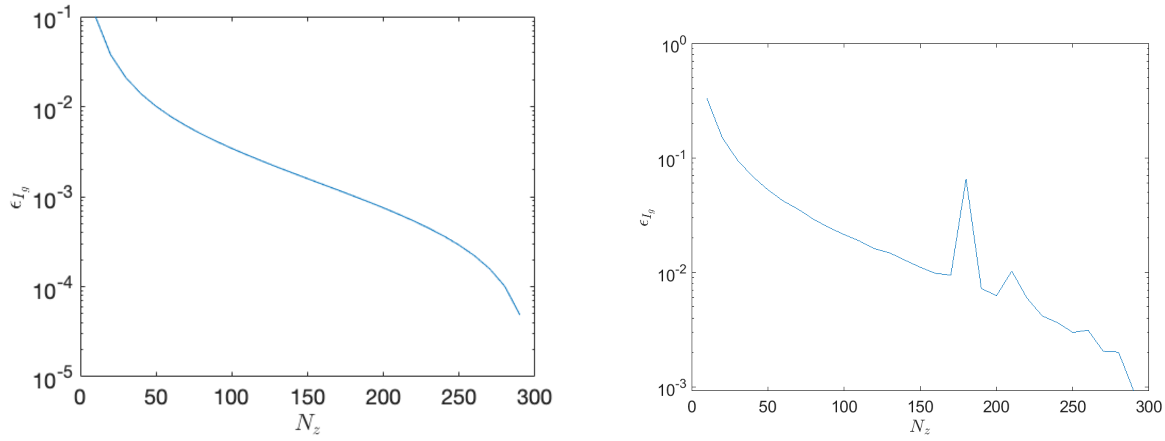


Figure 49: Plots showing the convergence of the Green's function at the most oscillating point, $G(1)$. The error is defined as the difference between the value of the Green's function integral for the current domain N_x to the value of the integral when the domain is $N_x = 2^{15}$. The test is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$

4.5.4 Convergence of I_g

I_g is used in both the forward problem and the inverse problem and plays an important role in the results. As the parameter N_z is not used for other calculations other than for I_g, the verification of I_g is important when determining N_z. Figure 50 shows the convergence of I_g as a function of N_z.



(a) **(b)**

Figure 50: Plots showing the convergence of I_g with respect to N_z. The simulation is done for $\epsilon = 2$, $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ with an exponential shear-assisted profile. Moreover, (a) is the 2D-case with $N_{ship} = 2^6$ and $N_x = 2^{15}$ while (b) is 3D-case with $N_{x,ship} = 2^6$ and $N_x = 2^{13}$.

By looking at the plots, it is clear that I_g is reasonably well converged for values of $N_z = 200$ with errors in the order of magnitude of 10^{-3}.

4.5.5 2D Wavelength

A comparison between the theoretical and measured wavelength in 2D is presented in Table 9.

Table 9: Table comparing the theoretical wavelength to the measured wavelength.

Fr	**Fr**$_h$	**Fr**$_s$	**Shear Direction**	**Shear Profile**	$\boldsymbol{\lambda}_m$	$\boldsymbol{\lambda}_t$	$\boldsymbol{\epsilon}_\lambda$
0.3	0.1	0	-	-	0.5652	0.5655	5.4×10^{-4}
0.5	0.1	0	-	-	1.5651	1.5708	3.6×10^{-3}
0.5	0.9	0	-	-	2.2016	2.2580	2.5×10^{-2}
0.5	0.1	0.5	Assisted	Linear	1.0445	1.0472	2.5×10^{-3}
0.5	0.1	0.5	Inhibited	Linear	3.0315	3.1416	3.5×10^{-2}
0.5	0.1	0.5	Assisted	Exponential	0.9619	0.9631	1.1×10^{-3}
0.5	0.1	0.5	Inhibited	Exponential	3.5485	8.8063	6.0×10^{-1}

As seen in the table, the relative error between the measured and theoretical wavelength are in most cases lower than 3%. The exception is for the exponential shear-inhibited case where a relatively high error is present. Moreover, the calculated wavelength is, in general, further away from the theoretical wavelength when the waves are heavily damped. This is the case for both the shear-inhibited cases as well as the shallow water case. One reason for this could be that as the waves are heavily damped, the effect of having a too high ϵ is bigger since ϵ can be seen as the viscous damping of waves.

4.5.6 Inverse Problem

The high pressure spikes happening at the back of the pressure patch might seem a little strange. The Green's function matrix have relatively high condition numbers, and as the inverse problem is a Fredholm type one problem, small inaccuracies in the input are known to create big differences in the results. To test if these pressure spikes are a result of inaccuracies in the inverse problem solver, the test described in section 3.3.5. Figure 51 shows this test for both the 2D and 3D case.

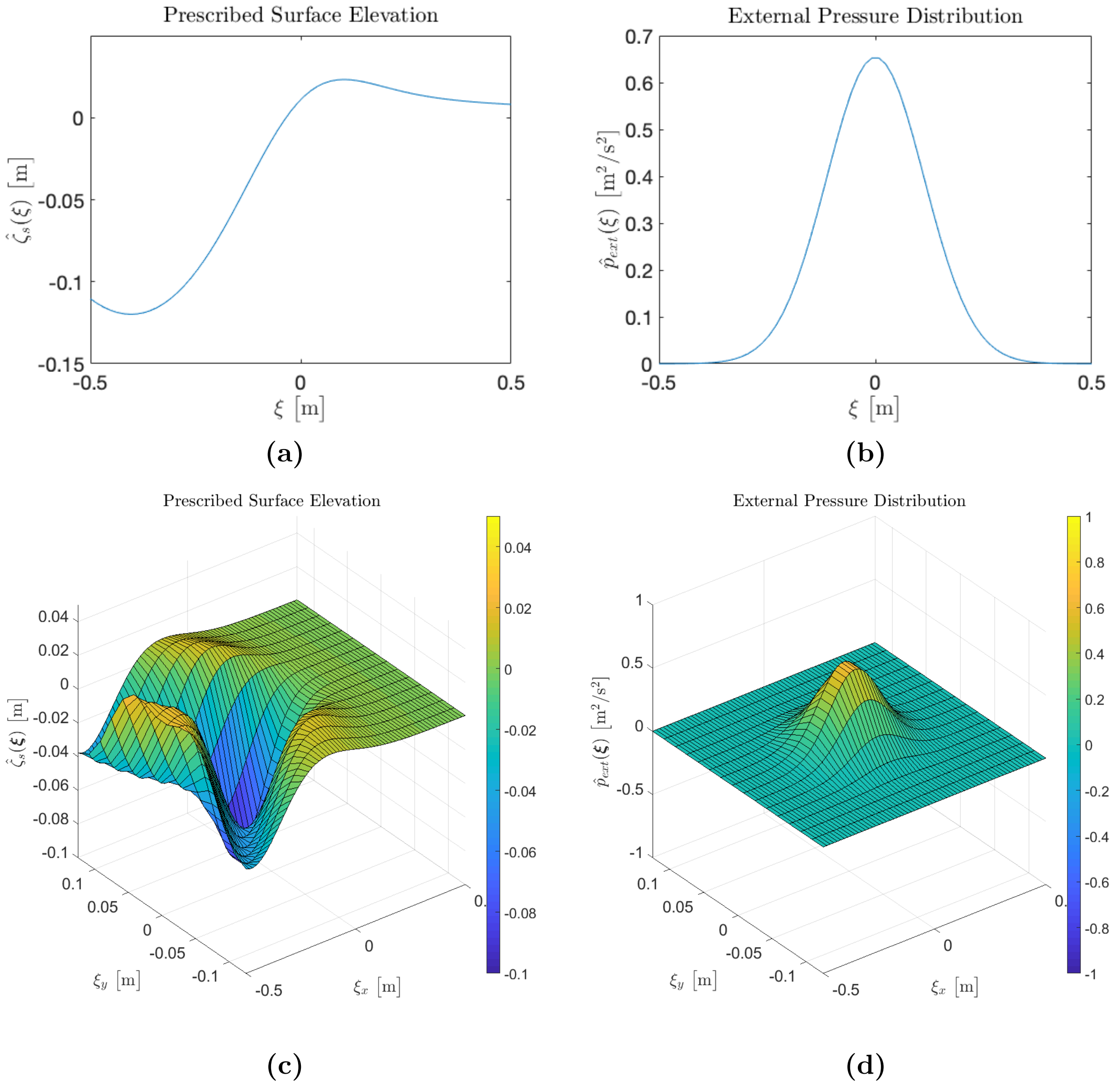


Figure 51: Plots showing the resulting pressure patch from the given prescribed surface elevation. The given surface elevation is known to be created by a Gaussian pressure patch. The simulation is done for $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$. (a,b) 2D case, (c,d) 3D case

As seen in the plots, the inverse solver is able to get the Gaussian pressure distribution back. In fact, the mean relative error of the pressure is $\epsilon_p = 3.2974 \times 10^{-12}$ in 2D and $\epsilon_p = 1.0776 \times 10^{-10}$ in 3D.

5 Discussion

Solving the inverse problem shows that there are some differences between the calculated pressure patch and the constant Gaussian pressure patch both in the near field and the far field. The calculated pressure patch is able to get the wanted prescribed surface elevation in the area of the ship to a much higher accuracy than the Gaussian. However, the waves created by the calculated pressure are of a higher amplitude than for the Gaussian pressure. In fact, the Gaussian pressure creates amplitudes closer to what is expected in real life, as the calculated pressure creates waves up to 50 times the height of the hull.

The high amplitudes appearing when using the calculated pressure patch is due to the high pressure spikes happening at the back of the patch. There could be several reasons for this. As explained above, a possible reason for such a result could be the high condition numbers of the Green's function matrix used to solve the linear equation system. However, the inverse solver is able to accurately reproduce the benchmark pressure without any big pressure spikes. Hence the high amplitudes are not likely to be caused by inaccuracies in the inverse solver.

A more troublesome issue is the convergence of I_g and the Green's function. Due to the computational effort needed in both time and memory, the calculations are not done with highly accurate solutions of I_g and the Green's function. This will create oscillations in the solution. Nonetheless, these oscillations will be all over the pressure patch and does not cover up the fact that there exist big pressure spikes at the back of the patch. Hence, the convergence of I_g and the Green's function alone cannot be the reason for the pressure spikes.

Another simplification made in order to make the problem computationally reasonable is the relatively high value of the radiation condition parameter ϵ. As mentioned in section 2.1.3, L.D Landau and E.M. Lifshitz approximate the relation for this parameter as $\epsilon = 2\nu k^2$, which means that the value of ϵ should be in the order of magnitude of 10^{-4} for the most of the wave spectrum that is contributing to the stationary waves. Furthermore, since the viscosity is often neglected in gravity waves, the desired way to treat ϵ is to let it go to zero after doing the integral. This is not possible, however, when doing the calculations numerically as done in this thesis. Nonetheless, the value of $\epsilon = 2$ is too high for normal ship waves. This is the reason why some of the wave patterns look odd for the 2D case, as the viscous damping of the waves is unnaturally high. Also, the Green's function integral is changed significantly compared to lower values of ϵ. However, since radiation condition parameter can be looked at as the viscous damping of waves, it is assumed that choice of ϵ will not have a significant effect on the result close to and in the area of the ship.

A possible explanation of the high pressures in the pressure patch is the unnatural shape forced on the surface elevation. First of all, a Gaussian ship hull is not a realistic hull for a ship. Second of all, a moving ship would not create the same effective wetted hull on the water surface as a ship standing still or a ship moving at another velocity. The calculated pressure patch would, in accordance with Newton's third law, also act on the ship hull. As seen in the results, there is, in most cases, a big negative pressure spike at the back of the pressure patch. This would create a downward force on the ship, making the ship tilt backward as can be seen in real life for boats traveling at a certain velocity. When trying to force the water surface to be exactly the same as a Gaussian ship standing still, there is a substantial negative pressure needed in order to "suck up" the water so quickly after being pushed downwards. Likewise, the big positive pressure spikes happening at the corners of the pressure patch can be explained by needing to stop the upward momentum of the water in order to create the corners of the ship hull.

