Inverting for Near Shore Bathymetry from Surface Wave Properties

Lasith Adhikari[1], Charnelle Bland[2], Lopamudra Chakravarty[3], Wenbin Dong[4], Olaniyi Samuel Iyiola[5] Gail Muldoon[6] Clint Seinen[7]

Mentors: Lea Jenkins[8], Ty Hesser[9], Matthew Farthing[10]

Abstract

Tracking coastal bathymetry is necessary for marine navigation, military activities, and assessment and prediction of storm damage as beaches evolve. Past efforts have derived surface wave properties from in situ bathymetric measurements. However, in situ measurements are costly and laborious to collect. As a result, direct observations of bathymetry are sparse in time and space. On the other hand, remotely sensed observations of surface conditions are becoming easier to obtain. We seek methods to invert for bathymetry given surface conditions.

We utilize linearized wave theory to estimate bathymetry near Duck, North Carolina given measurements of surface wave properties collected by the U.S. Army Corps of Engineers. We process observations of wave height, wave number, and bathymetry for assimilation into and validation of computational models of wave mechanics. We create a forward model to estimate wave number and wave height given bathymetry information. Several inverse methods including nonlinear least squares, Bayesian Markov Chain Monte Carlo, and Tikhonov regularization then use the surface wave data and forward model to estimate bathymetry along a one-dimensional profile.

Our results demonstrate reasonable estimates of depth, h, in the near shore region, within 500 m of the beach. All inverse methods are able to accurately reconstruct a sandbar located in this region which is an important feature for consideration of rip tides and coastal navigation. Accuracy of the methods drops off past $\sim$550 m from shore due to the increased depth and subsequent lower sensitivity of wave number to depth. We suggest several avenues for possible expansion of this work in the future which may improve overall accuracy of bathymetry estimates.

1 Introduction

Bathymetry is a measurement of submarine topography and can be used to understand shifts of the ocean floor and its depth. Knowledge of bathymetry is important for marine navigation,

[1]Applied Mathematics, University of California, Merced
[2]Mathematics & Statistics, University of West Florida
[3]Mathematical Sciences, Kent State University
[4]Civil Engineering, City College of New York
[5]Mathematical Sciences, University of Wisconsin-Milwaukee
[6]Geosciences, University of Texas Austin
[7]Mathematics & Statistics, University of Victoria
[8]Clemson University
[9]U.S. Army Corps of Engineers
[10]U.S. Army Corps of Engineers

both civilian and military, as well as for monitoring and predicting the effects of storms on coastal environments. While direct measurement of bathymetry is possible, the process tends to be cost and time prohibitive. For example, amphibious vehicles are capable of spatially limited surveys of bathymetry in difficult surf-zone conditions but require significant resources to operate. As a result of these factors, surveys tend to be sparse in time.

The research we conducted for this report focuses on a method to estimate bathymetry, using surface measurements collected via remote sensing platforms, i.e. airborne, satellite, or onshore platforms. While bathymetry data is currently sparse due to observational limitations, the physics of waves are reasonably well understood. In particular, a dispersion relationship can be used to relate water depth to surface properties such as wave length and wave period. This relationship makes it possible to estimate bathymetry given the observations of these parameters. Light Detection And Ranging (LiDAR) has been used to determine wave heights and Argus land-mounted video has been analyzed photogrammetrically to determine wave frequency and wave number. Moreover, these resources provide valuable inputs for estimating coastal bathymetry in a more efficient manner than is currently available.

Wave and bathymetric data has been collected in Duck, NC by the U.S. Army Corps of Engineers Coastal and Hydraulics Laboratory, including in situ measurements of bathymetry and measurements of the water surface. These measurements provide a method for testing algorithms to invert for bathymetry because the true bathymetry is available for comparison to the numerical estimates.

We invert for depth, h, using wave number with a 1D model derived using the energy flux method to create a correlation between wave length and depth from the water surface as shown in Figure 1.

