g-RIPS SENDAI 2023

MITSUBISHI-B

Final Paper

Authors:
Aya Akahori[1]
Emma Thomas[2]
Kaede Watanabe[3]
Cameron Watt[4]

Mentors:
Dr. Toshiaki Yachimura[+]
Ryuhei Takahashi[*]
Dr. Hiroshi Kameda[*]

[1] Okayama University
[2] University of Wisconsin-Madison
[3] Tohoku University
[4] University of Pittsburgh
[+] Academic Mentor, Tohoku University
[*] Industrial Mentor, Mitsubishi

September 26, 2023

Contents

1 Physical background **2**
1.1 Relationship between Doppler spectrum and ocean wave spectra 2
1.2 Radar Range 3

2 Mathematical Background **4**
2.1 Tikhonov Regularization 4
2.2 Gradient Descent 4
2.3 Dynamic Step Size 5
2.4 Riesz Representation 8

3 Methods **9**
3.1 Defining the Cost Function for the Tikhonov Regularization Problem 9
3.2 Calculating the Gradient of the Cost Function 9

4 Applying the Gradient Descent Algorithm **11**
4.1 Numerical Technique 11
4.2 Tuning The Algorithm 12
4.3 Numerical simulation 14

5 Further Research **16**
5.1 Improving the Gradient Descent Algorithm with Backtracking 16
5.2 Testing Different Wave Spectra 17
5.3 Environmental Effects 17

1 Physical background

Ocean HF radar works by emitting radio waves of different frequencies. The radar then receives radio waves that are reflected back by the ocean's surface. The signal of these reflected waves is then changed to a frequency domain equation using a Fourier transform so that the Doppler spectrum of the reflected radio waves can be analyzed. The signals that the radar receives back from the ocean are determined by Bragg scattering. Bragg resonance occurs when radio waves that are reflected by consecutive ocean wave peaks are in phase with one another and have maximum constructive interference. This requires the distance between the ocean wave peaks to be half of the radio wavelength. These reflected radio waves are said to be Bragg scattered.

First-order Bragg scattering occurs when an ocean wave moving directly towards or away from the radar has a wavelength that is half the radio wavelength. We can represent both ocean waves and radio waves using wavevectors. The magnitude of each wavevector is the wavenumber of the wave, and the direction is the direction of the physical wave. If $\mathbf{k_0}$ is a wavevector representing the radio wave and $\mathbf{k_d}$ is an ocean wave moving directly towards or away from the radar, then first order Bragg scattering requires that

$$\mathbf{k_d} = \pm 2\mathbf{k_0}. \tag{1.1}$$

Second-order Bragg scattering involves multiple ocean waves moving in different directions. It arises from both electromagnetic and hydrodynamic effects. In the case of electrodynamics effects, the second-order Bragg scattering is a result of a radio wave bouncing off first one ocean wave and then another in such a way that Bragg resonance occurs. Let $\mathbf{k_1}$ represent the first wave and $\mathbf{k_2}$ the second wave. In this situation, Bragg resonance occurs when

$$\mathbf{k_1} + \mathbf{k_2} = -2\mathbf{k_0}. \tag{1.2}$$

In the case of hydrodynamic effects, the second-order Bragg scattering occurs when the wave reflects off the intersection of two ocean waves. As long as the two intersecting ocean waves fulfill (1.2), Bragg resonance will occur and result in second-order Bragg scattering.

1.1 Relationship between Doppler spectrum and ocean wave spectra

Let $\sigma(\omega)$ be the Doppler spectrum of Bragg scattering, where ω is the Doppler frequency in radians per second ($\omega = 2\pi f$). If $\sigma_{(1)}$ is the Doppler spectrum of first-order Bragg scattering and $\sigma_{(2)}$ is the Doppler spectrum of second-order Bragg scattering, then

$$\sigma(\omega) = \sigma_{(1)} + \sigma_{(2)} \tag{1.3}$$

There exist theoretical equations describing the relationship between the ocean waves' spectra and the Doppler spectra of first and second-order Bragg scattering. Our research focuses on retrieving the wave spectrum, $S(\mathbf{k})$ of an ocean wave, $\mathbf{k}$, from the theoretical equation relating S($\mathbf{k}$) to $\sigma_{(2)}$ given below.

$$\sigma_{(2)}(\omega) = 2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2| S(m_1\mathbf{k}_1)S(m_2\mathbf{k}_2) \times\delta\left(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2}\right)\, dp\, dq. \tag{1.4}$$

Here, Γ is the coupling coefficient described below, $k_0 = |\mathbf{k_0}| = \frac{2\pi}{\lambda}$ where λ is the distance between ocean wave peaks, and delta is the Dirac delta function. The wave vectors $\mathbf{k}_1$ and $\mathbf{k}_2$ are parameterized by p and q as follows:

$$\mathbf{k_0} = (k_0, 0), \mathbf{k_1} = (p - k_0, q), \mathbf{k_2} = (-p - k_0, -q). \tag{1.5}$$

The coupling coefficient, $\Gamma(m_1\mathbf{k_1}, m_2\mathbf{k_2})$, is calculated by

$$\Gamma(m_1\mathbf{k_1}, m_2\mathbf{k_2}) = \Gamma_E(\mathbf{k_1}, \mathbf{k_2}) + \Gamma_H(m_1\mathbf{k_1}, m_2\mathbf{k_2}) \tag{1.6}$$

where

$$\Gamma_E(\mathbf{k_1}, \mathbf{k_2}) = \frac{1}{2}\frac{(\mathbf{k_1k_0})(\mathbf{k_2k_0})/k_0^2 - 2(\mathbf{k_1k_2})}{(\sqrt{\mathbf{k_1k_2}} - k_0\Delta))} \tag{1.7}$$

$$\Gamma_H(m_1\mathbf{k_1}, m_2\mathbf{k_2}) = -\frac{i}{2}\left(k_1 + k_2 - \frac{(k_1k_2 - \mathbf{k_1k_2})(\omega^2 + \omega_B^2)}{m_1m_2\sqrt{\mathbf{k_1k_2}}(\omega^2 - \omega_B^2)}\right) \tag{1.8}$$

with Δ being the average normalized impedance of the seawater.

1.2 Radar Range

The area of the sea's surface that the radar reflects off of is broken up into a grid. The signals received back by the radar are measured for each grid section, where the sections are separated by direction and range, as shown in Figure 1. The width of the direction and range separations is determined by the radar's azimuth and range resolutions, which in this project are approximately 15 deg and 1500 m. The total range is about 120 deg, and the total azimuth is about 50 km.