6 Concluding Remarks

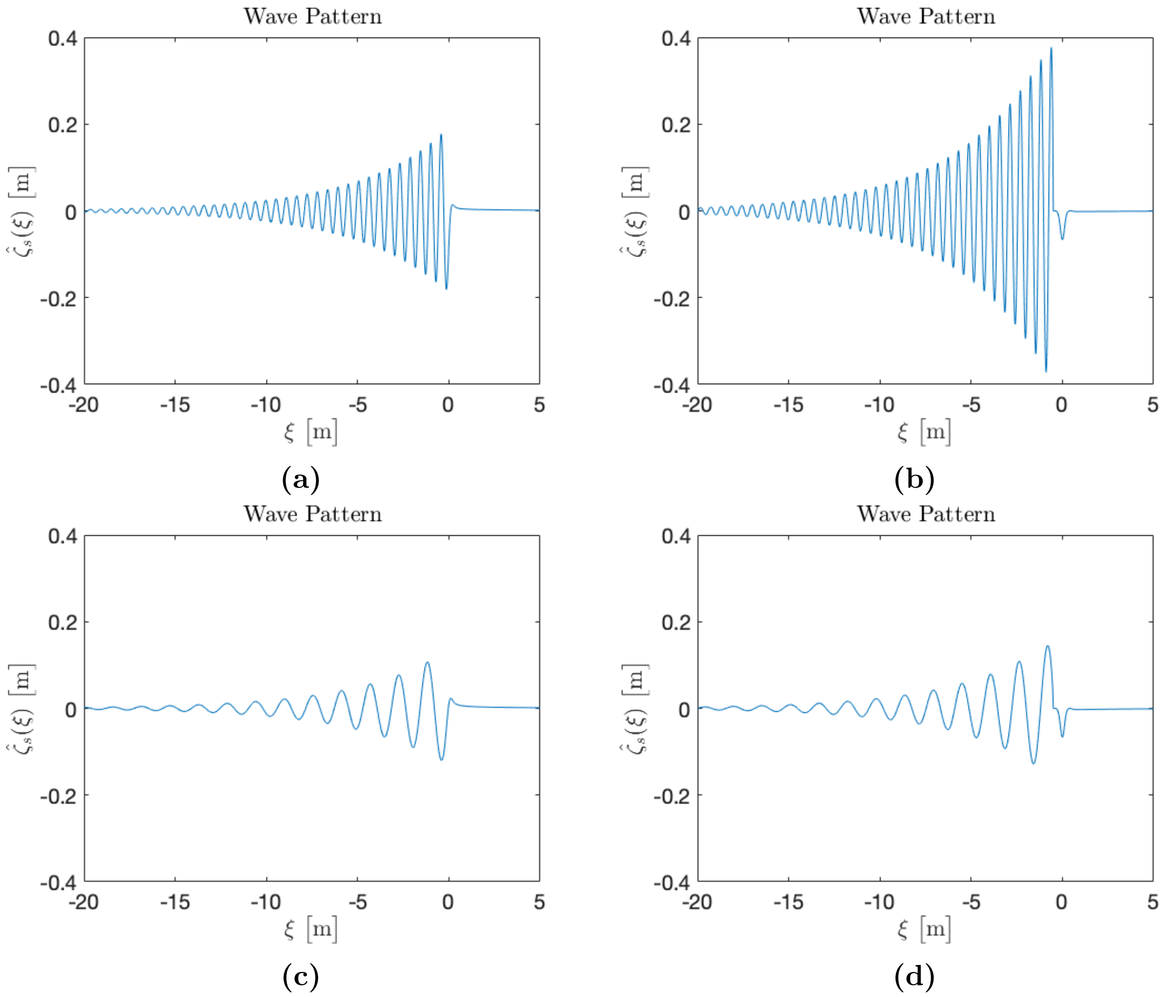
This thesis investigates a method for finding a pressure patch that is able to mirror a given effective wetted hull of a ship onto the water surface in a certain region. Compared to a constant Gaussian pressure patch, often used in investigations of ship waves with wave-current interactions, the calculated pressure patch gives a more accurate representation of the wanted surface elevation in the region of the ship. Similar to the surface elevation in the region of the ship, the pressure patch needed to get the prescribed surface elevation is dependent on the velocity, water depth, and shear current profile. Increasing the velocity of the moving ship causes the calculated pressure patch to have bigger pressure variations throughout the patch. When decreasing the water depth, the pressure patch needs smaller pressure spikes at the back of the patch in 2D, while in 3D the effect of decreasing water depth is negligible. The wave pattern created in both 2D and 3D is, however, affected by the water depth. Thus, agreeing with the results of [3]. Introducing a shear-assisted profile to the problem causes the pressure patch to have bigger negative pressure spikes at the back of the ship region while having a shear-inhibited profile makes the pressure patch have less negative pressure spikes. Furthermore, having a side-on shear profile causes an asymmetric pressure patch with more positive values upstream of the shear current and more negative values downstream.

While the results in the region of the ship are significantly different when using a calculated pressure patch compared to the constant Gaussian, the surface elevation difference is smaller in the far field. The same wave pattern can be seen from the calculated pressure patch as for the constant Gaussian. Furthermore, the effect of velocity, water depth, and shear current profile regarding the wave pattern are the same for both Gaussian and the calculated pressure patch. However, the effect of the pressure spikes is noticeable in the wave pattern for the calculated pressure patch. The effect of the pressure spikes are three-fold; First of all, the surface elevation in the near field is changed significantly. Second of all, the area at which the largest waves exist increase. Third of all, the amplitudes of the waves increase significantly compared to a constant Gaussian pressure patch. These differences in the wave pattern and the pressure spikes causing them indicates an issue in the way the calculated pressure patch model is currently used. While the prescribed surface elevation would be the correct effective wetted hull for a ship standing still, the given surface elevation is not natural for a moving ship. In accordance to Newton's third law, the big pressure spikes will create a force on the moving ship that affects the position of the ship on the water surface, both in height and in tilt angle. Therefore, there is a need for a more sophisticated method of predicting the effective wetted ship hull for different conditions if this model is to be used to simulate a moving ship, both regarding the wave pattern and the pressure forces acting on the ship hull.

7 Appendix A

7.1 Wave pattern

7.1.1 2D



(a)

(b)

(c)

(d)

Figure 52: Plots showing the wave pattern for (a,b) $Fr = 0.3$, $Fr_h = 0.1$, $Fr_s = 0$, (c,d) $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$, where (a,c) is a Gaussian pressure source, and (b,d) is the calculated pressure source.

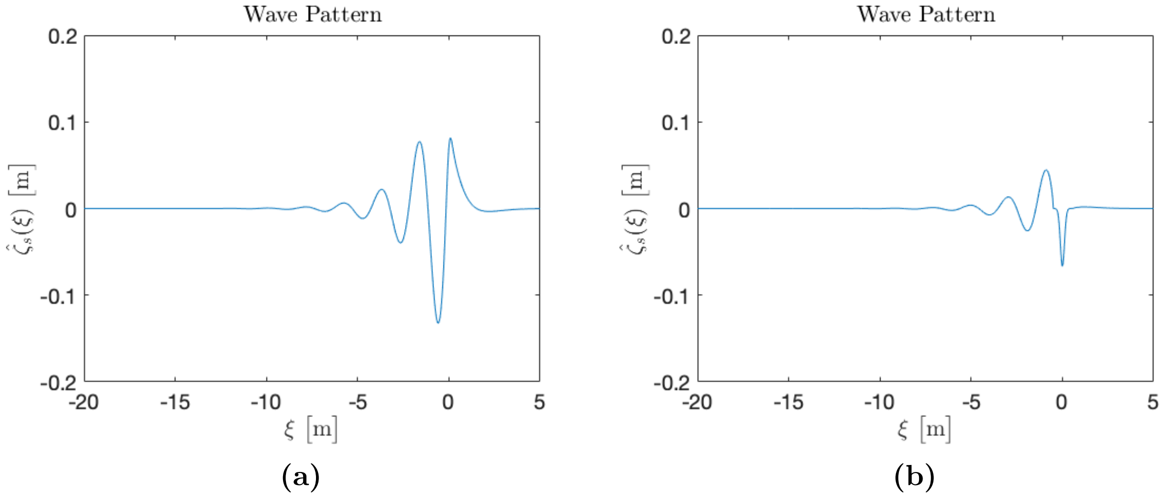


Figure 53: Plots showing the wave pattern for $Fr = 0.5$, $Fr_h = 0.9$, $Fr_s = 0$ for (a) a Gaussian pressure source, (b) the calculated pressure source.