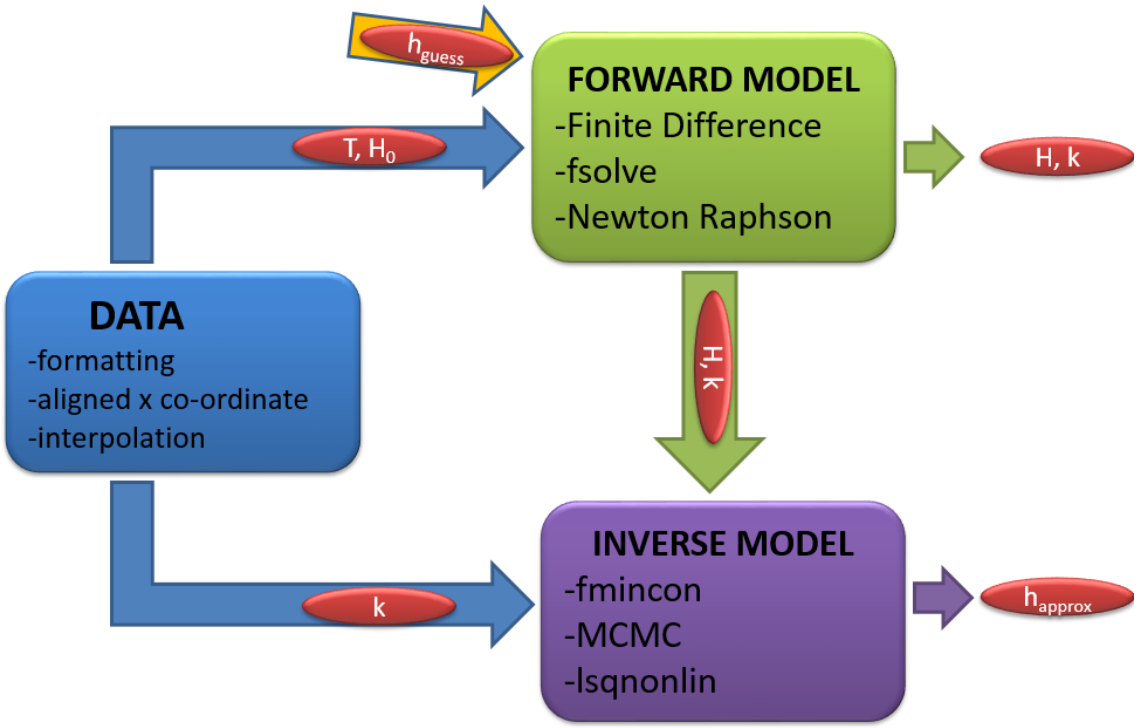

Figure 1: Flow chart of the workflow for this research

In section 2 we discuss ~~about how data is observed~~ and use to measure different wave parameters. ~~In section 3 we discuss about the forward problem and numerical results of forward problem.~~ Inversion method is discussed in section 4 with discussion related the methods applied. ~~In the last section 5 we discuss about the experimental results.~~ ~~These results were generated using both simulated and real data using some existing tools.~~

2 Data

Data for this project were collected by the U.S. Army Corps of Engineers (USACE) Engineer Research and Development Center (ERDC) during October 2015 at the Field Research Facility (FRF) in Duck, NC on the Outerbanks shown in Figure 2.[11] The data was collected via the BathyDuck project conducted by the Coastal and Hydraulic Laboratory (CHL). Data of interest includes wave height (H), wave number (k), wave period (T), and bathymetry (h) measurements. These data combine information collected through a Nortek Acoustic Wave and Current (AWAC) Profiler, a Light Amphibious Resupply Cargo (LARC-5) vessel, a Coastal Research Amphibious Buggy (CRAB), and Argus Beach Monitoring systems. The LARC-5 and CRAB vessels are shown below in Figure 3.

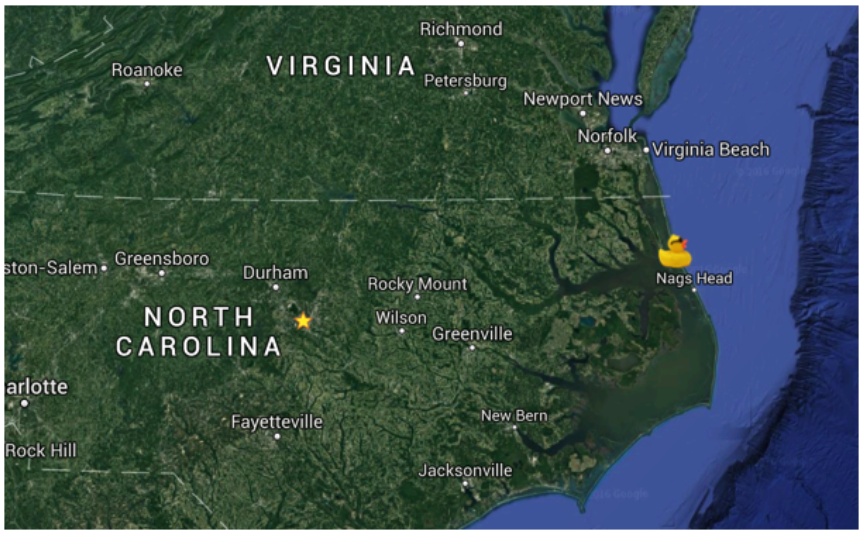

Figure 2: The location of the U.S. Army Corps of Engineers Field Research Facility in Duck, NC.