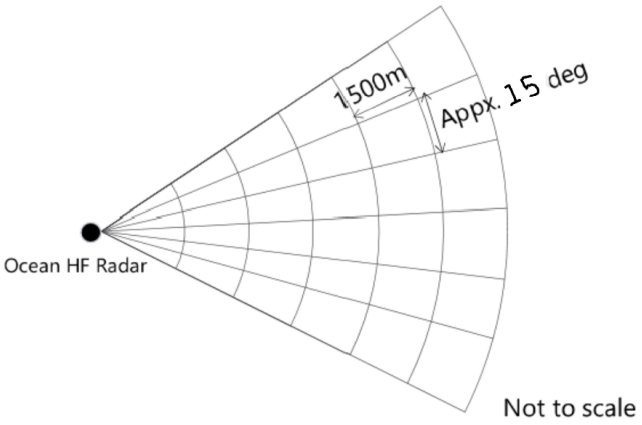

Figure 1: Radar Range Direction

2 Mathematical Background

2.1 Tikhonov Regularization

Tikhonov regularization is a technique used to solve ill-posed inverse problems of the form

$$F(x) = y_0 \tag{2.1}$$

where $F : X \to Y$ is a continuous operator between Hilbert spaces X and Y. The strategy is to consider

$$L(x) = \|F(x) - y_0\| \tag{2.2}$$

and find the x which minimizes this function. For ill-conditioned F, $x_0 = \arg\min_x L(x)$, may not accurately approximate the true solution [10]. In such cases, we introduce the regularization term $\lambda\|x\|$ for some regularization parameter $\lambda > 0$

$$L(x) = \|F(x) - y_0\| + \lambda\|x\|. \tag{2.3}$$

The regularization term controls the fitting of the solution to data. Small values of λ provide a weak constraint to the error function and, therefore, may result in an overfitted solution. An overfitted solution matches the data points very accurately, but may not provide a good approximation of the true solution. Larger values of λ will overregularize the solution resulting in underfitting. For finite-dimensional linear operators A, we can choose a regularization parameter to satisfy

$$\|Ax_\lambda - y_0\| = \tau\|e\| \tag{2.4}$$

where

$$x_\lambda = \|(A^*A + \lambda^2 I)^{-1}\|A^*y_0 \tag{2.5}$$

τ is a tolerance and e is the error due to noise [10]. Therefore, if we have a good estimate for the expected amount of noise $\|e\|$, then it is relatively easy to find a good regularization parameter to use. However, for infinite dimensional non-linear problems, choosing the right parameter can be a much more difficult task. As such, we will need to rely on experimental methods to determine what regularization parameter to use.

2.2 Gradient Descent

Gradient descent is an iterative method used for optimizing differentiable functions. Gradient descent algorithms can be used to locate the minima of finite dimensional cost functions of the form $f : \mathbb{R}^n \to \mathbb{R}$. The first step in the gradient descent algorithm is to pick an initial guess, x_0. Next, calculate the gradient of f at x_0. The vector $-\nabla f(x_0)$ then gives us the direction of the steepest descent of our cost function. After calculating the gradient, we set

$$x_1 = x_0 - \alpha\nabla f(x_0) \tag{2.6}$$

for a chosen step size α. For sufficiently small α, we have $f(x_1) < f(x_0)$. We then iterate this process by setting

$$x_{n+1} = x_n - \alpha \nabla f(x_n). \tag{2.7}$$

For small enough values of α, we will have a monotone decreasing sequence. Then if we assume our cost function is bounded below, then the monotone converge theorem ensures the existence of a minimum $y^* = \min_{n\to\infty} f(x_n)$. However, in general, the existence of $\lim_{n\to\infty} x_n$ is not guaranteed.

In order for $\lim_{n\to\infty} x_n$ to converge to a minimizer, we need to have a starting guess that is already sufficiently close to a minimizer. This can be difficult if there is little information about the behavior of the cost function f. One way to ensure convergence of the gradient descent method to a global minimizer is to prove the strict convexity of the cost function. If a function is not convex, we could attempt to locate the global minimizer of the cost function by running the gradient descent algorithm for many different starting points. This may be useful for determining more general properties of the cost function, such as the locations of many different minimizers and what the basins of attraction are for different minima. However, running the gradient descent algorithm for many different starting points may take too long to compute.

2.3 Dynamic Step Size

When using the gradient descent method, dynamic step sizes can be useful for reducing the number of iterations and also for finding global minima. Small step sizes will give more accurate trajectories toward the minimizer of the cost function. However, if the step sizes are too small, then the algorithm will take too many iterations to converge. Larger step sizes may not converge to a minimizer even with a good initial guess because the gradient only provides local information about the cost function.

When using a dynamic step size, we modify equation 2.7 by

$$x_{n+1} = x_n - \alpha_n \nabla f(x_n). \tag{2.8}$$

Here we see that the step size a_n can now change as the descent algorithm runs. In order for the gradient descent method to converge quickly, we want to have large step sizes when $\|x_n - x^*\|$ is large and small step sizes when $\|x_n - x^*\|$ is small. For the first case, the larger step sizes cause the values of x_n to quickly move through the domain of f, which allows us to quickly find a point that is relatively close to a minimum. As we get closer to a minimum, we then use smaller steps to hone in on a precise value for the minimizer. One problem with using dynamic steps is determining the optimal step size. You cannot simply calculate the value of $\|x_n - x^*\|$ as described because that would require us to know what the minimizer is, to begin with. If you have some knowledge about what the minimum of the cost function is, as is the case for some least squares problems, then you can adjust the step size according to $\|f(x_n) - f(x^*)\|$. For many problems, though, we do not know the expected minimum. In this situation, we need a different method for determining the step size.

There are a few things to consider when attempting to determine what step size to use. First, we will note that a sufficiently large step size from x_n may result in $f(x_{n+1}) > f(x_n)$

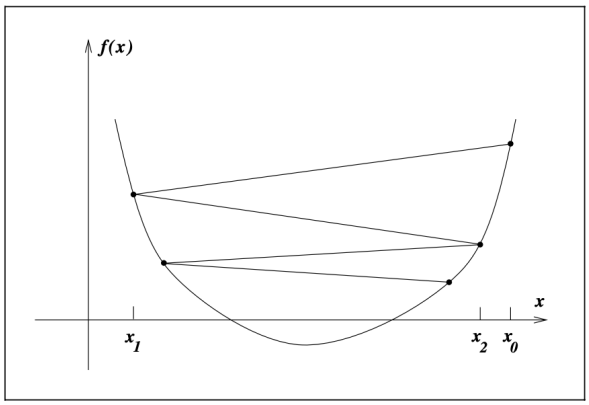

Figure 2: Gradient descent algorithm with insufficient decrease. Reprinted from [8].