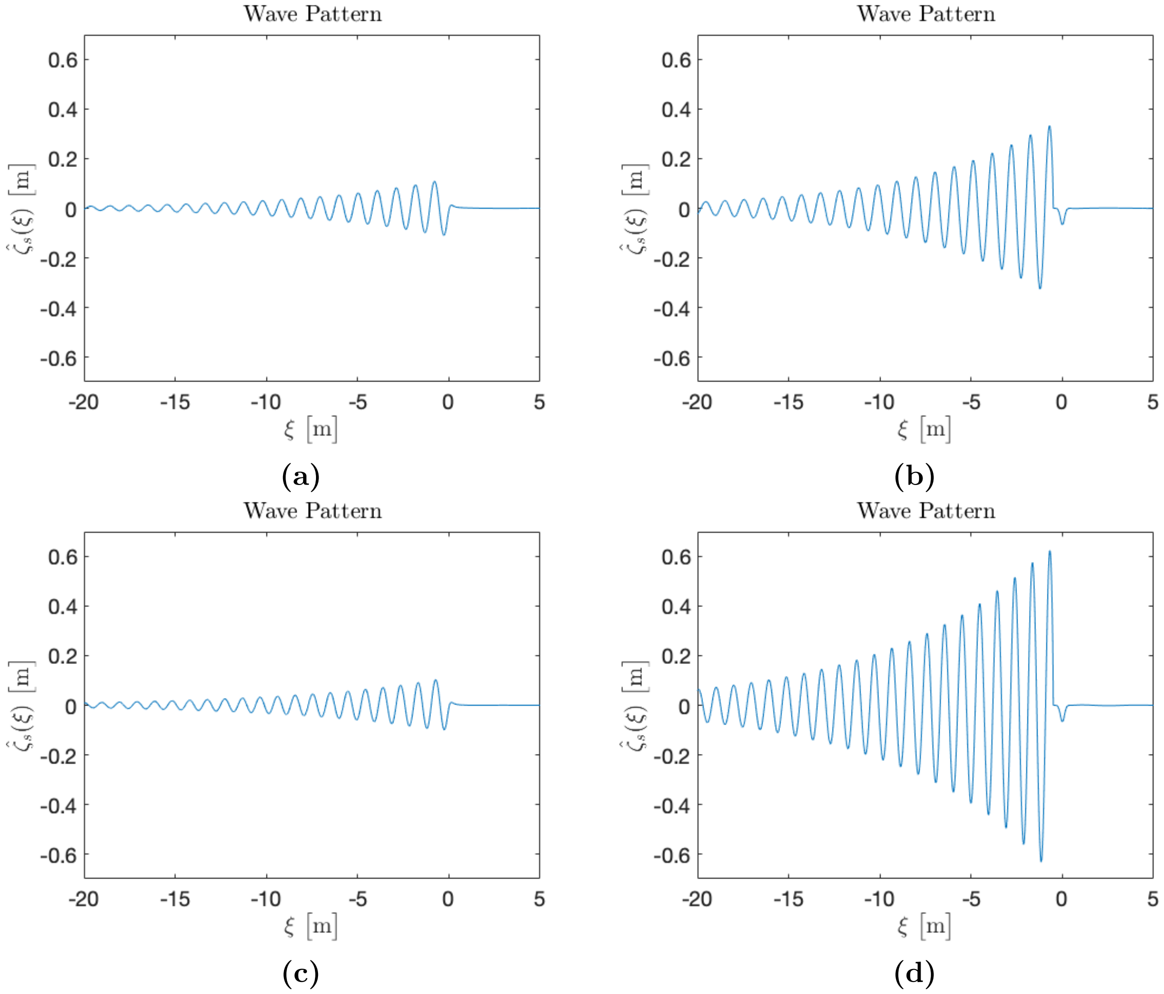
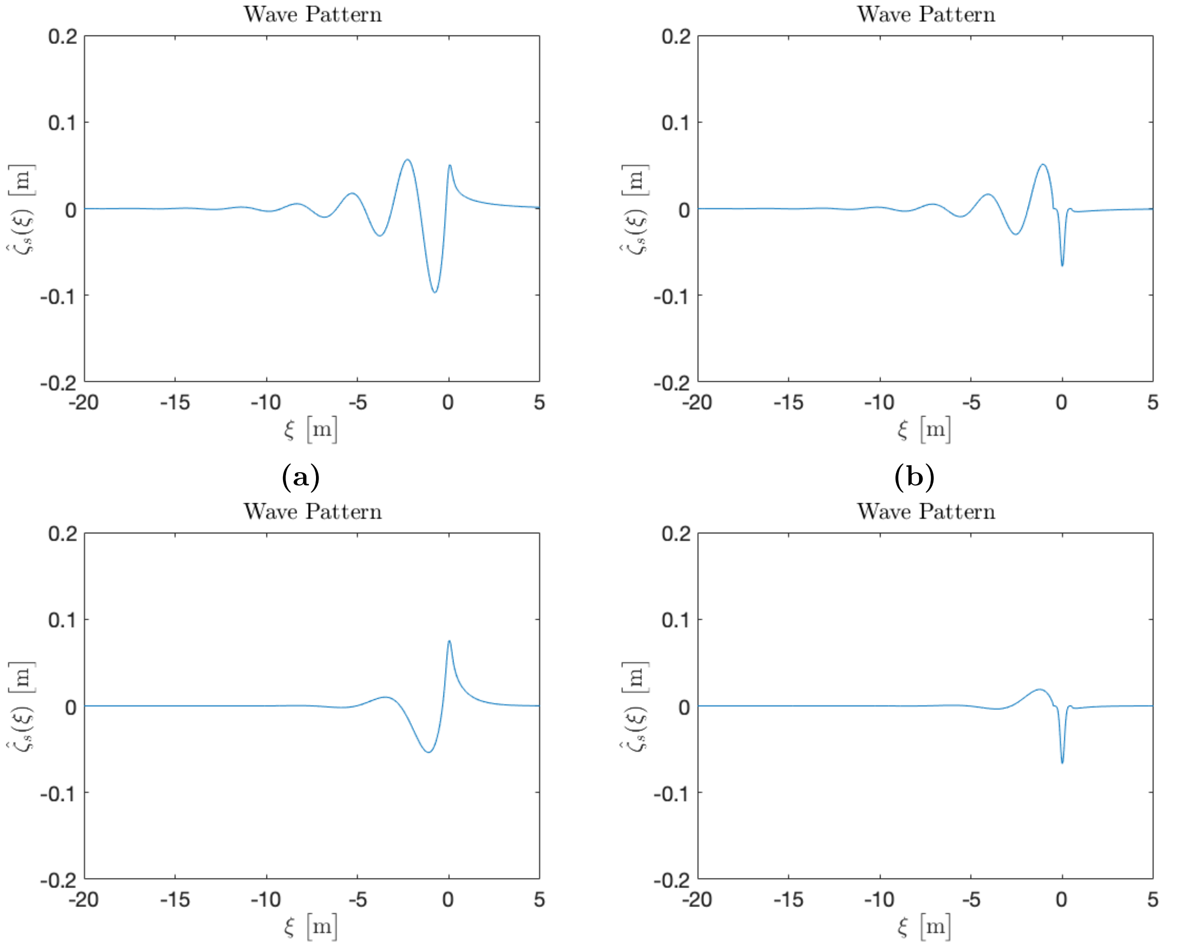


Figure 54: Plots showing the wave pattern of a shear-assisted system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for (a,b) a linear shear profile, (c,d) a exponential shear profile, where (a,c) is a Gaussian pressure source, (b,d) is the calculated pressure source.



(a)

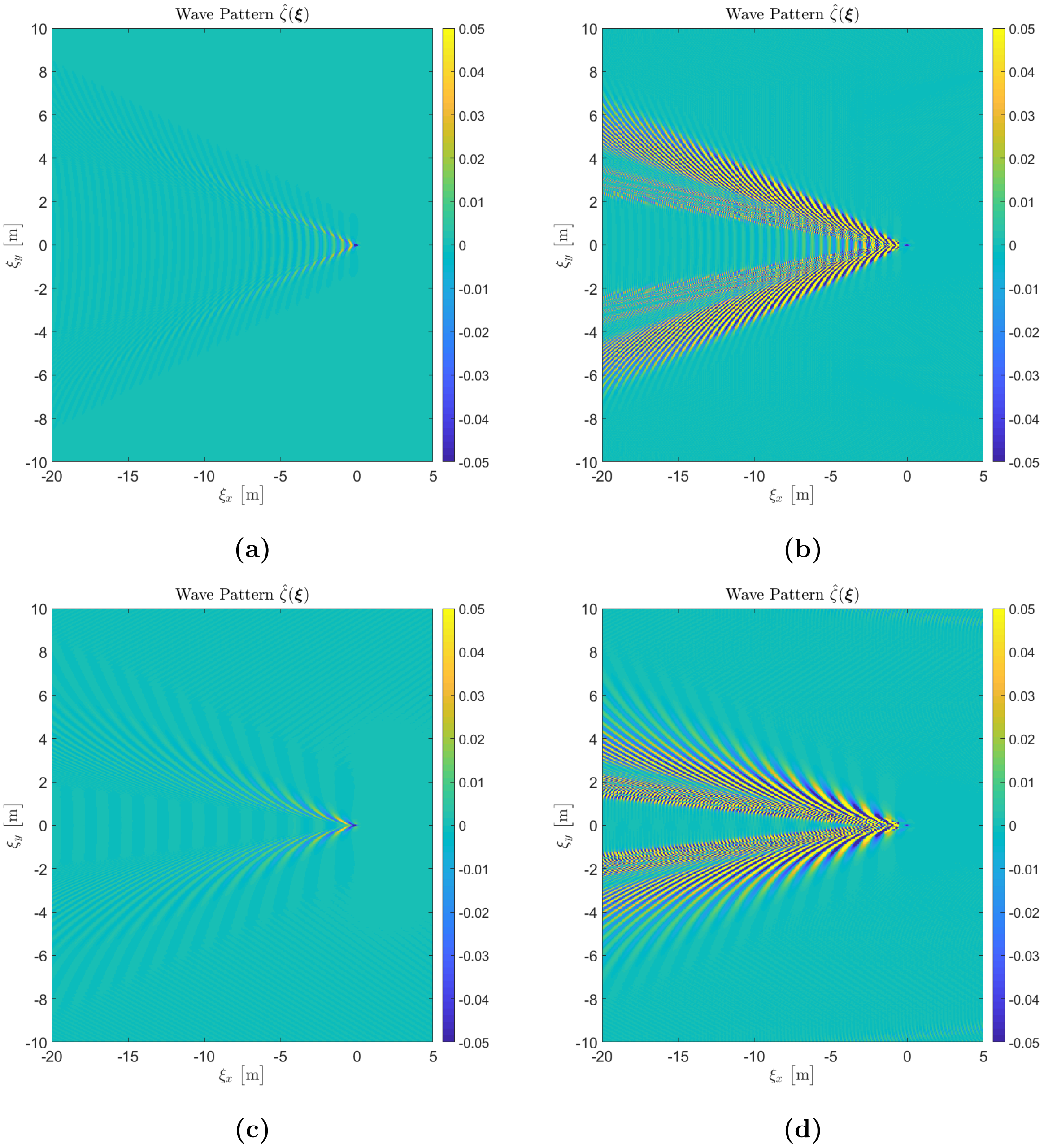
(b)

(c)

(d)

Figure 55: Plots showing the wave pattern of a shear-inhibited system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for (a,b) a linear shear profile, (c,d) a exponential shear profile, where (a,c) is a Gaussian pressure source, (b,d) is the calculated pressure source.

7.1.2 3D



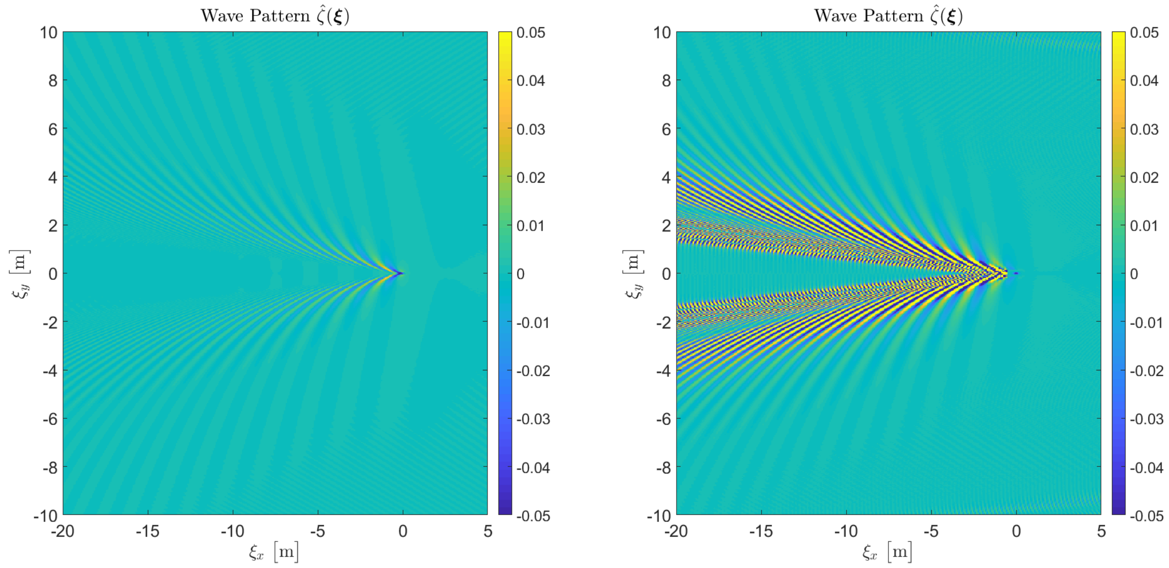
(a)

(b)

(c)

(d)

Figure 56: Plots showing the wave pattern for (a,b) $Fr = 0.3$, $Fr_h = 0.1$, $Fr_s = 0$, (c,d) $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0$, where (a,c) is a Gaussian pressure source, and (b,d) is the calculated pressure source.



(a)

(b)

Figure 57: Plots showing the wave pattern for $Fr = 0.5$, $Fr_h = 0.9$, $Fr_s = 0$ for (a) a Gaussian pressure source, (b) the calculated pressure source.