[11] The data is available in netcdf format at:
http://chlthredds.erdc.dren.mil/thredds/catalog/frf/projects/bathyduck/catalog.html

Figure 3: The LARC (left) and CRAB (right) instruments are used to measure near coastal bathymetry. Image source: http://www.frf.usace.army.mil/aboutUS/equipment.shtml

The following sections discuss in more detail the observations and how they were used. Note in physical space, the boundary point used for the 1-dimensional problem is located 1150 m offshore. For numerical simplicity, all observations are transformed such that $x = 0\,m$ corresponds to the offshore boundary point and $x = 1150\,m$ is the shoreline.

2.1 Boundary Condition

Boundary conditions for this project were collected through a bottom-mounted AWAC profiler located approximately 1150 meters offshore at a depth of 11 meters in Figure 3.

Figure 4: Acoustic Wave and Current Profiler. Image source: http://www.nortek-as.com/en/products/wave-systems/awac

Vast amounts of wave data has been collected by the AWAC system at the offshore boundary. For the 1-dimensional problem, the forcing condition at this boundary is the significant wave height and peak frequency.

2.2 Bathymetry

Survey data collected on 1 October 2015 was considered for this analysis. These data were measured via the CRAB and Trimble Real Time Kinematic (RTK) GPS system. Elevation data from six cross-sections, perpendicular to the shoreline, spaced over a 100-meter portion of the beach were combined to create the 2D surface shown in Figure 5.

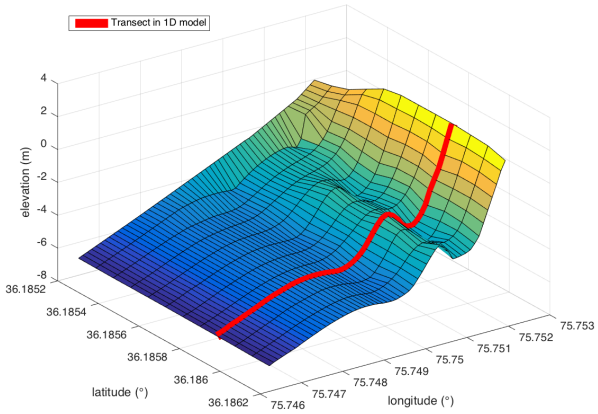

Figure 5: Measured and gridded 2D bathymetry in the survey area on 1 October 2015. The red line shows the transect considered in the 1D problem.

For the 1D problem, a single slice of the 2D bathymetry was used as model input, identified by the red line in Figure 5. In a cartesian coordinate system, this line (a.k.a. 'transect') is located at $y = 950$ meters.