This is a situation we definitely want to avoid since we cannot locate the minimum of f if the values of $f(x_n)$ are not monotonically decreasing. Therefore, the simplest restriction that we can introduce is

$$f(x_{n+1}) \leq f(x_n). \tag{2.9}$$

This condition, however, is not sufficient to ensure quick convergence, as can be seen in figure 2. We can remedy this by introducing another term that requires the next estimate to decrease by a certain amount

$$f(x_{n+1}) \leq f(x_n) - c_1\alpha_n\|\nabla f(x_n)\| \tag{2.10}$$

for some constant $c_1 \in (0, 1)$. This is referred to as the Armijo condition. The Armijo constant c_1 is a tuning parameter that indicates how strict of a decrease we are demanding for our function. The Armijo condition is depicted in figure 3. Values of c_1 closer to 0 are less strict. In figure 3, this corresponds to $l(\alpha)$ having a smaller slope which would increase the size of the acceptable regions. A value of $c_1 = 0$ would correspond to $l(\alpha)$ being a horizontal line and gives us the condition in equation 2.9. We exclude 0 from the possible values of the Armijo condition because we want to improve upon the condition in equation 2.9. Values of c_1 that are closer to 1 require a more strict decrease. A value of $c_1 = 1$ corresponds to $l(\alpha)$ being the tangent line. We exclude this value because there might not be any values of $f(x)$ in the direction of the gradient that is below the tangent line, as is the case in figure 3. For values of $c_1 \in (0, 1)$, we get a secant line $l(\alpha)$ with a slope greater than the tangent line. There are guaranteed to be values of $f(x)$ on this line in the direction of the gradient that lies below the line. One issue with choosing a strict Armijo constant is that the upper bound line $l(\alpha)$ may be so steep that the acceptable region only exists close to the initial guess. This is undesirable as it results in a very short step size giving little improvement and requiring more iterations. We, therefore, need to carefully choose our step size in order to accomplish a faster computation. As with the regularization parameter, we rely on experimental methods to determine what parameter to choose.

In addition to the Armijo condition, we can add more conditions to restrict the acceptable region for the step size. We would like for the next step in the iteration to be closer to a

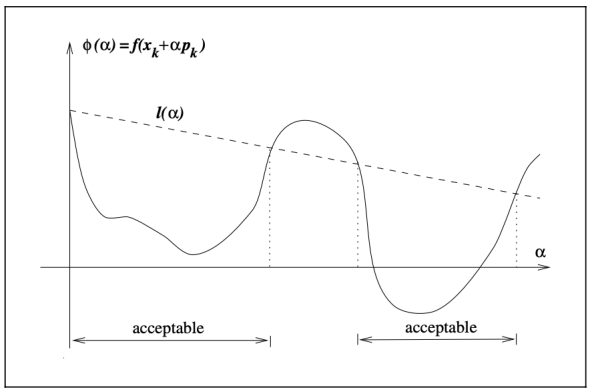

Figure 3: Example of Armijo Condition. Reprinted from [8].

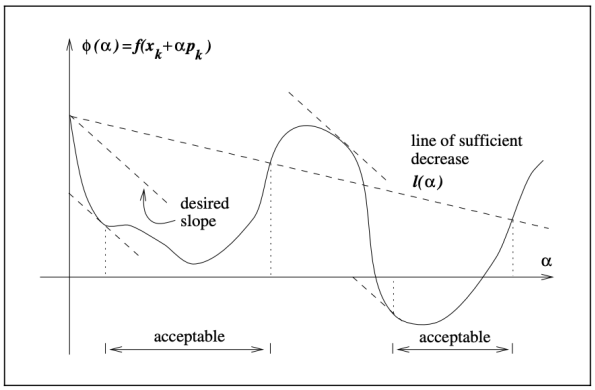

Figure 4: Example of Wolfe Condition. Reprinted from [8]

minimum than the current step. Without a good understanding of the cost function, it is difficult to know if you are getting closer or further from a minimum. However, we do know that any minimum will have a gradient of 0 and, therefore, a directional derivative of 0 in every direction. Thus, we would like the next iteration to have a less steep directional derivative than the current directional derivative in the direction of $\nabla f(x_n)$

$$|\langle \nabla f(x_{n+1}), \nabla f(x_n) \rangle| \leq |\langle \nabla f(x_n), \nabla f(x_n) \rangle|. \tag{2.11}$$

As with the Armijo rule, this restriction is useful, but we can introduce a parameter c_2 to demand further improvement. This gives us our second condition

$$|\langle \nabla f(x_{n+1}), \nabla f(x_n) \rangle| \leq c_2 |\langle \nabla f(x_n), \nabla f(x_n) \rangle| \tag{2.12}$$

for some constant $c_2 \in (c_1, 1)$. This is referred to as the strong Wolfe condition. The constant c_2 determines how shallow of a slope we demand. Values of c_2 that are close to 1 will more closely approximate condition 2.11 and are therefore less restrictive. Values closer to c_1 will cause the desired slope to be flatter, which provides a stricter improvement demand.

One important concern when adding these restrictions to the possible step size is whether there exist step sizes that can actually satisfy both these conditions simultaneously. We will

prove that such a step length will always exist. However, we will first add the assumption that $f(x)$ be bounded below [8].

Proof. First, we will try to find a range of values such that the Armijo condition is satisfied. Let $\phi(\alpha) = f(x_n - \alpha\nabla f(x_n))$ and $l(\alpha) = f(x_n) - c_1\alpha\langle\nabla f(x_n), \nabla f(x_n)\rangle$. First note that $\phi(0) = l(0)$. Furthermore, note that

$$-c_1\|f(x_n)\| > -\|f(x_n)\| \tag{2.13}$$

$$l'(0) > \phi'(\alpha). \tag{2.14}$$

In particular, there exists some $\alpha_1 > 0$ such that $l(\alpha_1) > \phi(\alpha_1)$. However, we note that $f(x)$ is bounded below, so $\phi(\alpha)$ is bounded below, but $l(\alpha)$ is not. Therefore, there exists some $\alpha_2 > \alpha_1$ such that $l(\alpha_2) < \phi(\alpha_2)$. Thus, by the mean value theorem, there exists an $\alpha' \in (\alpha_1, \alpha_2)$ such that $l(\alpha') = \phi(\alpha')$. Then we let $\alpha' = \min\{\alpha : l(\alpha) = \phi(\alpha)\}$. For $\alpha \in (0, \alpha')$, the Armijo condition is satisfied.

Then for the Wolfe condition, we use the mean value theorem to show that there exists an $\alpha'' \in (0, \alpha')$ such that

$$\begin{aligned}
\phi(\alpha') - \phi(0) &= \alpha'\phi'(\alpha'') \\
l(\alpha') - f(x_n) &= -\alpha'\langle\nabla f(x_n - \alpha''\nabla f(x_n)), \nabla f(x_n)\rangle \\
-c_1\alpha'\langle\nabla f(x_n), \nabla f(x_n)\rangle &= -\alpha'\langle\nabla f(x_n - \alpha''\nabla f(x_n)), \nabla f(x_n)\rangle \\
c_1|\langle\nabla f(x_n), \nabla f(x_n)\rangle| &= |\langle\nabla f(x_n - \alpha''\nabla f(x_n)), \nabla f(x_n)\rangle|.
\end{aligned} \tag{2.15}$$

Furthermore, we note that $c_2 > c_1$, so

$$c_2|\langle\nabla f(x_n), \nabla f(x_n)\rangle| > |\langle\nabla f(x_n - \alpha''\nabla f(x_n)), \nabla f(x_n)\rangle|. \tag{2.16}$$

Therefore, the strong Wolfe condition is satisfied. □

2.4 Riesz Representation

Until this point, we have only discussed the mathematical theory for a finite-dimensional setting. However, our problem requires us to apply the gradient descent method to an infinite dimensional cost function. Thus we need to define what the gradient is for a cost function from an infinite dimensional domain. First, we will introduce the concept of the Fréchet derivative. The Fréchet derivative generalizes the concept of a derivative to functions between normed vector spaces. Consider a function $f : V \to W$ between normed vector spaces. We say f is Fréchet differentiable if there exists a bounded linear operator $Df : V \to W$ such that

$$\lim_{\|h\|_V \to 0} \frac{\|f(x+h) - f(x) - Df(h)\|_W}{\|h\|_V} = 0. \tag{2.17}$$

We call Df, the Fréchet derivative of f. Now we introduce the Riesz representation theorem.