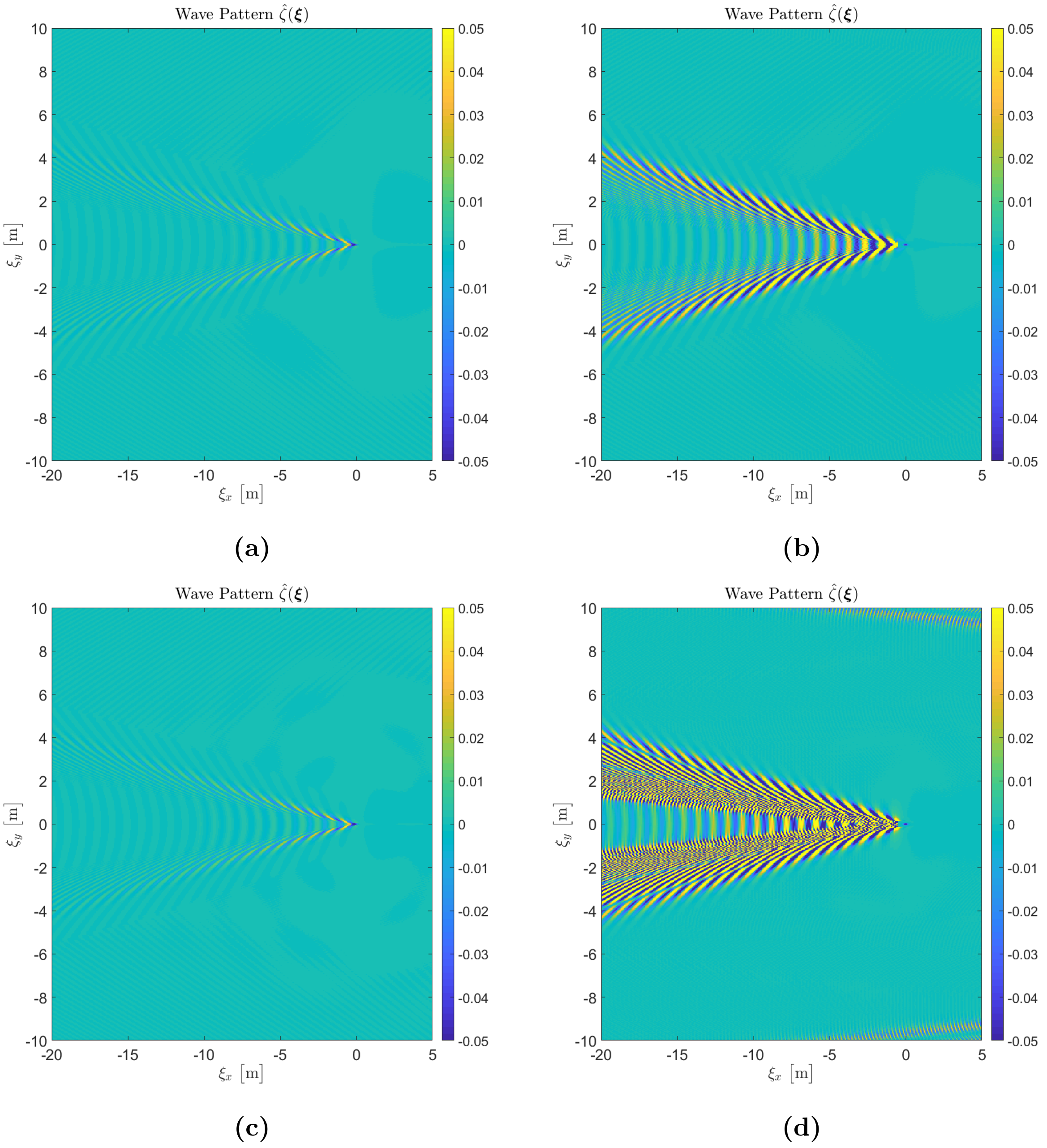
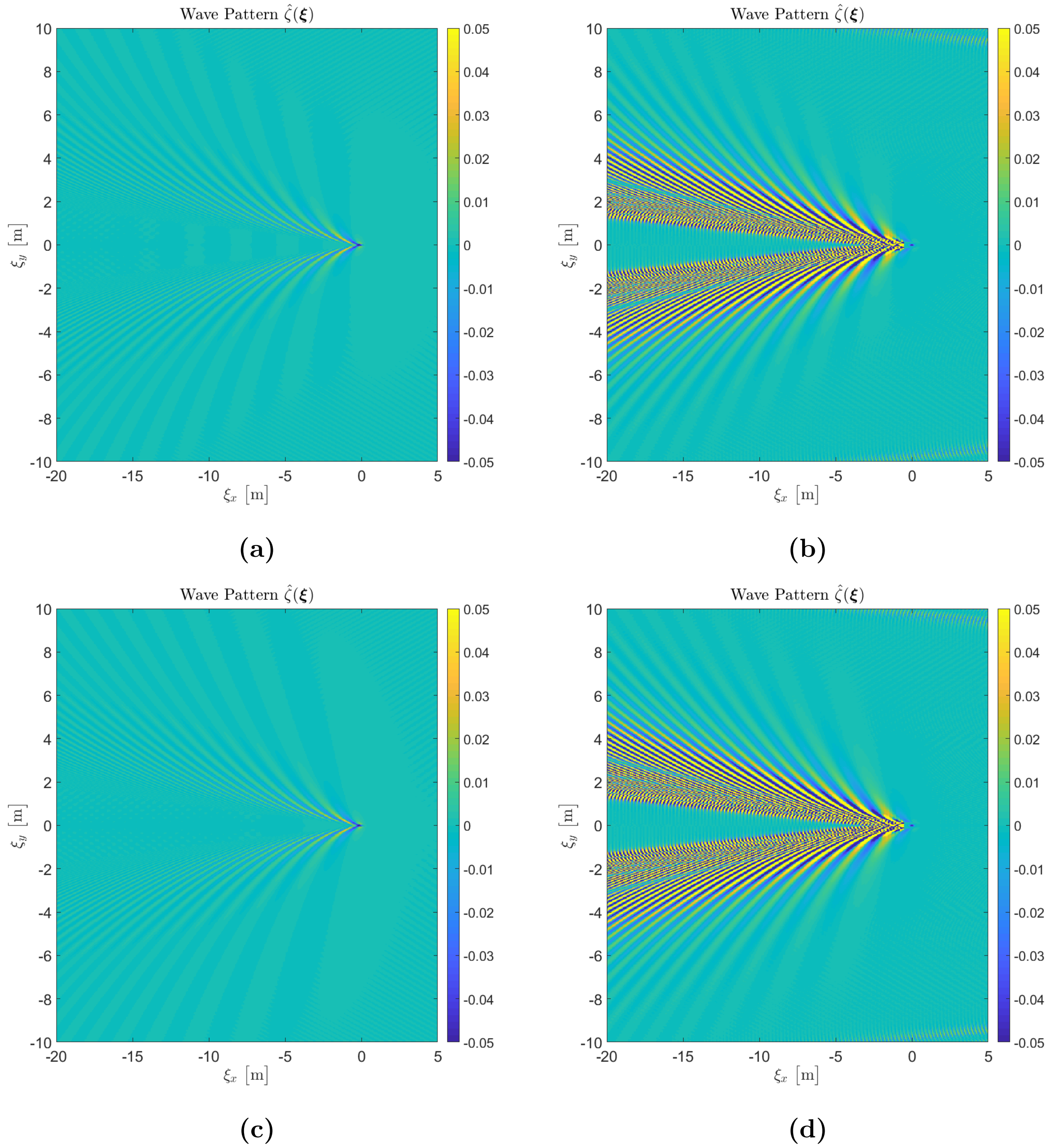


Figure 58: Plots showing the wave pattern of a shear-assisted system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for (a,b) a linear shear profile, (c,d) a exponential shear profile, where (a,c) is a Gaussian pressure source, (b,d) is the calculated pressure source.



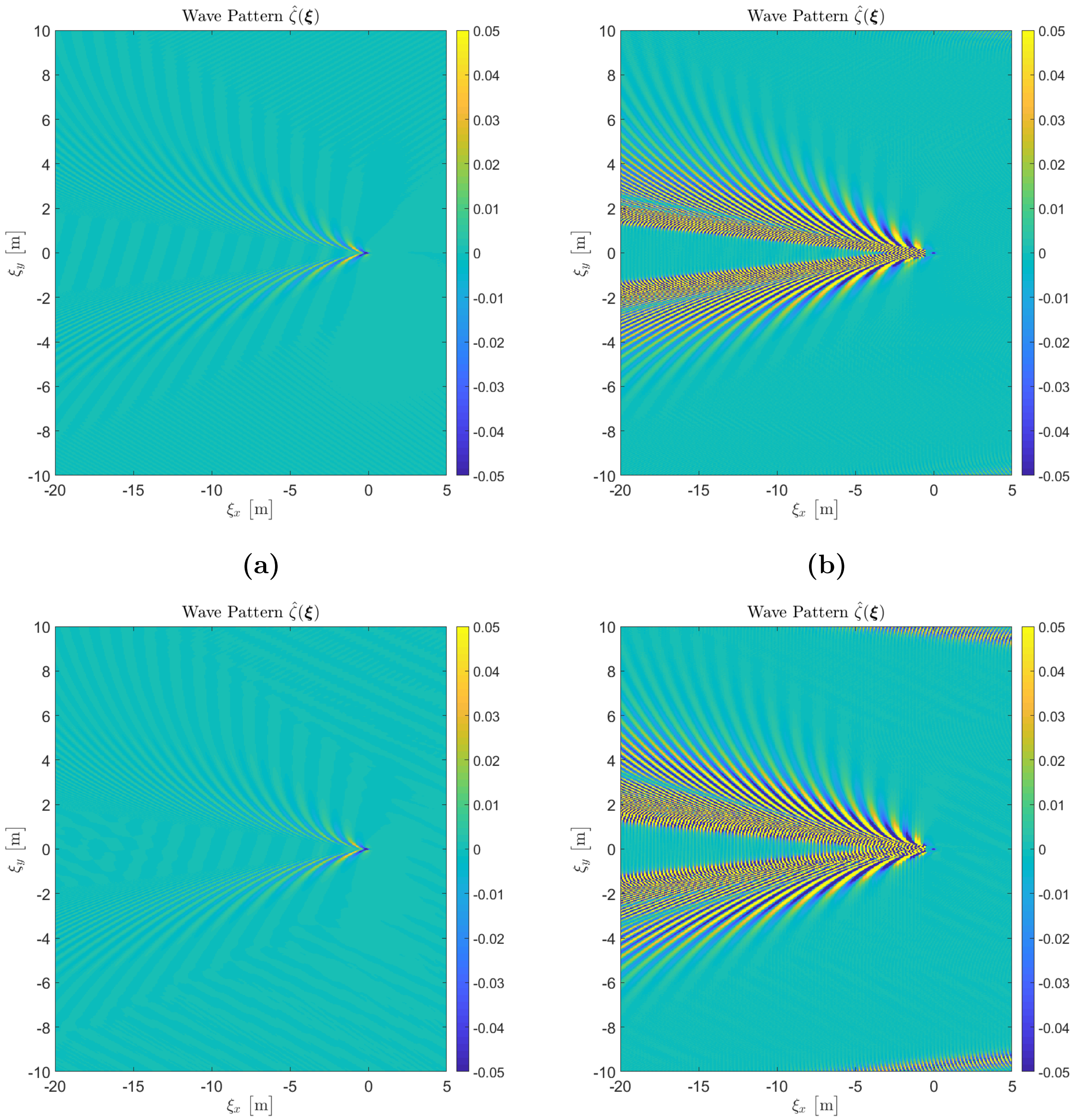
(a)

(b)

(c)

(d)

Figure 59: Plots showing the wave pattern of a shear-inhibited system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for (a,b) a linear shear profile, (c,d) a exponential shear profile, where (a,c) is a Gaussian pressure source, (b,d) is the calculated pressure source.



(a)

(b)

(c)

(d)

Figure 60: Plots showing the wave pattern of a side-on shear system with $Fr = 0.5$, $Fr_h = 0.1$, $Fr_s = 0.5$ for (a,b) a linear shear profile, (c,d) a exponential shear profile, where (a,c) is a Gaussian pressure source, (b,d) is the calculated pressure source.