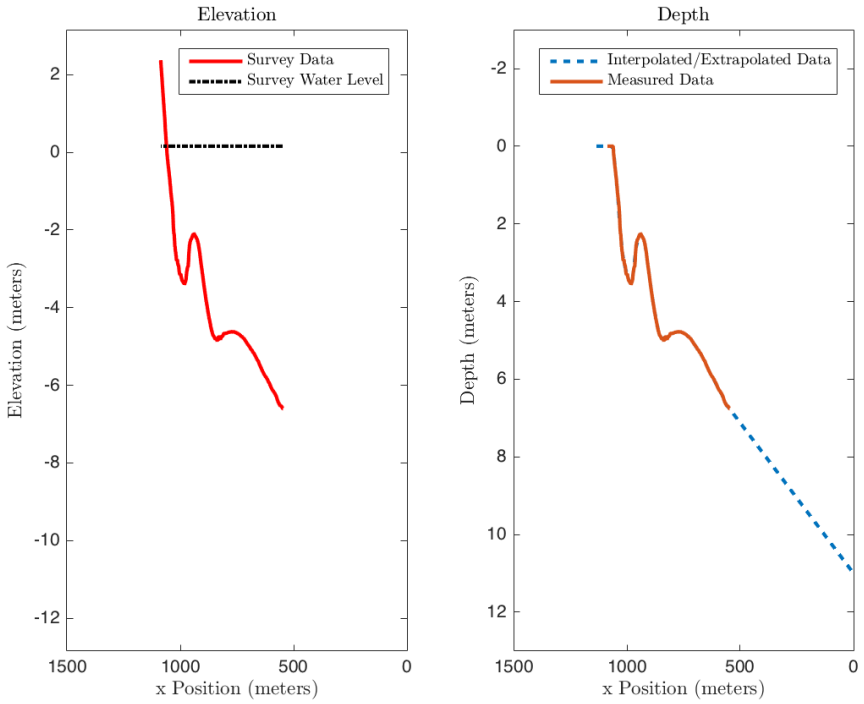

Figure 6: 1D Bathymetry - elevation data (left) and depth data (right).

Survey data provided sea floor elevation referenced to the North American Vertical Datum of 1988 (NAVD88) (Figure 6, *left*). To provide proper input data for the 1D model, elevation was transformed to depth data (Figure 6, *right*). Once transformed, depth was discretized by interpolating between measured data points via Matlab's built-in pchip method. Pchip was chosen for the interpolation due to its shape-preserving nature so as to not introduce non-physical oscillations. Between the boundary condition and the nearest measured depth point, linear interpolation was used to fill in missing data.

2.3 Wave number

Wave number is a measure of the number of waves per unit distance and is inversely proportional to wave speed. Hourly observations using an Argus video monitoring system mounted on shore are available during October 2015. Photogrammetry is performed on the video to derive the dominant wave frequencies and wave numbers in the survey area [4]. Data is available for a 2D area at the FRF survey site. A 1D profile is extracted from the 2D data along a transect corresponding to the position of the model boundary point ($y = 950$ m). Figure 7 shows statistics for wave number, k, along the 1D transect. Wave number is shown to be slightly more variable over time close to the coast. Mean wave number increases toward the shoreline, as expected.

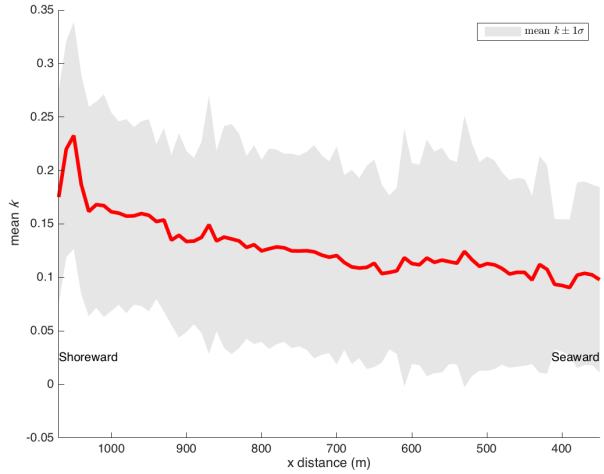

Figure 7: Wave number along the transect where the model boundary condition is located. Mean wave number, k, during October 2015 is shown in red. Gray envelopes show $\pm$ 1σ standard deviation in k. Wave number is observed to be relatively larger and more variable closer to shore.

3 The Problem

Although there have been uncertainties in capturing the topography of the ocean near shore, mathematical methods can estimate bathymetry using the dispersion relationship between wavelength and the period. Stockdon et al. used video imagery, which compared true wave signal and remotely sensed video signal to create a linear representation between wave amplitudes and phases [8]. Holman et al. used a 2-dimensional method with Kalman filtering to estimate the depth, h [4].

3.1 Forward Problem

Though there are many wave dynamic models showing interactions among wave parameters, our interest is on the near shore zone which have a relatively shallow water depth and small wave amplitude (mainly gravity wave). Based on the basic assumptions such as, incompressibility, inviscid and small amplitude, shallow water model with linear wave is therefore an appropriate model to consider. In [3], the shallow water model is constructed by continuity equation and simplified Navier-Stokes equation of momentum. However, the constraint is due to little or no data related to the velocity distribution under water. Therefore, the energy flux method is considered the most appropriate considering the data available to us. Energy flux method in one dimension is given as [1]

$$\frac{d}{dx}\left(E\, C_g\right) = -\delta \tag{1}$$

where $E\,C_g$ represents the wave energy flux which is the product of wave energy E and linear theory group speed C_g given respectively as