Riesz Representation Theorem 1. Given a Hilbert space, X, and T $\in L(X, \mathbb{R})$, $\exists! v_T \in X$ such that for all $h \in X$

$$T(v) = \langle v_T, h\rangle \tag{2.18}$$

and

$$||v_T||_X = ||T||_{L(X,\mathbb{R})}. \tag{2.19}$$

Applying the Reisz representation theorem to the Fréchet derivative of f at x gives us a $v_{Df}(x)$ such that

$$Df(x)[h] = \langle v_{Df}(x), h \rangle. \tag{2.20}$$

This parallels the equation for a finite-dimensional directional derivative given by

$$D_u f(x) = \langle \nabla f(x), u \rangle. \tag{2.21}$$

Therefore, for a differentiable function f between normed vector spaces, we define the gradient of f to be the Reisz representation of the derivative

$$\nabla f(x) = v_{Df}(x). \tag{2.22}$$

Using this definition allows us to apply the gradient descent algorithm to any function defined on a normed vector space. In addition, we can use this definition to apply the Armijo and Wolfe conditions to our problem.

3 Methods

3.1 Defining the Cost Function for the Tikhonov Regularization Problem

The problem of determining the wave spectrum $\mathrm{S}(\mathbf{k})$ from a known Doppler spectrum $\sigma_{(2)}(\omega)$ can be expressed as a Tikhonov regularization problem. First, we define the non-linear operator $A : L(\mathbb{R}^2, \mathbb{R}) \to L(\mathbb{R}, \mathbb{R})$ as

$$A(S) = 2^6 \pi k_0^4 \sum_{m_1=\pm 1} \sum_{m_2=\pm 1} \iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2 | S(m_1\mathbf{k}_1) S(m_2\mathbf{k}_2) \times \delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\, dp\, dq. \tag{3.1}$$

The right-hand side is identical to the right-hand side of equation 1.4, but $A(S)$ is a function of the wave spectrum rather than a function of the Doppler frequency. From this operator, we define a cost function, $L : L(\mathbb{R}^2, \mathbb{R}) \to \mathbb{R}^2$

$$L(S) = ||A(S) - \sigma_{(2)}(\omega)||^2_{L^2(\mathbb{R})} + \lambda ||S||^2_{L^2(\mathbb{R}^2)} \tag{3.2}$$

where λ is our regularization parameter. Our goal is to minimize this cost function using a gradient descent algorithm to determine the wave spectrum.

3.2 Calculating the Gradient of the Cost Function

We need to determine an expression for the gradient of the cost function, but first, we will show that $A(S)$ is Fréchet differentiable.

Proof.

$$\begin{aligned}
A(S+h) =& 2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2|[S(m_1\mathbf{k}_1)+h(m_1\mathbf{k}_1)] \\
&\times [S(m_2\mathbf{k}_2)+h(m_2\mathbf{k}_2)]\delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\,dp\,dq \qquad (3.3)\\
=& 2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2|\Big[S(m_1\mathbf{k}_1)S(m_2\mathbf{k}_2) + S(m_1\mathbf{k}_1)h(m_2\mathbf{k}_2) \\
&\times S(m_2\mathbf{k}_2)h(m_1\mathbf{k}_1) + h(m_1\mathbf{k}_1)h(m_2\mathbf{k}_2)]\delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\,dp\,dq \qquad (3.4)\\
=& A(S) + 2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2|\Big[S(m_1\mathbf{k}_1)h(m_2\mathbf{k}_2) \\
&+ S(m_2\mathbf{k}_2)h(m_1\mathbf{k}_1)\Big]\delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\,dp\,dq + o(h). \qquad (3.5)
\end{aligned}$$

Therefore, the Fréchet derivative of A is given by

$$\begin{aligned}
A'(S)[h] =& 2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2|\Big[S(m_1\mathbf{k}_1)h(m_2\mathbf{k}_2) \\
&+ S(m_2\mathbf{k}_2)h(m_1\mathbf{k}_1)\Big]\delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\,dp\,dq. \qquad (3.6)
\end{aligned}$$

□

We can then use this to express the Fréchet derivative of our cost function

$$L'(S)[h] = 2\langle A(S) - \sigma_2(\omega), A'(S)[h]\rangle_{L^2(\mathbb{R})} + 2\lambda\langle S, h\rangle_{L^2(\mathbb{R}^2)}. \qquad (3.7)$$

Now we need to determine the Riesz representation of the derivative of the cost function. The Riesz representation of $L'(S)$ is a function $v_S \in L^2(\mathbb{R}^2)$ such that

$$L'(S)[h] = \iint_{\mathbb{R}^2} v_S(p,q)h(p,q)dpdq. \qquad (3.8)$$

One way for us to calculate v_S is to rewrite the integral for $L'(S)[h]$ in a way that allows us to factor out the h. In order to do this, we need to make sure that each instance of h in equation 3.6 has the same argument. This can be accomplished by splitting up $A'(S)$ into 8 terms, each of which takes the form

$$2^6\pi k_0^4 \sum_{m_1=\pm1}\sum_{m_2=\pm1}\iint_{\mathbb{R}^2} |\Gamma(m_1\mathbf{k}_1, m_2\mathbf{k}_2)^2|\Big[S(m_i\mathbf{k}_i)h(m_j\mathbf{k}_j)\Big]\delta(\omega - m_1\sqrt{gk_1} - m_2\sqrt{gk_2})\,dp\,dq, \qquad (3.9)$$

$i, j \in \{1, 2\}$, $j \neq i$. In order to factor out h from each term, we want to perform substitutions so that the arguments of h in each term are identical. For each term, we make the substitution $(m_jk_j) = (x, y)$. When j=1, this substitution gives us $x = m_1(-p - k_0)$ and $y = -m_1q$, then solve for p and q to get $p = \frac{x}{m_1} + k_0$ and $q = -\frac{y}{m_2}$. When j=2, this substitution gives us