8 Appendix B

8.1 Example Script for 2D

```
% Example script for arbitrary shear
g = 9.81;                       % Gravitational acceleration

Fr = 0.5;                       % Froude number

L = 1;                          % Length of ship

V = Fr*sqrt(g*L);               % Speed of moving ship

Frh = 0.1;                      % Height Froude number

h = ((V/Frh)^2)/g;              % Water depth

nz = 200;                       % Number of points in z-direction

z = linspace(0,-h,nz)';         % z-vector

dz = -z(2);                     % z-step

Frs = 0;                        % Shear Froude number

direction = 'Assisted';         % Direction of the shear current, can
                                % choose between 'Assisted' and
                                % 'Inhibited'

shearProfile = 'Exponential';   % Type of shear profile, can choose
                                % between 'Exponential' and 'Linear'

[U,dU0,ddU] = createArbitraryShear2D(Fr,Frs,L,z,direction,shearProfile);

n_ship = 2^6;                   % Number of points on the ship

nx = 2^15;                      % Number of points in the whole domain

epsilon = 2;                    % The radiation condition number

incPlt = false;                 % Include plot of ship?

zeta_s = createShip2D(L,n_ship,incPlt);     % Ship geometry

PltOpt = 'None';                % Plotting options for the results,
                                % options: 'All', 'Essentials', 'None'

p_ext = -g*zeta_s;          % Gaussian pressure source usually used
                            % for constant pressure source cases
OptIn = {p_ext,'Pressure'}; % One example of optional input to the
                            % solver. Other options are 'Greens' with
                            % a Green's function matrix

[Wavelength_t,Ig,p_ext,zeta,dk,k_max,zeta_error,maxAmp] = ...
    solvePressureShip2D(z,L,n_ship,nx,epsilon,V,U,dU0,ddU,shearProfile,...
    zeta_s,PltOpt);
```

```
dx = 0.015873
dk = 0.01208
k_max = 197.9083
Error = -0.071096
maxAmp = 0.14488
Wavelength = 1.5708
No plots for you
```

8.2 2D solver

```
% Mandatory Input: z-vector, length of ship, number of points on ship,
% number of points on ship , radiation condition parameter, ship
% velocity, shear current, shear current derivative at surface,
% souble derivative of shear current, shear profile

% Optional Input:

% zeta_s (hull shape)

% PltOpt (Plotting options) = 'None', 'Essentials' or 'All'

% OptIn (Optional input) = {data,'Greens' or 'Pressure'}
% (Green's function or external pressure distribution)

% Example of Green's function input:
% load('G_ij.mat')
% OptIn = {G_ij,'Greens'};

% Example of pressure input:
% load('Gauss_p.mat')
% OptIn = {p_ext,'Pressure'};

function [Wavelength_t,Ig,p_ext,zeta,dk,k_max,zeta_error,maxAmp] = ...
    solvePressureShip2D(z,L,n_ship,nx,epsilon,V,U,dU0,ddU,...
    shearProfile,zeta_s,PltOpt,OptIn)

switch nargin
    case 13
        switch OptIn{2}
            case 'Pressure'
                hasG_ij = false;
                hasp_ext = true;
                p_ext = OptIn{1};
            case 'Greens'
                hasG_ij = true;
                hasp_ext = false;
                G = OptIn{1};
            otherwise
                error("Invalid optional input. The input needs to be either
                'Greens' or 'Pressure'.")
        end
        hasship = true;
        n_ship_test = length(zeta_s);
        if n_ship_test ~= n_ship
            n_ship = n_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end

    case 12
        hasG_ij = false;
        hasp_ext = false;
        hasship = true;
        n_ship_test = length(zeta_s);
        if n_ship_test ~= n_ship
            n_ship = n_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end

    case 11
        hasG_ij = false;
        hasp_ext = false;
        hasship = true;
        PltOpt = 'None';
        n_ship_test = length(zeta_s);
        if n_ship_test ~= n_ship
            n_ship = n_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end

    case 10
```

```
        hasG_ij = false;
        hasp_ext = false;
        hasship = false;
        PltOpt = 'None';

    otherwise
        error("Invalid number of input.")
end



% Initialize
g = 9.81;                 % Gravitational acceleration
h = -z(end);              % Water depth
l = L/2;                  % Length of half the ship

% Size of small domain
x_L = -l;                         % Lowest value of xi while in ship region
x_H = l;                          % Highest value of xi while in ship region
xi_ship = linspace(x_L,x_H,n_ship);     % xi for the ship region
dx = xi_ship(2) - xi_ship(1);

% Size of big domain domain
x_L2 = -dx*((nx-1)/2);      % Lowest value of xi in the whole domain
x_H2 = dx*((nx-1)/2);       % Highest value of xi in the whole domain
xi = linspace(x_L2, x_H2, nx);  % xi for the whole domain

% Useful relations between big domain and ship
nx_0 = nx/2;                          % The integer at which xi = 0 is
ship_dom = (nx_0-(n_ship/2 - 1)):(nx_0+(n_ship/2));     % Ship domain

% Shape of hull
if ~hasship
    zeta_s = -(1/15)*exp(-((pi/l)*xi_ship).^2); % Create the a standard
                                                % ship geometry
end
zeta_s = zeta_s(:);                   % Creating the ship geometry vector

% Fourier space
[k, dk] = build_k_mesh2D(nx,dx);
k_shift = fftshift(k);
k_max = max(k(:));

% Size of Greens domain
nx_green = n_ship*2-1;
xi_green = linspace(x_L*2,x_H*2,nx_green);
```

```
% Check if Ig calculations are necessary
if dU0 ~= 0 && ~strcmp(shearProfile,'Linear')

    % Calculate vertical velocity
    w = findVel2D(U,V,ddU,k,z);

    % Calculate Ig
    Ig = findIg2D(w,U,V,ddU,k,z);

    clear('w')                  % Clear up memory

    Ig_shift = fftshift(Ig);    % Shift Ig for use in Greens calculations
else
    Ig = 0;
    Ig_shift = 0;
end

% Green's function
if ~hasp_ext
    if ~hasG_ij
        G = greenMatrix(Ig_shift,epsilon,V,dU0,xi_green,k_shift,h);
        hasG_ij = true;
    end

    p_ext = findPressure(G,zeta_s,n_ship,dx);
end

% Finding the wave patter
zeta = forwardProblem(Ig,p_ext,ship_dom,epsilon,V,dU0,k,h);

% Post Processing
wavePattern_ship = zeta(ship_dom);

% Error
zeta_error = mean(abs((wavePattern_ship(:) - zeta_s(:))))/max(zeta_s(:));

% Max amplitude
maxAmp = max(real(zeta(:)));

% Wavelength
Fr = V/sqrt(g*L);
Wavelength_t = findWavelengthShear(Fr,U,ddU,dU0,z,L,h,shearProfile);


% Display important info
```

```
disp(['dx = ', num2str(dx)])
disp(['dk = ', num2str(dk)])
disp(['k_max = ', num2str(k_max)])
disp(['Error = ', num2str(zeta_error)])
disp(['maxAmp = ', num2str(maxAmp)])
disp(['Wavelength = ', num2str(Wavelength_t)])

% Creating Plotting Structures
plottingParam.xi_ship = xi_ship;
plottingParam.xi = xi;
plottingParam.xi_green = xi_green;
plottingParam.l = l;

plottingData.p_ext = p_ext;
plottingData.zeta_s = zeta_s;
plottingData.wavePattern_ship = wavePattern_ship;
plottingData.zeta = zeta;

if hasG_ij
    plottingData.G_ij = G;
end

switch PltOpt
    case 'None'
        disp('No plots for you')
    otherwise
        plottingWaves(plottingParam,plottingData,PltOpt,hasG_ij)
end

end

%%%%%%%%%%%%%%%%% Local Functions %%%%%%%%%%%%%%%%%%%%%%

% Building the Fourier space

% This function builds a mesh in fourier space from a mesh defined in
% real space, using MATLAB's convention for ordering of frequencies.

% Input: nx - number of mesh columns, ny - number of mesh rows
% dx - mesh element width in x-direction

% Output: Wave number, step size in Fourier space

function [k, dk] = build_k_mesh2D(nx,dx)
dk = 2*pi/(nx*dx);
k = [0:floor(nx/2)-1, -ceil(nx/2):-1]*dk;
```

```
end

% Finding the vertical velocity

% Input: Shear current, ship velocity, double derivative of shear
% current, wave number, z-vector

% Output: Vertical velocity in fourier space

function w = findVel2D(U,V,ddU,k,z)
dz = abs(z(2)-z(1));
nz = length(z);
n = length(k);

a_diag = ones(nz-3,1);
c_diag = ones(nz-3,1);
rightSide = zeros(nz-2,1);
rightSide(1) = -1;

w = zeros(n,nz);
w(:,1) = 1;
ddU_dummy = ddU(2:end-1);
U_dummy = U(2:end-1);
w_dummy = zeros(n,nz-2);

for i =  1:n
    b_diag = -2 - dz^2*k(i)^2 - dz^2 * ddU_dummy./(U_dummy-V);
    velMat = diag(a_diag,1) + diag(b_diag) + diag(c_diag,-1);
    w_dummy(i,:) = velMat\rightSide;

end

w(:,2:end-1) = w_dummy;
end

% Finding Ig

% Input: Vertical velocity, shear current, ship velocity, double
% derivative of shear current, wave number, z-vector

% Output: Ig

function Ig = findIg2D(w,U,V,ddU,k,z)
h = -z(end);
n = size(k,2);
```

```
Ig = zeros(size(k));
for i =  1:n
    w_v = w(i,:)’;

    Ig_integrand = (ddU.*w_v.*sinh(k(i)*(z+h)))./((k(i)*(U-V))*cosh(k(i)*h));

    high_var = (abs((k(i)*(z+h))) > 300);

    if sum(high_var(:)) > 0
        if k(i) > 0
            Ig_integrand(high_var) = (ddU(high_var).*w_v(high_var)...
                .*exp(k(i)*z(high_var)))./(k(i)*(U(high_var)-V));
        else
            Ig_integrand(high_var) = (ddU(high_var).*w_v(high_var)...
                .*(-exp(-k(i)*z(high_var))))./(k(i)*(U(high_var)-V));
        end
    end

    null_Var = (k(i)*(U-V)) == 0;
    Ig_integrand(null_Var) = 0;

    Ig(i) = trapz(z,Ig_integrand);
end
end


% Finding Green’s Function Matrix
% Input: Ig, radiation condition parameter, ship velocity, shear
% current derivative at the surface, xi, wave number, water depth

% Output: Green’s function matrix

function G = greenMatrix(Ig,epsilon,V,dU0,xi,k,h)
% Initiate
g = 9.81;
[km,xim] = meshgrid(k,xi);
[Igm,~] = meshgrid(Ig,xi);

% Calculating the denominator Delta_r or Dr
Phi = 2*V*km.*(1+Igm) + dU0*tanh(km*h);
Phi = sign(Phi);
D_r = (1+Igm).*(km*V).^2 + V*(dU0*km).*tanh(km*h) - g*km.*tanh(km*h) ...
    + 1i*epsilon*Phi;

% Green’s Integrand
I = (km.*tanh(km*h).*exp(1i*km.*xim))./((2*pi)*D_r);
```

```
I(:,length(k)/2 + 1) = 0;

G = trapz(k,I,2);
end

% Solving the inverse problem
% Input: Green's function matrix, ship geometry, number of points on
% ship, step size in real space: dx

% Output: Pressure patch

function p_ext = findPressure(G,zeta_s,n_ship,dx)

% Greens Function in regular space
G_ij = zeros(n_ship);                % Green's function matrix

for i = 1:n_ship
    green_dom = n_ship + (i-1):-1:i;
    G_ij(i,:) = G(green_dom);        % Fill in Green's function matrix
end

b = zeta_s/dx;

p_ext = G_ij\b;

p_ext = p_ext';
end

% Forward Problem
% Input: Ig, External Pressure Distribution, ship domain, radiation
% condition parameter, ship velocity, the shear current derivative at
% the surface, wave number, water depth

% Output: Wave pattern

function zeta = forwardProblem(Ig,p_ext,ship_dom,epsilon,V,dU0,k,h)
g = 9.81;
nx = length(k);
p_ext_big = zeros(1,nx);    % Need to define the external pressure
                            % field for the whole domain

p_ext_big(ship_dom) = p_ext;

% Fourier space
P_ext = fft(p_ext_big);
```

```
% Calculating the denominator Delta_r or Dr
Phi = 2*V*k.*(1+Ig) + dU0*tanh(k*h);
Phi = sign(Phi);
D_r = (1+Ig).*(k*V).^2 + V*(dU0*k).*tanh(k*h) - g*k.*tanh(k*h) ...
    + 1i*epsilon*Phi;

% Integrand
I = P_ext.*k.*tanh(k*h)./D_r;
I(1) = 0;

% Solution
zeta = ifft(I);
end

% Calculate the wavelength
% Input: Froude number, Froude Number, shear current, double
% derivative of shear current, z-vector, length of ship, water depth,
% shear current profile

% Output: wavelength

function wavelength = findWavelengthShear(Fr,U,ddU,dU0,z,L,h,shearProfile)
g = 9.81;
V = Fr*sqrt(g*L);

wavelength = 0.01;          % Initial guess
max_error = 1e-10;
max_iter = 100;
error = 1;
iter = 0;
while error > max_error && iter < max_iter
    iter = iter + 1;
    if ~strcmp(shearProfile,'Linear')
        w = find_vel_wavelength_2D(U,V,ddU,wavelength,z);
        w = w(:);
        Ig = find_Ig_wavelength_2D(w,U,V,ddU,wavelength,z);
        dIg = find_dIg_wavelength_2D(w,U,V,ddU,wavelength,z);
    else
        Ig = 0;
        dIg = 0;
    end

    f = (4*pi^2*V^2*(1 + Ig))./wavelength^2 -...
        (2*g*pi*tanh((2*h*pi)/wavelength))/wavelength +...
        (2*dU0*pi*V*tanh((2*h*pi)/wavelength))/wavelength;
    df = -((8*pi^2*V^2*(1 + Ig))/wavelength^3)...
        + (4*g*h*pi^2*sech((2*h*pi)/wavelength)^2)/wavelength^3 ...
        - (4*dU0*h*pi^2*V*sech((2*h*pi)/wavelength)^2)/wavelength^3 ...
        + (2*g*pi*tanh((2*h*pi)/wavelength))/wavelength^2 ...
        - (2*dU0*pi*V*tanh((2*h*pi)/wavelength))/wavelength^2 ...
        + (4*pi^2*V^2*dIg)/wavelength^2;

    wavelength_new = wavelength - f/df;
    error = abs(wavelength_new - wavelength);
    wavelength = wavelength_new;
end

end

% Plotting function
function plottingWaves(plottingParam,plottingData,PltOpt,hasG_ij)

set(0, 'DefaultTextInterpreter', 'latex')
set(0, 'DefaultAxesFontSize', 20)

switch PltOpt
    case 'Essentials'

        figure
        imagesc(plottingParam.xi,real(plottingData.zeta))
        title('Wave Pattern')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}(\xi)$ \big[m\big]')

        figure
        plot(plottingParam.xi,real(plottingData.zeta))
        xlim([-20,5])
        title('Surface Elevation')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}_s(\xi)$ \big[m\big]')

    case 'All'
        figure
        plot(plottingParam.xi_ship,real(plottingData.p_ext))
        title('External Pressure Distribution')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{p}_{ext}(\xi)$ \big[m\textsuperscript{2}/s\textsuperscript{2}]'

        figure
        plot(plottingParam.xi_ship,real(plottingData.p_ext))
        ylim([-3,1])
        title('External Pressure Distribution')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{p}_{ext}(\xi)$ \big[m\textsuperscript{2}/s\textsuperscript{2}]'

        figure
        plot(plottingParam.xi_ship,real(plottingData.zeta_s))
        ylim([-0.07,0.01])
        title('Prescribed Surface Elevation')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}_s(\xi)$ \big[m\big]')

        figure
        plot(plottingParam.xi_ship,real(plottingData.wavePattern_ship))
        ylim([-0.2,0.2])
        title('Ship results')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}(\xi)$ \big[m\big]')

        figure
        plot(plottingParam.xi,real(plottingData.zeta))
        title('Wave Pattern')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}(\xi)$ \big[m\big]')

        figure
        plot(plottingParam.xi,real(plottingData.zeta))
        xlim([-20,5])
        ylim([-0.7,0.7])
        title('Wave Pattern')
        xlabel('$\xi$ \big[m\big]')
        ylabel('$\hat{\zeta}_s(\xi)$ \big[m\big]')


        if hasG_ij
            figure
            plot(plottingParam.xi_green,real(plottingData.G_ij))
            ylim([-3,3])
            title("Green's Function")
            xlabel('$\xi$ \big[m\big]')
            ylabel('$G(\xi)$ \big[s\textsuperscript{2}/m\textsuperscript{3}\big]')
        end
end

end
```