$$E = \frac{1}{8}\rho g H_{rms}^2, \tag{2}$$

and

$$C_g = \frac{c}{2}\left(1 + \frac{2kh}{\sinh(2kh)}\right), \tag{3}$$

with water density, $\rho = 1000\,kg/m^3$, the gravitational acceleration, $g = 9.8\,m/sec^2$, the local wave phase speed, $c = \frac{\sigma}{k}$, k is the wave number, h is the water depth and H is the wave height and the wave period, T, is assumed to be constant.
The root mean square of height H_{rms} is defined as

$$H_{rms} = 0.707H$$

.
We consider the wave breaking function δ proposed by [1], which provides local mean rate of energy dissipation, given as

$$\delta = \frac{1}{4h}B\rho g f H_{rms}^3\left[(R^3 + \frac{3}{2}R)exp(-R^2) + \frac{3}{4}\sqrt{\pi}(1 - erf(R))\right], \tag{4}$$

with the local wave breaking height, $H_b = 0.78h$, $R = \frac{H_b}{H_{rms}}$ and the frequency, $f = \frac{1}{T}$.

Dispersion relationship can describe effect of the dispersion depending on the properties of wave in various medium including water. This way, it gives the relations between different wave dynamics, for example depth h and wave number k. For this reason, dispersion relationship is an important component in our study [3]

$$\sigma^2 = gk\tanh(kh) \tag{5}$$

where $\sigma = \frac{2\pi}{T}$ is the angular frequency.

3.2 Numerical Solution of the Forward Model

Upwind scheme is applied to obtain a numerical solution with appropriate initial and boundary conditions. The goal is to provide wave height H. In the process, MATLAB function fsolve is applied to obtain the wave number. Furthermore, the Newton-Raphson method is used to verify the solution of wave number k obtained by the fsolve function.

3.2.1 Discretization and Implementation of the Model

We set spatial variable $x = 0$ at the shore and $1150\,m$ apart we set the x_N. Thus we consider the mesh spacing either as $\Delta x = 10$ or as $\Delta x = 25\,m$. It means the index points are set in either $10\,m$ or $25\,m$ apart.We have applied equal spacing for this numerical scheme.. We provide the period T as constant. Together to this we apply the boundary H_0 taken from the real data.

The wave number k plays an important role in estimating the wave height H. To calculate the wave number MATLAB function fsolve used as non-linear solver to find k from dispersion relationship (5) At each index point, wave number, k, is generated with initial guess k_0:

$$k_0 = \frac{\sigma}{\sqrt{gh}}.$$

This initial condition is chosen because in shallow water $tan(k\,h) \simeq k\,h$ the dispersion relation provides

$$\sigma^2 \simeq g\,k^2\,h$$

Newton-Raphson method is applied to verify the wave number from fsolve. Therefore, using same initial condition the approximate solution is obtained from

$$k_{i+1} \;=\; k_i - \frac{gk_i \tanh(k_i h) - \sigma^2}{g\tanh(k_i h) - ghk_i sech^2(k_i h)}, \tag{6}$$

Thus at each index forward finite difference expression is calculated as

$$E_i = \frac{\delta_{i-1}\Delta x}{(C_g)_i} + \frac{E_{i-1}(C_g)_{i-1}}{(C_g)_i}$$

In the above expression δ will be updates its value at each index points using (4). the wave numbers will employed during the calculation of group celerity C_g. After estimating the energy E at each index points we update the value of the root mean square of height H_{rms}. This updated value is applied in the $H = \frac{H_{rms}}{0.7}$ to obtain the wave height at each index points.

The implementation of the algorithm is as follows

Algorithm 1 Algorithm to estimate wave height H

procedure
Initialization:
    Mesh spacing: Δx
    Initial depth: h
    Wave period: T
    Boundray condition of height: H_0
Step 1: Estimate wave number k
- $\sigma = \frac{2\,\pi}{T}$
- $\sigma^2 = gk\tanh(kh)$

Step 3:Compute wave height H
- *Compute wave breaking function:* δ
- *Compute wave group celerity:* C_g
- *Compute:* $H:\quad E_i = \frac{\delta i-1 * \Delta x}{(C_g)_i} + \frac{E_{i-1} * (C_g)_{i-1}}{(C_g)_i}$
- *Update :* $(H_{rms})_i = \sqrt{\frac{8.0\,E_i}{\rho\,g}}$
- *Compute :* $H_i = \frac{(H_{rms})_i}{0.707}$

3.2.2 Numerical Results

We present some of the numerical results in the following Figures.