$x = -m_2(p - k_0)$ and $y = -m_1 q$, then solve for p and q to get $p = -\frac{x}{m_2} - k_0$ and $q = -\frac{y}{m_2}$. We then substitute these values as the p and q parameters of k_j. As an example, we will look at the term where $i = 1$, $j = 2$, and $m_i = m_j = 1$. In this case, we make the substitution $x = -p - k_0$ and $y = -q$, then solve for p and q to get $p = -x - k_0$ and $q = -y$. Substituting these values in as the parameters for $\mathbf{k_1}$ gives us:

$$2^6 \pi k_0^4 \sum_{m_1=\pm 1} \sum_{m_2=\pm 1} \iint_{\mathbb{R}^2} \left| \Gamma\Big((-x - 2k_0, -y), (x, y) \Big) \right|^2 \Big[S((-x - 2k_0, -y) h(x, y) \Big] \tag{3.10}$$

$$\times \, \delta \left(\omega - \sqrt{g}[(x + 2k_0)^2 + y^2]^{\frac{1}{4}} - \sqrt{g}[x^2 + y^2]^{\frac{1}{4}} \right) dx dy. \tag{3.11}$$

The remaining terms can be calculated with similar substitutions. Grouping results, factoring out the $h(x, y)$, and adding the normalization term gives us

$$v_S(x, y) = \int_{-\infty}^{\infty} 4(A(S) - \sigma_{(2)}(\omega)) \sum_{m_1=\pm 1} \sum_{m_2=\pm 2} \left| \Gamma \left(-\frac{m_1}{m_2}(x + 2m_2 k_0), y), (x, y) \right) \right|^2$$

$$\times \delta \left(\omega - m_1 \sqrt{g} \left[\left(\frac{1}{m_2} x + 2k_0 \right)^2 + y^2 \right]^{\frac{1}{4}} - m_2 \sqrt{g}[x^2 + y^2]^{\frac{1}{4}} \right) S \left(-\frac{m_1}{m_2}(x + 2m_2 k_0, y) \right) d\omega$$

$$+ 2\lambda S. \tag{3.12}$$

We can now define

$$\nabla L(S) = v_S(x, y). \tag{3.13}$$

Thus, we can use this to define our gradient descent recursion as

$$S_{n+1} = S_n - \alpha_n v_{S_n}. \tag{3.14}$$

The expression for the gradient of L also allows us to rewrite the Armijo condition as

$$L(S_{n+1}) \leq L(S_n) - c_1 \alpha_n \| v_{S_n} \|_{L^2(\mathbb{R}^2)}. \tag{3.15}$$

Similarly, we can rewrite the Wolfe condition as

$$|\langle v_{S_{n+1}}, v_{S_n} \rangle_{L^2(\mathbb{R}^2}| \leq c_2 |\langle v_S, v_S \rangle|_{L^2(\mathbb{R}^2)}. \tag{3.16}$$

4 Applying the Gradient Descent Algorithm

4.1 Numerical Technique

We suppose that we are given a second-order Doppler spectrum $\sigma_{(2)}(\omega)$ with the aim of finding the wave spectra $S(\mathbf{k})$ that produced it. First, we interpolate the data of $\sigma_{(2)}(\omega)$ to get a function to $\mathbb{R}$. The radar provides data for $\omega \in [-4.6116\pi, 4.6116\pi]$. We then choose an initial wave spectrum S_0 to begin the gradient descent algorithm. A partition of the pq-plane is chosen to numerically calculate the integral of $A(S_0)$. The current version of

our code breaks the relevant segment of the pq-plane into a 30×30 grid. We note that the double integral in equation 3.1 includes a delta function. This delta function defines a curve of integration, but the argument of the delta function is too complicated to find an explicit parameterization. Therefore, we approximate the delta function as

$$\delta(x) \approx \begin{cases} \frac{1}{2\epsilon} & \text{if } |x| < \epsilon \\ 0 & \text{otherwise} \end{cases}. \tag{4.1}$$

Once an approximation for $A(S_0)$ is calculated, the Reisz representation is numerically calculated using the approximation for $\delta(x)$ again. Lastly, the original function S_0 is adjusted according to equation 3.14.

4.2 Tuning The Algorithm

There are a couple of challenges that we need to address when using the Gradient descent algorithm. The first is that the behavior of $L(S)$ is difficult to predict. Therefore, we are unable to determine if the solution we get from the gradient descent algorithm is the actual solution or if there is another wave spectrum that produces the same minimum. One way to fix this would be to figure out how to determine a good starting guess for the gradient descent algorithm. However, this requires more information about the characteristics that could affect the wave spectra.

Additionally, we know that $L(S)$ is bounded below 0, but we have no information about how small the minimum value of $L(S)$ is expected to be. Therefore, it is unknown if we are close to a minimum or not at any step in the iteration. This makes it difficult to determine when to stop the algorithm. One way to determine a stopping point would be to check for the rate of improvement and stop once the improvement slows to a chosen amount.

Our main method for determining what parameters to use for the gradient descent algorithm will be to train the algorithm on Doppler spectra created from known wave spectra. When constructing the wave Doppler spectra, we want to use functions that closely model real wave spectra. We use the model from Kataoka [6]. First, we describe the wave spectra as a function of frequency and direction:

$$S(\mathbf{k}) = S_f(f)G(\theta|f) \tag{4.2}$$

where S_f is the frequency spectra and G is the wave spectra distribution. The frequency spectra are given by

$$S_f(f) = 0.257 H_s^2 T_s (T_s f)^{-5} \exp\left[-1.03(T_s f)^{-4}\right] \tag{4.3}$$

where H_s is the height of the significant wave and T_s is the period of the significant wave. The wave spectra distribution is given by

$$G(\theta|f) = G_0 \cos^{2\tilde{s}(f)}\left(\frac{\theta - \theta_0}{2}\right) \tag{4.4}$$

where G_0 is a normalization constant and $\tilde{s}$ is the wave index. The wave index is given by

$$\tilde{s}(f) = \begin{cases} s_{\max}\left(\frac{f}{f_p}\right)^5 & \text{if } f \le f_p \\ s_{\max}\left(\frac{f}{f_p}\right)^{-2.5} & \text{if } f > f_p \end{cases} \tag{4.5}$$

where $s_{\max}$ is a parameter and f_p is the peak wave frequency. Since $G(\theta|f)$ is a distribution, we require that

$$\int_{-\pi}^{\pi} G(\theta|f)d\theta = 1 \tag{4.6}$$

Therefore,

$$G_0 = \left[\int_{-\pi}^{\pi} \cos^{2\tilde{s}(f)}\left(\frac{\theta-\theta_0}{2}\right) d\theta\right]^{-1}. \tag{4.7}$$

An example of a wave spectrum constructed from this model is shown in figure 5. The figure indicates a dominant wave traveling with a frequency of $f = \frac{1}{T_s} = 0.33$ in the direction of $\theta_0 = 0.5$. The mass of the wave spectra is concentrated near $\left(\frac{1}{T_s}, \theta_0\right)$ and continuously decreases to 0 at further points.