8.3 Example Script for 3D

```
% Example script for arbitrary shear
g = 9.81;               % Gravitational acceleration

Fr = 0.5;               % Froude number

L = 1;                  % Length of ship

V = Fr*sqrt(g*L);       % Speed of moving ship

Vx = V;                 % Speed of moving ship in x-direction

Vy = 0;                 % Speed of moving ship in y-direction

Frh = 0.1;              % Height Froude number

h = ((V/Frh)^2)/g;      % Water depth

nz = 200;               % Number of points in z-direction

z = linspace(0,-h,nz)'; % z-vector

dz = -z(2);             % z-step

Frs = 0;                % Shear Froude number

direction = 'SideOn';   % Direction of the shear current, can choose
                        % between 'Assisted', 'Inhibited', and
                        % 'SideOn'

shearProfile = 'Exponential';   % Type of shear profile, can choose
                                % between 'Exponential' and 'Linear'

[Ux,Uy,dUx0,dUy0,ddUx,ddUy] = ...
    createArbitraryShear(Fr,Frs,L,z,direction,shearProfile);

nx_ship = 2^6;          % Number of points on the ship in x-direction

nx = 2^13;              % Number of points in the whole domain

AR = 0.25;              % The aspect ratio

epsilon = 2;            % The radiation condition number

incPlt = false;         % Include plot of ship?
```

```
zeta_s = createShip(L,AR,nx_ship,incPlt);   % Ship geometry

PltOpt = ’None’;         % Plotting options for the results,
                        % options: ’All’, ’Essentials’, ’None’

p_ext = -g*zeta_s;      % Gaussian pressure source usually used
                        % for constant pressure source cases

OptIn = {p_ext,’Pressure’}; % One example of optional input to the
                            % solver. Other options are ’Greens’ with
                            % a Green’s function matrix

[wavePattern_ship,Ig,p_ext,zeta,dkx,dky,k_max,zeta_error,maxAmp] = ...
    solvePressureShipParallel(z,L,AR,nx_ship,nx,epsilon,Vx,Vy,Ux,Uy,dUx0,...
    dUy0,ddUx,ddUy,shearProfile,zeta_s,PltOpt);
```

```
dx = 0.015873
dy = 0.016667
dkx = 0.04832
dky = 0.18408
k_max = 273.3186
Error = -57387.7417
maxAmp = 0.62602
No plots for you
```