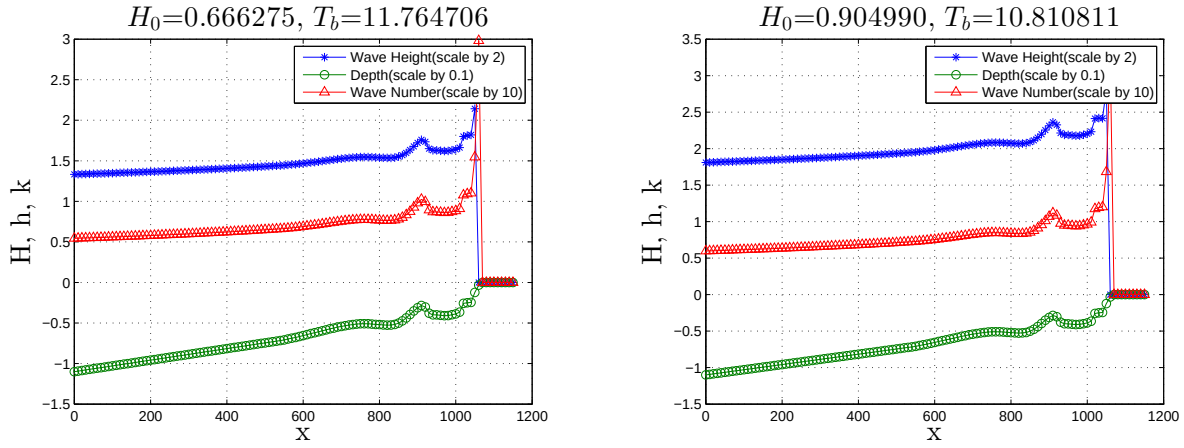

Figure 8: Water Depth(h), Wave Height(H) and Wave Number vary along x direction (Two sets of boundary conditions are applied for following figures. The boundary conditions for the left is extracted from the data of 2015/10/9 22:00-23:00 and for the right is extracted from the data of 2015/10/9 03:00-04:00). Note that $x = 0m$ is located offshore at the boundary point.

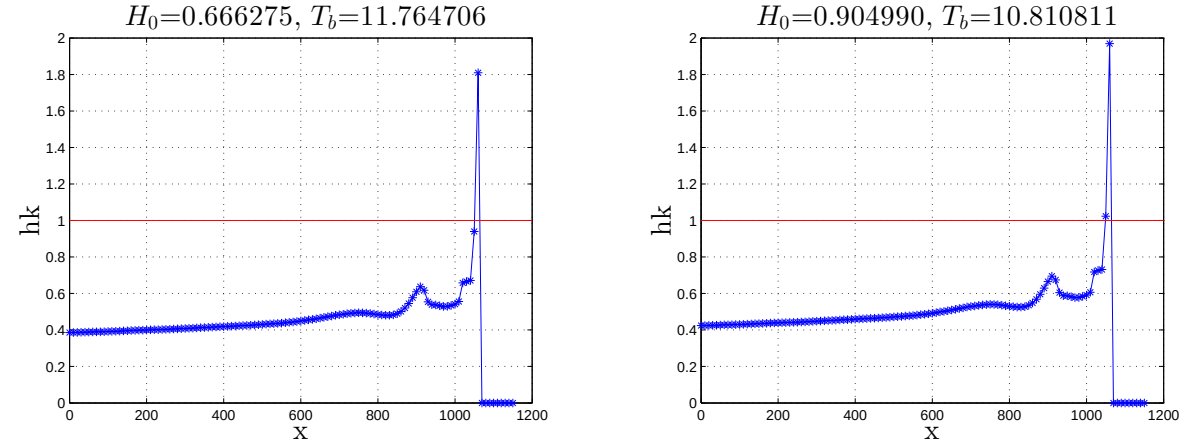

Figure 9: Relative Depth varies with x direction

The following remarks are made from the numerical results obtained from the forward model.