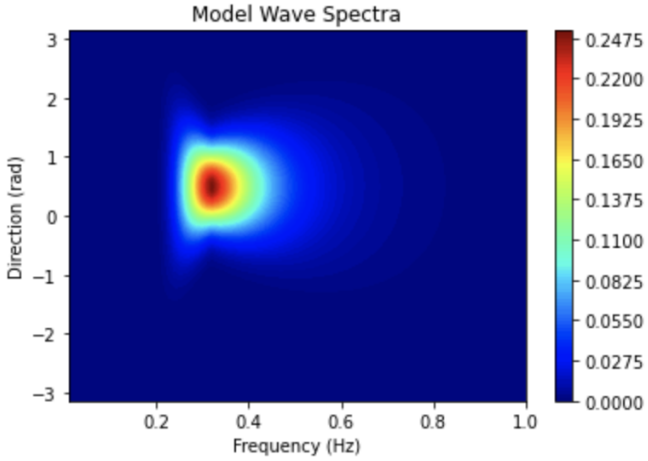

Figure 5: Model wave spectrum for $H_s = 1$, $T_s = 3$, $s_{\max} = 10$, $f_p = 0.317$, $\theta_0 = 0.5$ in the frequency-direction plane.

The wave spectrum is constructed in the $f\theta$-plane with $f \in [10^{-2}, 1]$ and angle values of $\theta \in [-\pi, \pi]$. The spectra is then converted to the pq-plane with the relations

$$\theta = \arctan\left(\frac{p}{q}\right) \tag{4.8}$$

$$f = \frac{1}{2\pi}\sqrt{g}(p^2+q^2)^{1/4}. \tag{4.9}$$

The model wave spectrum is then used to construct a second-order Doppler Wave spectrum as described in section 4.1. The second-order Doppler spectrum resulting from the wave spectrum in figure 5 is depicted in figure 7. For this computation, $\epsilon = 0.01$ was used for the delta function approximation. One important thing to note is that for a frequency $\omega = \sqrt{2gk_0}$, there is only first-order Bragg scattering, so we expect an absolute power of 0 at these values.

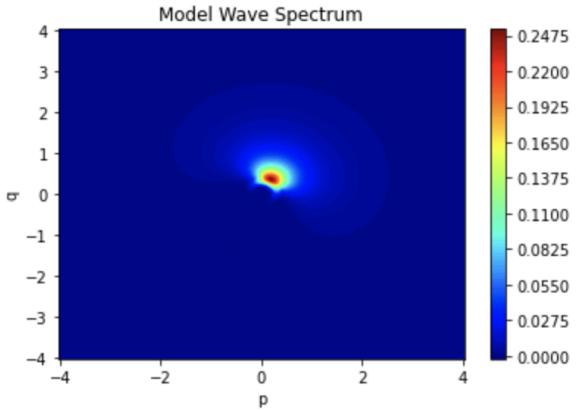

Figure 6: Model wave spectrum for $H_s = 1$, $T_s = 3$, $s_{\max} = 10$, $f_p = 0.317$, $\theta_0 = 0.5$ in the pq-plane.

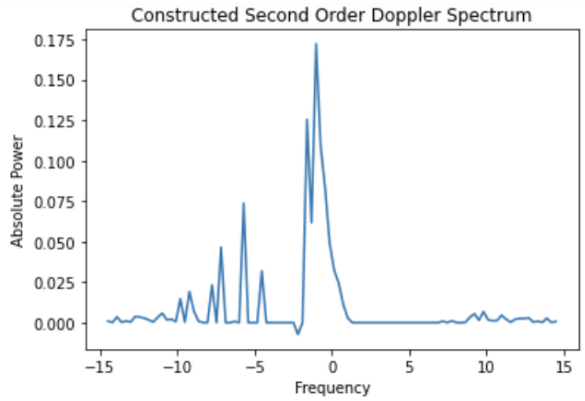

Figure 7: Second order Doppler spectrum constructed from the model wave spectrum in figure 5.

4.3 Numerical simulation

We solved the direct problem by using the wave spectrum with $H_s = 1$, $T_s = 3$, $s_{\max} = 10$, $f_p = 0.317$, $\theta_0 = 0$, and constructing σ_2. These are the contour lines of the sample wave spectrum (figure 8) and constructed σ_2 (figure 9).

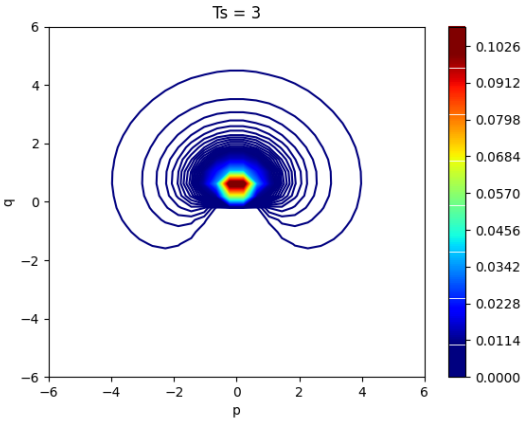

Figure 8: The model of wave spectrum with parameters $H_s = 1$, $T_s = 3$, $s_{\max} = 10$, $f_p = 0.317$, $\theta_0 = 0$.

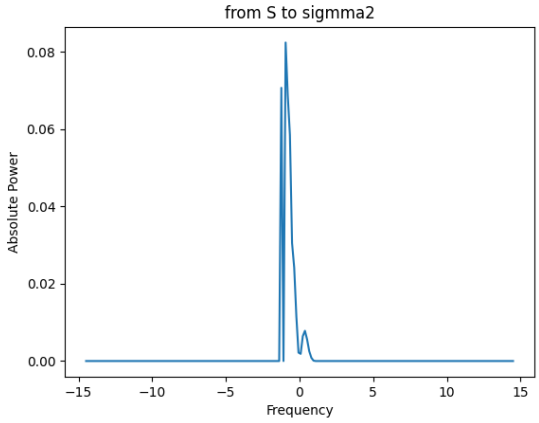

Figure 9: Second-order Doppler spectrum constructed from the model wave spectrum in the figure.

By using constructed σ_2, we tried to reconstruct the initial value. The algorithm is as below.

1. Set initial S_0, $n = 0$.
2. Using the Riesz representation formula, find the steepest gradient, $\nabla L(S)$.
3. From the formula, $S_{n+1} = S_n - \alpha \nabla L(S)$, generate next S value.

4. Iterate the process from 2 to 3 until convergence.

By setting the initial condition to get σ_2 as S_0, the result of the numerical simulation is in Figure 10. The parameters are step size = 1, iteration number = 30, regularization term = 10^{-4}, and calculation time is about 54 sec. This is the almost same as the initial condition.