8.4 3D solver

```
% Mandatory Input: z-vector, length of ship, aspect ratio, number of
% points on ship in x-direction on ship, number of points on ship in
% x-direction, radiation condition parameter, ship velocity in x-
% direction, ship velocity in y-direction, shear current in x-
% direction, shear current in y-direction, derivative of shear current
% at surface in x-direction, derivative of shear current at surface in
% y-direction, double derivative of shear current in x-direction,
% double derivative of shear current in y-direction, shear profile

% Optional Input:

% zeta_s (hull shape)

% PltOpt (Plotting options) = ’None’, ’Essentials’ or ’All’

% OptIn (Optional input) = {data,’Greens’ or ’Pressure’}
% (Green’s function or external pressure distribution)
% Example of Green's function input:
% load('G_ij.mat')
% OptIn = {G_ij,'Greens'};

% Example of pressure input:
% load('Gauss_p.mat')
% OptIn = {p_ext,'Pressure'};

function [wavePattern_ship,Ig,p_ext,zeta,dkx,dky,k_max,zeta_error,maxAmp]...
    = solvePressureShipParallel(z,L,AR,nx_ship,nx,epsilon,Vx,Vy,Ux,Uy,...
    dUx0,dUy0,ddUx,ddUy,shearProfile,zeta_s,PltOpt,OptIn)

switch nargin
    case 18
        switch OptIn{2}
            case 'Pressure'
                hasG_ij = false;
                hasp_ext = true;
                p_ext = OptIn{1};
            case 'Greens'
                hasG_ij = true;
                hasp_ext = false;
                G_ij = OptIn{1};
            otherwise
                error("Invalid optional input. The input needs to be either
                'Greens' or 'Pressure'.")
        end
        hasship = true;
        nx_ship_test = size(zeta_s,2);
        if nx_ship_test ~= nx_ship
            nx_ship = nx_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end
        AR_test = nx_ship/size(zeta_s,1);
        if AR_test ~= AR
            AR = AR_test;
            disp('Changed AR to the aspect ratio of the given ship')
        end

    case 17
        hasG_ij = false;
        hasp_ext = false;
        hasship = true;
        nx_ship_test = size(zeta_s,2);
        if nx_ship_test ~= nx_ship
            nx_ship = nx_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end
        AR_test = nx_ship/size(zeta_s,1);
        if AR_test ~= AR
            AR = AR_test;
            disp('Changed AR to the aspect ratio of the given ship')
        end

    case 16
        hasG_ij = false;
        hasp_ext = false;
        hasship = true;
        PltOpt = 'None';
        nx_ship_test = size(zeta_s,2);
        if nx_ship_test ~= nx_ship
            nx_ship = nx_ship_test;
            disp('Changed nx_ship to the dimension of the given ship')
        end
        AR_test = nx_ship/size(zeta_s,1);
        if AR_test ~= AR
            AR = AR_test;
            disp('Changed AR to the aspect ratio of the given ship')
        end

    case 15
        hasG_ij = false;
        hasp_ext = false;
        hasship = false;
        PltOpt = 'None';

    otherwise
        error("Invalid number of input.")
end



% Initialize
h = -z(end);                % Water depth
l = L/2;                    % Length of half the ship
W = L*AR;                   % Width of ship
w_half = W/2;               % Half the width of the ship

% Size of small domain
x_L = -l;
x_H = l;
xi_ship_x_v = linspace(x_L,x_H,nx_ship);
```

```
dx = xi_ship_x_v(2) - xi_ship_x_v(1);

ny_ship = 2^nextpow2(nx_ship*AR);
y_L = -w_half;
y_H = w_half;
xi_ship_y_v = linspace(y_L,y_H,ny_ship);
dy = xi_ship_y_v(2) - xi_ship_y_v(1);

[xi_ship_x,xi_ship_y] = meshgrid(xi_ship_x_v,xi_ship_y_v);

% Size of big domain domain
ny = 2^nextpow2(nx*AR);
x_L2 = -dx*((nx-1)/2);
x_H2 = dx*((nx-1)/2);
y_L2 = -dy*((ny-1)/2);
y_H2 = dy*((ny-1)/2);
xi_x_v = linspace(x_L2, x_H2, nx);
xi_y_v = linspace(y_L2, y_H2, ny);

% Useful relations between big domain and ship
nx_0 = nx/2;            % The integer at which xi = 0 is
ny_0 = ny/2;            % The integer at which xi = 0 is
ship_dom_x = (nx_0-(nx_ship/2 - 1)):(nx_0+(nx_ship/2));
ship_dom_y = (ny_0-(ny_ship/2 - 1)):(ny_0+(ny_ship/2));

% Shape of hull
if ~hasship
    zeta_s = -(1/15)*exp(-(((pi/l)*xi_ship_x).^2 + ((pi/w_half)*xi_ship_y).^2));
end
zeta_s_flipped = flip(zeta_s);
zeta_s_v = zeta_s_flipped(:);

% Fourier space
[kx, ky, dkx, dky] = build_k_mesh(nx,ny,dx,dy);
kx_shift = fftshift(kx);
ky_shift = fftshift(ky);
kx_v_shift = kx_shift(1,:);
ky_v_shift = ky_shift(:,1)';
k = sqrt((kx.^2)+(ky.^2));
k_shift = sqrt((kx_shift.^2)+(ky_shift.^2));
k_max = max(k(:));

% Size of Greens domain
nx_green = nx_ship*2-1;
ny_green = ny_ship*2-1;
xi_green_x_v = linspace(x_L*2,x_H*2,nx_green);
```

```
xi_green_y_v = linspace(y_L*2,y_H*2,ny_green);
[xi_green_x,xi_green_y] = meshgrid(xi_green_x_v,xi_green_y_v);

dU0 = sqrt(dUx0^2 + dUy0^2);

% Check if Ig calculations are necessary
if dU0 ~= 0 && ~strcmp(shearProfile,'Linear')
    % Calculate vertical velocity
    w = findVel(Ux,Uy,Vx,Vy,ddUx,ddUy,k,kx,ky,z);

    % Calculate Ig
    Ig = findIg(w,Ux,Uy,Vx,Vy,ddUx,ddUy,k,kx,ky,z);
    clear('w')
    Ig_shift = fftshift(Ig);
else
    Ig = 0;
    Ig_shift = 0;
end

% Green's function
if ~hasp_ext
    if ~hasG_ij
        tic
        G_ij = greenMatrix(Ig_shift,nx,ny,nx_green,ny_green,epsilon,Vx,...
            Vy,dUx0,dUy0,xi_green_x,xi_green_y,kx_shift,ky_shift,k_shift,...
            kx_v_shift,ky_v_shift,h);
        toc
        hasG_ij = true;
    end

    p_ext = findPressure(G_ij,zeta_s_v,nx_ship,ny_ship,nx_green,...
        ny_green,dx,dy);
end

% Finding the wave patter
zeta = forwardProblem(Ig,p_ext,ship_dom_x,ship_dom_y,epsilon,Vx,Vy,dUx0,...
    dUy0,kx,ky,k,h);

% Post Processing
wavePattern_ship = zeta(ship_dom_y,ship_dom_x);

% Error
zeta_error = mean(abs((wavePattern_ship(:) - zeta_s(:))))/max(zeta_s(:));

maxAmp = max(real(zeta(:)));
```

```
% Display important info
disp(['dx = ', num2str(dx)])
disp(['dy = ', num2str(dy)])
disp(['dkx = ', num2str(dkx)])
disp(['dky = ', num2str(dky)])
disp(['k_max = ', num2str(k_max)])
disp(['Error = ', num2str(zeta_error)])
disp(['maxAmp = ', num2str(maxAmp)])

% Creating Plotting Structures
plottingParam.xi_ship_x_v = xi_ship_x_v;
plottingParam.xi_ship_y_v = xi_ship_y_v;
plottingParam.xi_x_v = xi_x_v;
plottingParam.xi_y_v = xi_y_v;
plottingParam.xi_green_x_v = xi_green_x_v;
plottingParam.xi_green_y_v = xi_green_y_v;
plottingParam.l = l;

plottingData.p_ext = p_ext;
plottingData.zeta_s = zeta_s;
plottingData.wavePattern_ship = wavePattern_ship;
plottingData.zeta = zeta;
if hasG_ij
    plottingData.G_ij = G_ij;
end

switch PltOpt
    case 'None'
        disp('No plots for you')
    otherwise
        plottingWaves(plottingParam,plottingData,PltOpt,hasG_ij)
end

end

%%%%%%%%%%%%%%%%% Local Functions %%%%%%%%%%%%%%%%%%%%%%

% Building the Fourier space

%This function builds a mesh in fourier space from a mesh defined in real
%space, using MATLAB's convention for ordering of frequencies.

% Input: nx - number of mesh columns, ny - number of mesh rows,
% dx - mesh element width in x-direction, dy - mesh element width
% in y-direction
% Output: Wave number in x-direction, wave number in y-direction,
% step size in x-direction, step size in y-direction

function [kx, ky, dkx, dky] = build_k_mesh(nx,ny,dx,dy)

dkx = 2*pi/(nx*dx);
dky = 2*pi/(ny*dy);

kxv = [0:floor(nx/2)-1, -ceil(nx/2):-1]*dkx;
kyv = [0:floor(ny/2)-1, -ceil(ny/2):-1]*dky;
[kx, ky] = meshgrid(kxv,kyv);

end

% Finding the vertical velocity

% Input: Shear current in x-direction, shear current in y-direction,
% ship velocity in x-direction, ship velocity in y-direction, double
% derivative of shear current in x-direction, double derivative of
% shear current in y-direction, wave number, wave number in
% x-direction, wave number in y-direction, z-vector

% Output: Vertical velocity in fourier space

function w = findVel(Ux,Uy,Vx,Vy,ddUx,ddUy,k,kx,ky,z)
dz = abs(z(2)-z(1));
h = -z(end);
n = length(z);
ny = size(k,1);
nx = size(k,2);

a_diag = ones(n-3,1);
c_diag = ones(n-3,1);
rightSide = zeros(n-2,1);
rightSide(1) = -1;
rightSide(end) = 0;

w = zeros(size(k,1),size(k,2),n);
w(:,:,1) = 1;
ddUx_dummy = ddUx(2:end-1);
ddUy_dummy = ddUy(2:end-1);
Ux_dummy = Ux(2:end-1);
Uy_dummy = Uy(2:end-1);
z_dummy = z(2:end-1);
w_dummy = zeros(size(k,1),size(k,2),n-2);
```

```
parfor i = 1:ny
    for j =  1:nx
        if k(i,j)==0
            w_dummy(i,j,:) = z_dummy/h + 1;
        else
            nullVar = abs(kx(i,j)*(Ux_dummy-Vx)...
                + ky(i,j)*(Uy_dummy-Vy)) < 1e-6;

            b_diag = -2 - dz^2*k(i,j)^2 - dz^2 * (kx(i,j)*ddUx_dummy ...
                + ky(i,j)*ddUy_dummy)./(kx(i,j)*(Ux_dummy-Vx)...
                + ky(i,j)*(Uy_dummy-Vy));
            b_diag(nullVar) = - 2 -dz^2*k(i,j)^2;

            velMat = diag(a_diag,1) + diag(b_diag) + diag(c_diag,-1);
            w_dummy(i,j,:) = velMat\rightSide;
        end

    end
end

w(:,:,2:end-1) = w_dummy;
end

% Finding Ig

% Input: Vertical velocity, shear current in x-direction, shear current
% in y-direction, ship velocity in x-direction, ship velocity in
% y-direction, double derivative of shear current in x-direction,
% double derivative of shear current in y-direction, wave number,
% wave number in x-direction, wave number in y-direction, z-vector

% Output: Ig

function Ig = findIg(w,Ux,Uy,Vx,Vy,ddUx,ddUy,k,kx,ky,z)
h = -z(end);
ny = size(k,1);
nx = size(k,2);

Ig = zeros(size(k));
parfor i = 1:ny
    for j =  1:nx
        w_v = w(i,j,:);
        w_v = w_v(:);
        high_var = (k(i,j)*(z+h)) > 300;
        Ig_integrand = ((kx(i,j)*ddUx + ky(i,j)*ddUy).*w_v.*sinh(k(i,j)*(z+h)))./...
            ((k(i,j)*(kx(i,j)*(Ux-Vx) + ky(i,j)*(Uy-Vy)))*cosh(k(i,j)*h));
        Ig_integrand(high_var) = ((kx(i,j)*ddUx(high_var) + ky(i,j)...
            *ddUy(high_var)).*w_v(high_var).*exp(k(i,j)*z(high_var)))./...
            (k(i,j)*(kx(i,j)*(Ux(high_var)-Vx) + ky(i,j)*(Uy(high_var)-Vy)));

        null_Var = (kx(i,j)*(Ux-Vx) + ky(i,j)*(Uy-Vy))==0;
        Ig_integrand(null_Var) = 0;

        Ig(i,j) = trapz(z,Ig_integrand);
    end
end
end


% Finding Green's Function Matrix
% Input: Ig, number of points in x-direction whole domain, number of
% points in y-direction whole domain, number of points in x-direction
% Green's matrix, number of points in y-direction Green's matrix,
% radiation condition parameter, ship velocity in x-direction, ship
% velocity in y-direction , shear current derivative at the surface in
% x-direction, xi in x-dierction, xi in y-dierction, wave number in
% x-direction, wave number in y-direction, wave number, k_x vector
% k_y vector, water depth

% Output: Green's function matrix

% Finding Green's Function Matrix
function G_ij = greenMatrix(Ig,nx,ny,nx_green,ny_green,epsilon,Vx,Vy,dUx0,...
    dUy0,xi_x,xi_y,kx,ky,k,kx_v,ky_v,h)
% Initiate
g = 9.81;
G_ij = zeros(ny_green,nx_green);

% Calculating the denominator Delta_r or Dr
Phi = 2*(kx*Vx + ky*Vy).*(1+Ig) + (kx*dUx0 + ky*dUy0).*tanh(k*h)./k;
Phi((ny/2)+1,(nx/2)+1) = 0;
Phi = sign(Phi);
D_r = (1+Ig).*(kx*Vx + ky*Vy).^2 + (kx*Vx + ky*Vy).*(kx*dUx0 + ky*dUy0)...
    .*tanh(k*h)./k - g*k.*tanh(k*h) + 1i*epsilon*Phi;

% Green's Integrand without exponential
I_we = k.*tanh(k*h)./(((2*pi)^2)*D_r);
parfor j = 1:nx_green
    for i = 1:ny_green
        I = I_we .* exp(1i*(xi_x(i,j)*kx + xi_y(i,j)*ky));
        I((ny/2)+1,(nx/2)+1) = 0;
        % Greens Function in regular space
        G_ij(i,j) = trapz(ky_v,trapz(kx_v,I,2));
    end
end

end

% Solving the inverse problem
% Input: Green's function matrix, ship geometry, number of points on
% ship in x-direction, number of points on ship in y-direction,
% number of points in x-direction Green's matrix, number of points in
% y-direction Green's matrix, step size in real space x-direction,
% step size in real space y-direction

% Output: Pressure patch
function p_ext = findPressure(G_ij,zeta_s_v,nx_ship,ny_ship,nx_green,...
    ny_green,dx,dy)
% Creating a big 2D Matrix for G
G_ijkt = zeros(nx_ship*ny_ship);
nx_green_0 = (nx_green+1)/2;
ny_green_0 = (ny_green+1)/2;

for j = 1:nx_ship
    for i = 1:ny_ship
        green_dom_x = (nx_green_0+(j-1)):-1:j;                  % t
        green_dom_y = (ny_green_0-(i-1)):ny_green-(i-1);        % k
        G_dummy = G_ij(green_dom_y,green_dom_x);

        index = ny_ship*(j-1) + i;
        G_ijkt(index,:) = G_dummy(:);
    end
end

A = G_ijkt;

b = zeta_s_v/(dy*dx);

p_ext_v = A\b;

p_ext_v = p_ext_v';

p_ext = reshape(p_ext_v,[ny_ship,nx_ship]);

p_ext = flip(p_ext);
end
```

```
% Forward Problem
% Input: Ig, External Pressure Distribution, ship domain in x-direction,
% ship domain in y-direction, radiation condition parameter,
% ship velocity in x-direction, ship velocity in y-direction,
% shear current derivative at the surface in x-direction, shear current
% derivative at the surface in x-direction, wave number in x-direction,
% wave number in y-direction, wave number, water depth

% Output: Wave pattern

function zeta = forwardProblem(Ig,p_ext,ship_dom_x,ship_dom_y,epsilon,Vx...
    ,Vy,dUx0,dUy0,kx,ky,k,h)
g = 9.81;
[ny, nx] = size(k);
p_ext_big = zeros(ny,nx);                               % Need to define the external

p_ext_big(ship_dom_y,ship_dom_x) = p_ext;

% Fourier space
P_ext = fft2(p_ext_big);


% Calculating the denominator Delta_r or Dr
Phi = 2*(kx*Vx + ky*Vy).*(1+Ig) + (kx*dUx0 + ky*dUy0).*tanh(k*h)./k;
Phi((ny/2)+1,(nx/2)+1) = 0;
Phi = sign(Phi);
D_r = (1+Ig).*(kx*Vx + ky*Vy).^2 + (kx*Vx + ky*Vy).*(kx*dUx0 + ky*dUy0)...
    .*tanh(k*h)./k - g*k.*tanh(k*h) + 1i*epsilon*Phi;

% Integrand
I = P_ext.*k.*tanh(k*h)./D_r;
I(1,1) = 0;

% Solution
zeta = ifft2(I);

end

% Plotting function
function plottingWaves(plottingParam,plottingData,PltOpt,hasG_ij)

set(0, 'DefaultTextInterpreter', 'latex')
set(0, 'DefaultAxesFontSize', 20)

switch PltOpt
    case 'Essentials'
    figure
    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,...
      real(plottingData.zeta), [-0.01,0.01])
    set(gca,’YDir’,’normal’)
    title(’Surface Elevation $\hat{\zeta}($\boldmath$\xi$)’)
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\xi_y$ \big[m\big]’)

    figure
    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,...
      real(plottingData.zeta))
    xlim([-20,5])
    ylim([-10,10])
    set(gca,’YDir’,’normal’)
    title(’Surface Elevation $\hat{\zeta}_s($\boldmath$\xi$)’)
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\xi_y$ \big[m\big]’)

    figure
    plot(plottingParam.xi_x_v,real(plottingData.zeta(...
      length(plottingParam.xi_y_v)/2,:)))
    xlim([plottingParam.xi_x_v(1),plottingParam.xi_x_v(end)])
    % xlim([xi2(length(xi2)*3/4),xi2(end)])
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\hat{\zeta}$ \big[m\big]’)
    title(’Surface Elevation Along Center of Domain’)

  case ’All’
%    figure
%    imagesc(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,real(plottingDa
%    set(gca,’YDir’,’normal’)
%    caxis manual
%    caxis([-30,30]);
%    colorbar
%    title(’External Pressure Distribution $\hat{p}_{ext}($\boldmath$\xi$)’)
%    xlabel(’$\xi_x$ \big[m\big]’)
%    ylabel(’$\xi_y$ \big[m\big]’)

    figure
    surf(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,...
      real(plottingData.p_ext))
    zlim([-30,30])
    caxis manual
    caxis([-30,30]);
    colorbar
    title(’External Pressure Distribution’)
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\xi_y$ \big[m\big]’)
    zlabel(’$\hat{p}_{ext}($\boldmath$\xi$) \big[m\textsuperscript{2}/s\textsuper

%    figure
%    imagesc(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,real(plottingDa
%    set(gca,’YDir’,’normal’)
%    caxis manual
%    caxis([-6,4]);
%    colorbar
%    title(’External Pressure Distribution’)
%    xlabel(’$\xi_x$ \big[m\big]’)
%    ylabel(’$\xi_y$ \big[m\big]’)

    figure
    surf(plottingParam.xi_ship_x_v(2:end),plottingParam.xi_ship_y_v,...
      real(plottingData.p_ext(:,2:end)))
    title(’External Pressure Distribution’)
    zlim([-6,4])
    caxis manual
    caxis([-6,4]);
    colorbar
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\xi_y$ \big[m\big]’)
    zlabel(’$\hat{p}_{ext}($\boldmath$\xi$) \big[m\textsuperscript{2}/s\textsuper

%    figure
%    imagesc(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,real(plottingDa
%    set(gca,’YDir’,’normal’)
%    caxis manual
%    caxis([-1,1]);
%    colorbar
%    title(’External Pressure Distribution’)
%    xlabel(’$\xi_x$ \big[m\big]’)
%    ylabel(’$\xi_y$ \big[m\big]’)

    figure
    surf(plottingParam.xi_ship_x_v(2:end),plottingParam.xi_ship_y_v,...
      real(plottingData.p_ext(:,2:end)))
    title(’External Pressure Distribution’)
    zlim([-1,1])
    caxis manual
    caxis([-1,1]);
    colorbar
    xlabel(’$\xi_x$ \big[m\big]’)
    ylabel(’$\xi_y$ \big[m\big]’)
```

```
zlabel(’$\hat{p}_{ext}($\boldmath$\xi$) \big[m\textsuperscript{2}/s\textsuper

figure
imagesc(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,...
    real(plottingData.zeta_s))
set(gca,’YDir’,’normal’)
caxis manual
caxis([-0.1,0.05]);
colorbar
title(’Prescribed Surface Elevation $\hat{\zeta}_s($\boldmath$\xi$’)
xlabel(’$\xi_x$ \big[m\big]’)
ylabel(’$\xi_y$ \big[m\big]’)

figure
surf(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,...
    real(plottingData.zeta_s))
zlim([-0.1,0.05])
caxis manual
caxis([-0.1,0.05]);
colorbar
title(’Prescribed Surface Elevation’)
xlabel(’$\xi_x$ \big[m\big]’)
ylabel(’$\xi_y$ \big[m\big]’)
zlabel(’$\hat{\zeta}_s($\boldmath$\xi$) \big[m\big]’)

figure
imagesc(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,...
    real(plottingData.wavePattern_ship))
set(gca,’YDir’,’normal’)
caxis manual
caxis([-0.1,0.05]);
colorbar
title(’Ship results’)
xlabel(’$\xi_x$ \big[m\big]’)
ylabel(’$\xi_y$ \big[m\big]’)

figure
surf(plottingParam.xi_ship_x_v,plottingParam.xi_ship_y_v,...
    real(plottingData.wavePattern_ship))
zlim([-0.1,0.05])
caxis manual
caxis([-0.1,0.05]);
colorbar
title(’Ship results’)
xlabel(’$\xi_x$ \big[m\big]’)
ylabel(’$\xi_y$ \big[m\big]’)
```

```
    zlabel('$\hat{\zeta}($\boldmath$\xi$) \big[m\big]')

%    figure
%    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,real(plottingData.zeta),
%    set(gca,'YDir','normal')
%    colorbar
%    title('Wave Pattern $\hat{\zeta}($\boldmath$\xi$)')
%    xlabel('$\xi_x$ \big[m\big]')
%    ylabel('$\xi_y$ \big[m\big]')

%    figure
%    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,real(plottingData.zeta))
%    set(gca,'YDir','normal')
%    caxis manual
%    caxis([-0.05,0.05]);
%    colorbar
%    title('Wave Pattern $\hat{\zeta}($\boldmath$\xi$)')
%    xlabel('$\xi_x$ \big[m\big]')
%    ylabel('$\xi_y$ \big[m\big]')

    figure
    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,...
        real(plottingData.zeta))
    set(gca,'YDir','normal')
    xlim([-20,5])
    ylim([-10,10])
    caxis manual
    caxis([-0.1,0.1]);
    colorbar
    title('Wave Pattern $\hat{\zeta}($\boldmath$\xi$)')
    xlabel('$\xi_x$ \big[m\big]')
    ylabel('$\xi_y$ \big[m\big]')

    figure
    imagesc(plottingParam.xi_x_v,plottingParam.xi_y_v,...
        real(plottingData.zeta))
    set(gca,'YDir','normal')
    xlim([-20,5])
    ylim([-10,10])
    caxis manual
    caxis([-0.05,0.05]);
    colorbar
    title('Wave Pattern $\hat{\zeta}($\boldmath$\xi$)')
    xlabel('$\xi_x$ \big[m\big]')
    ylabel('$\xi_y$ \big[m\big]')
```

```
figure
plot(plottingParam.xi_ship_x_v,real(plottingData.p_ext(...
    length(plottingParam.xi_ship_y_v)/2,:)))
xlabel('$\xi$ \big[m\big]')
ylabel('$\hat{p}_{ext}(\xi)$ \big[m\textsuperscript{2}/s\textsuperscript{2}]'
title('External Pressure Distribution')

figure
plot(plottingParam.xi_x_v,real(plottingData.zeta(...
    length(plottingParam.xi_y_v)/2,:)))
xlim([plottingParam.xi_x_v(1),plottingParam.xi_x_v(end)])
% xlim([xi2(length(xi2)*3/4),xi2(end)])
xlabel('$\xi_x$ \big[m\big]')
ylabel('$\hat{\zeta}$ \big[m\big]')
title('Surface Elevation Along Center of Domain')

figure
plot(plottingParam.xi_x_v,real(plottingData.zeta(...
    length(plottingParam.xi_y_v)/2,:)))
xlim([-plottingParam.l,plottingParam.l])
xlabel('$\xi_x$ \big[m\big]')
ylabel('$\hat{\zeta}$ \big[m\big]')
title('Ship Results Along Center')

if hasG_ij
    figure
    imagesc(plottingParam.xi_green_x_v,plottingParam.xi_green_y_v,...
        real(plottingData.G_ij))
    set(gca,'YDir','normal')
    caxis manual
    caxis([-150,150]);
    colorbar
    title('Greens function $G($\boldmath$\xi$)')
    xlabel('$\xi_x$ \big[m\big]')
    ylabel('$\xi_y$ \big[m\big]')

    figure
    surf(plottingParam.xi_green_x_v,plottingParam.xi_green_y_v,...
        real(plottingData.G_ij))
    zlim([-150,150]);
    caxis manual
    caxis([-150,150]);
    colorbar
    title('Greens function')
    xlabel('$\xi_x$ \big[m\big]')
    ylabel('$\xi_y$ \big[m\big]')
```

```
            zlabel('$G($\boldmath$\xi$) \big[s\textsuperscript{2}/m\textsuperscript{3
        end
    end

    end
```