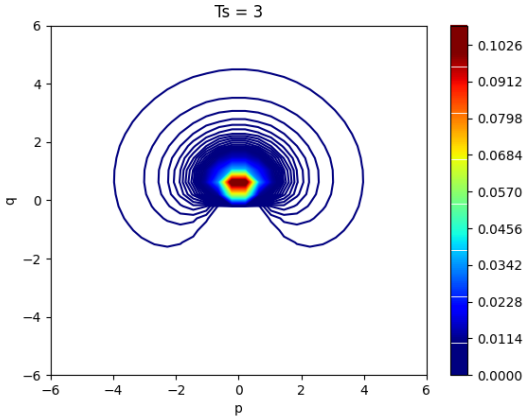

Figure 10: The same figure 8

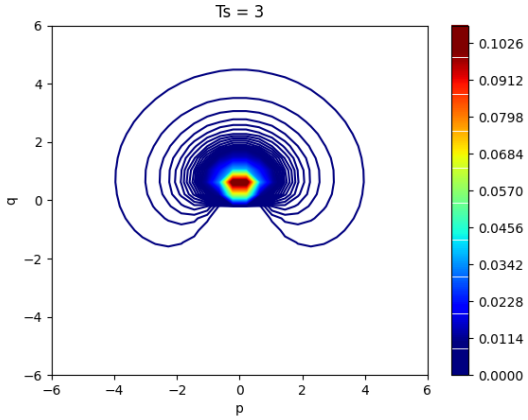

Figure 11: Reconstruction.

Next, we add perturbation to the initial data. Let ϵ be the amplitude of perturbation. When $\epsilon = 0.001$, the initial condition is figure 11.

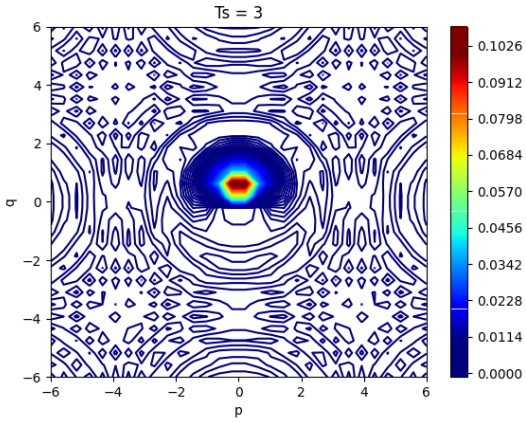

Figure 12: $\epsilon = 0.001$

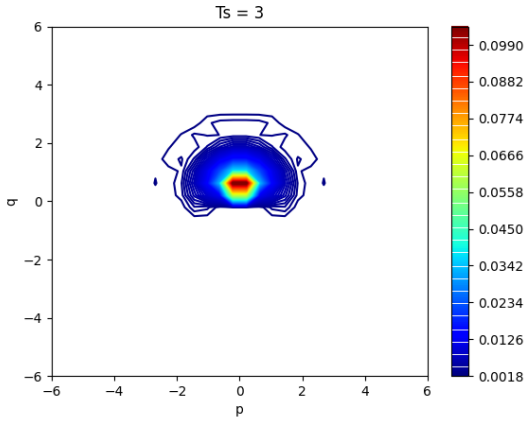

Figure 13: Reconstruction when $\epsilon = 0.001$.

We tried to reconstruct the initial condition (figure 8) from this and changed the iteration number = 200. The calculation time is about 267 sec, and we change the value less than 0.002 to 0 (figure 12).

When $\epsilon = 0.003$, the initial condition is figure 14, and we reconstructed from this with the same parameters of $\epsilon = 0.003$. The calculation time is 263 sec, and the result is figure 15 with changing the value from less than 0 to 0.

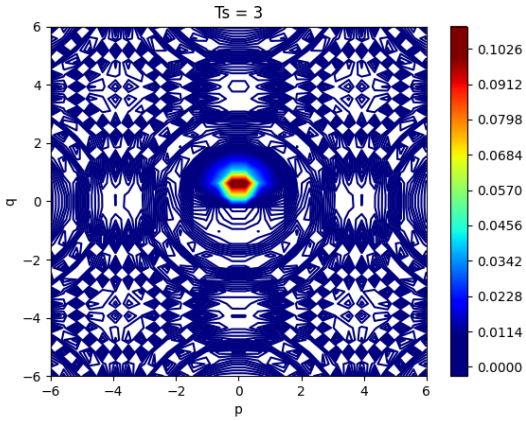

Figure 14: $\epsilon = 0.003$

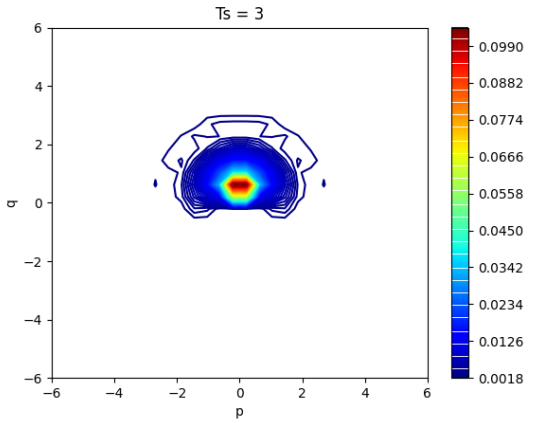

Figure 15: Reconstruction when $\epsilon = 0.003$.

From these results, we can almost reconstruct the initial condition with regularization terms and changing minimum values. However, it is difficult to determine what parameters to use to reconstruct the initial conditions. In the future, we need to find out a good way to set the parameters and a stronger system to reconstruct the wave spectrum without changing minimum values.

5 Further Research

5.1 Improving the Gradient Descent Algorithm with Backtracking

The main improvement that we would like to add to the gradient descent algorithm is the Armijo and Wolfe conditions mentioned in section 2.3. These conditions would hopefully increase the rate of convergence of the algorithm. In addition to improving convergence speed, the Armijo and Wolfe conditions would also improve the ability of the gradient descent algorithm to avoid local minima and locate a global minimum. In order to implement these conditions, we would use a backtracking line search algorithm. The step size for a backtracking line search at the n^{th} step in the gradient descent is defined by

$$\alpha_n = M\beta^{m_n} \tag{5.1}$$

where $M > 0$ and $\beta \in (0, 1)$ are constants, and m_n is the first non-negative integer such that both the Armijo and Wolfe conditions are satisfied. The backtracking algorithm iterates through test step sizes

$$\alpha = M\beta^{m} \tag{5.2}$$

over m until the conditions are met, and then the n^{th} step size for the gradient descent is defined accordingly. One issue with this algorithm is that for poorly chosen M and β, the Armijo and Wolfe conditions may never be satisfied for the tested step sizes. In section 2.3, we proved the existence of a step size that satisfies both conditions. However, this does not

guarantee that the backtracking line search will find such a value. The backtracking algorithm tests for the conditions at discrete values defined by equation 5.2. When moving from one test step to the next, some information about the cost function is ignored. Referring back to figure 4, we can see that the acceptable regions are positive intervals on the real axis that do not contain 0 as an endpoint. This is an issue because it is possible for a backtracking search to skip over these acceptable regions, and if the test step ends up between 0 and the infimum of the acceptable regions, then the algorithm will be stuck in an infinite loop. In addition to skipping over acceptable regions, poorly chosen backtracking parameters may also result in the test steps skipping over a global minimum, which could lead to an inaccurate result. In order to gain more information about the line search, we need to use a larger backtracking parameter β. This will give a better chance of finding a value in an acceptable region, but it may result in unnecessary calculations if the test step is far from an acceptable region.