References

[1] S. Å. Ellingsen and I. Brevik, "How linear surface waves are affected by a current with constant vorticity," *European Journal of Physics*, vol. 35, no. 025005, 2014.

[2] S. Å. Ellingsen, "Ship waves in the presence of uniform vorticity," *Journal of Fluid Mechanics*, vol. 742, no. R2, 2014.

[3] Y. Li and S. Å. Ellingsen, "Ship waves on uniform shear current at finite depth: wave resistance and critical velocity," *Journal of Fluid Mechanics*, vol. 791, pp. 539–567, 2016.

[4] Y. Li, B. K. Smeltzer, and S. Å. Ellingsen, "Transient wave resistance upon a real shear current," *European Journal of Mechanics-B/Fluids*, 2017, (in press, DOI:10.1016/j.euromechu.2017.08.012).

[5] O. M. Faltinsen, *Hydrodynamics of high-speed marine vehicles.* Cambridge university press, 2005.

[6] M. Benzaquen, A. Darmon, and E. Raphaël, "Wake pattern and wave resistance for anisotropic moving disturbances," *Physics of Fluids*, vol. 26, no. 092106, 2014.

[7] F. Noblesse, J. He, Y. Zhu, L. Hong, C. Zhang, R. Zhu, and C. Yang, "Why can ship wakes appear narrower than kelvin's angle?" *European Journal of Mechanics-B/Fluids*, vol. 46, pp. 164–171, 2014.

[8] M. Rabaud and F. Moisy, "Ship wakes: Kelvin or mach angle?" *Physical Review Letters*, vol. 110, no. 21, 2013.

[9] A. Darmon, M. Benzaquen, and E. Raphaël, "Kelvin wake pattern at large froude numbers," *Journal of Fluid Mechanics*, vol. 738, 2014.

[10] T. H. Havelock, "Some cases of wave motion due to a submerged obstacle," *Proceedings of the Royal Society of London. Series A*, vol. 93, no. 654, pp. 520–532, 1917.

[11] Y. Li and S. Å. Ellingsen, "Initial value problems for water waves in the presence of a shear current," in *The Twenty-fifth International Ocean and Polar Engineering Conference.* International Society of Offshore and Polar Engineers, 2015.

[12] Y. Li and S. Ellingsen, "Direct integration method for surface waves on depth dependent flows," in *EGU General Assembly Conference Abstracts*, 2018.

[13] B. J. Binder, M. G. Blyth, and S. W. McCue, "Free-surface flow past arbitrary topography and an inverse approach for wave-free solutions," *IMA Journal of Applied Mathematics*, vol. 78, no. 4, pp. 685–696, 2013.

[14] Y. Li, "Surface water waves on depth dependent flows," Ph.D. dissertation, Norwegian University of Science and Technology, 2017.

[15] P. H. LeBlond and L. A. Mysak, *Waves in the Ocean.* Elsevier, 1981, vol. 20.

[16] J. Lighthill, *Waves in fluids.* Cambridge University Press, 1978.

[17] L. Landau and E. Lifshitz, *Fluid mechanics*, 2nd ed. Elsevier, 1987.

[18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C. The Art of scientific Computing, 2nd Edition.* Cambridge University Press, 1992.

NTNU

Kunnskap for en bedre verden