The backtracking algorithm requires us to choose two parameters: the max possible step M and the backtracking parameter β. The magnitude $\|S\|_{L^2(\mathbb{R}^2)}$ depends on the significant wave height and frequency. Therefore, if reasonable bounds can be obtained for either of these values, the Kataoka model [6] could be integrated for these extreme values to obtain a bound $\mathbf{S}$ for $\|S\|_{L^2(\mathbb{R}^2)}$. Then for two arbitrary wave spectra S_1 and S_2,

$$\|S_1 - S_2\|_{L^2(\mathbb{R}^2)} \leq \|S_1\|_{L^2(\mathbb{R}^2)} + \|S_2\|_{L^2(\mathbb{R}^2)} \leq 2\mathbf{S}. \tag{5.3}$$

If bounds for the wave height and frequency cannot be obtained, the max step size could be determined experimentally. The backtracking parameter is much more difficult to determine because of the unpredictable behavior of the cost function 3.2. One strategy for deciding the backtracking constant would be to start with large values of β, which give the most information about the cost function. Since a large value of β would result in the backtracking algorithm scanning the cost function for detail, these values would be expected to locate the correct minima. Then the value of β could be decreased to a β_l where the gradient descent no longer converges to the desired minima. This would provide a lower bound for β. Running this procedure for many different gradient descent trials will produce a set B of β_l. Then we can choose β to be greater than or equal to $\sup(B)$.

5.2 Testing Different Wave Spectra

Another way to improve the gradient descent is to test the gradient descent algorithm by attempting to recover more complicated wave spectra. The main wave spectra we would consider would be a summation of wave spectra from the Kataoka model [6]. Using a sum of two or more wave spectra with different parameters would allow us to represent surface waves composed of multiple wave trains. Testing the gradient descent algorithm on more complicated wave spectra could allow us to fine-tune the parameters to work in a broader range of scenarios. This may make the gradient descent code more accurate when testing on a Doppler spectrum of unknown wave spectra.

5.3 Environmental Effects

Lastly, to optimize our algorithm, we would like to use environmental factors to try to determine a good initial guess for the gradient descent. We hope to use the Kataoka model [6]

for starting spectra. However, this requires us to define parameters for the significant wave height H_s, significant wave period, T_s, and wave direction θ_0. There are many environmental factors that influence surface waves, including ocean currents and ocean depth.

First, we will discuss the effects of currents. The main effect of ocean currents is that wave trains tend to align with ocean currents [13]. Therefore, if we have information about the dominant currents in the radar range, then we can use this to define a direction for the initial wave spectrum. It is important to note, however, that the current direction and dominant wave train direction do not necessarily coincide. Therefore, it might be useful to look for an alternative method to determine the dominant wave train direction. When water currents oppose the wave currents, the waves are slowed. Since the wave energy is determined by its speed and height, conservation of energy causes the wave height to increase [13]. Therefore, if we have knowledge about the average wave height in the radar range, then data about the ocean currents could be used to determine what wave height to use for our initial guess.

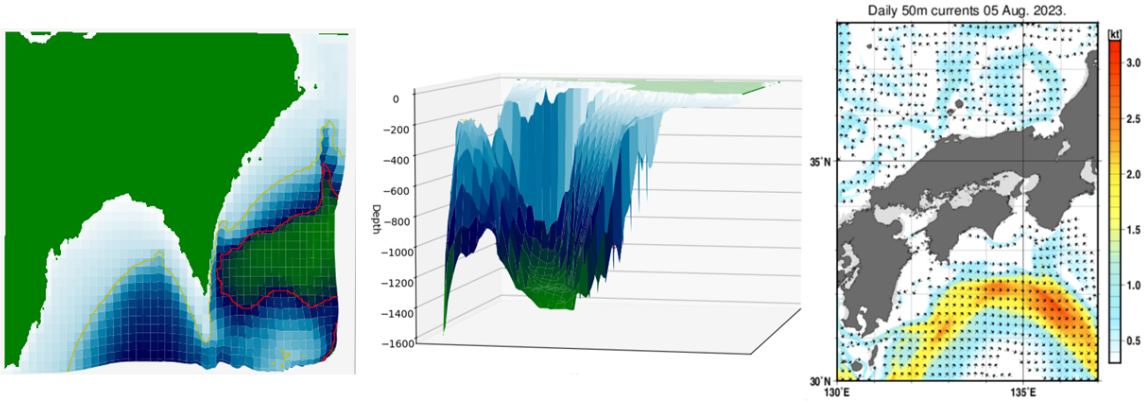

Figure 16: Bathymetric maps of Muroto coast and Ocean Current map

The ocean depth also affects the wavefronts. In particular, shallow areas tend to cause waves to move slower. Again, this leads to an increase in wave height. Additionally, if the wavefronts are not aligned with the ocean depth contours, then the slowing in shallower waters causes the wave direction to change due to refraction. This could also be a useful factor in determining the wave direction for an initial guess. However, it would still require some initial knowledge of the wave direction at some ocean depth to be useful. Figure 16 shows the depth of the ocean around the Muroto coast. The yellow line represents a depth of 200m. The red line represents a depth of 1000m. At this contour line, there is a sudden drop in ocean depth. Waves move faster in deeper water, but as they enter shallow water, they slow, and the conservation of energy increases wave height. We hope that these effects can be explored in more detail to determine how this affects the parameters in our initial guess.

References

[1] Peter Mathé Abhishake Rastogi Gilles Blanchard. *Convergence Analysis of Tikhonov Regularization for Non-Linear Statistical Inverse Problems*. 2020.

[2] Japan Meteorological Agency. *Daily 50m Current*. 2023. URL: https://www.data.jma.go.jp/kaiyou/data/db/kaikyo/daily/current_HQ.html?areano=8&stryy=2023&strmm=08&strdd=03.

[3] D. E. Barrick. "Extraction of Wave Parameters from Measured HF Radar Sea-Echo Doppler Spectra". In: *Radio Sci, 415–424* (1977).

[4] Christian Cason. *Regularization of Inverse Problems*. Feb. 2021.

[5] J.M.J. Journée and W.W. Massie. *Offshore Hydromechanics*. Delft University of Technology, 2001.

[6] Tomoya Kataoka. "A Bayesian Inversion for a Directional Spectrum of Ocean Waves in Shallow Water Using HF Radar". In: (2016).

[7] Thuy T. Le and Loc H. Nguyen. "The gradient descent method for the convexification to solve boundary value problems of quasi-linear PDEs and a coefficient inverse problem". In: *Journal of Scientific Computing* (2022).

[8] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 2006.

[9] J. Paduan and H. Graber. "Introduction to High-Frequency Radar: Reality and Myth". In: *Oceanography 10* (1997).

[10] Y. Park et al. "Parameter Determination for Tikhonov Regularization Problems in General Form". In: *Journal of Computational and Applied Mathematics* (2018).

[11] Aditya Tiwari. "Understanding Overfitting and Underfitting in Machine Learning". In: (2019).

[12] S. Twomey. *Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements*. 1977.

[13] Trevor Whittaker and Jonathan Hancock. *Environmental Loads, Motions, and Mooring Systems*. 2020.