1. Figure 2 shows that the wave height is positively correlated to the depth. We can see the similar characteristics between depth and wave number. This is the reason for why we can perform the simulation by providing H or k to obtain the depth h. The wave height

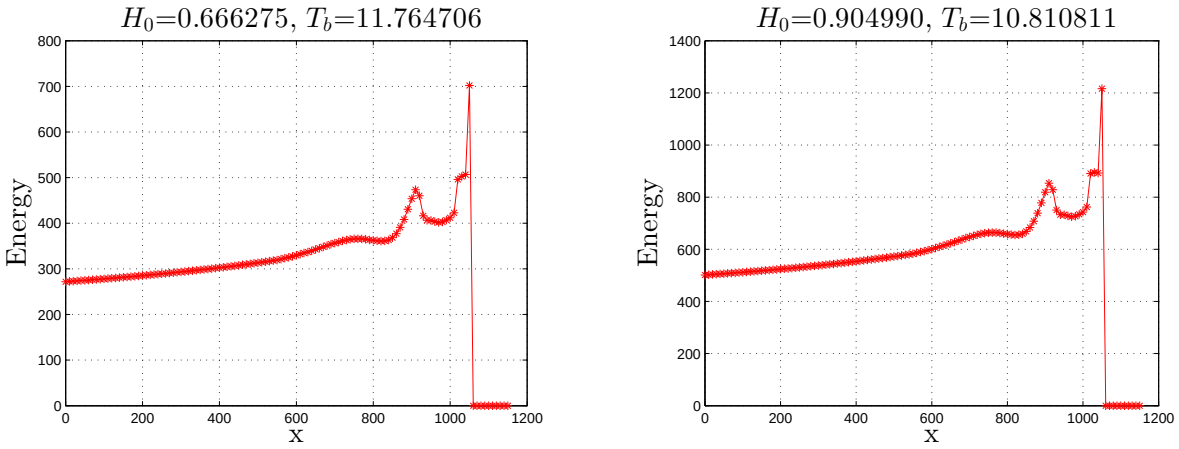

Figure 10: Wave Energy varies along x direction

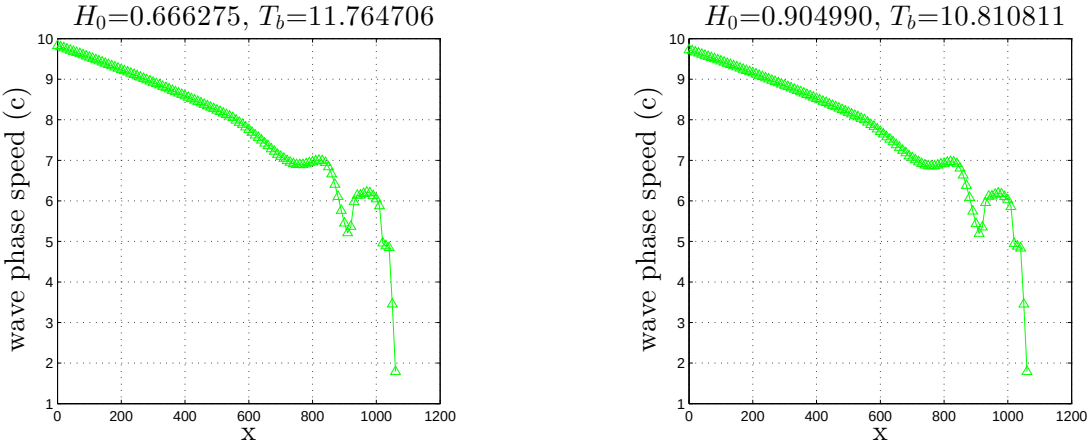

Figure 11: Wave Phase Speed (c) varies with x direction

curve and wave number curve will rapidly increase to a significant level around where depth is very close to zero, and the data there is not reliable because of the poor behavior of equations around $h = 0\,m$ (We need to force wave height to zero when depth is close to zero, or the wave height given by those equations will turn to infinity). However, the shape of wave height curve or wave number curve around the peak of depth curve shows our model has a reasonable good response to the changing depth.

2. Figure 3 shows the shallow water assumption which is $hk \ll 1$ (The red horizontal line is $hk = 1$). We can see that our model is well fit the criteria at most data points except some where their depths are close to zero. Also the relatively flat slope of hk reflected that the wave number k will increase as the depth decreases. The data turns to be zero around $x = 1050\,m$ as the equation cannot provide reliable data when $h = 0\,m$.

3. Figure 4 is the variation of wave energy along x-axis. The energy will be accumulated unless the wave height meets the break condition.

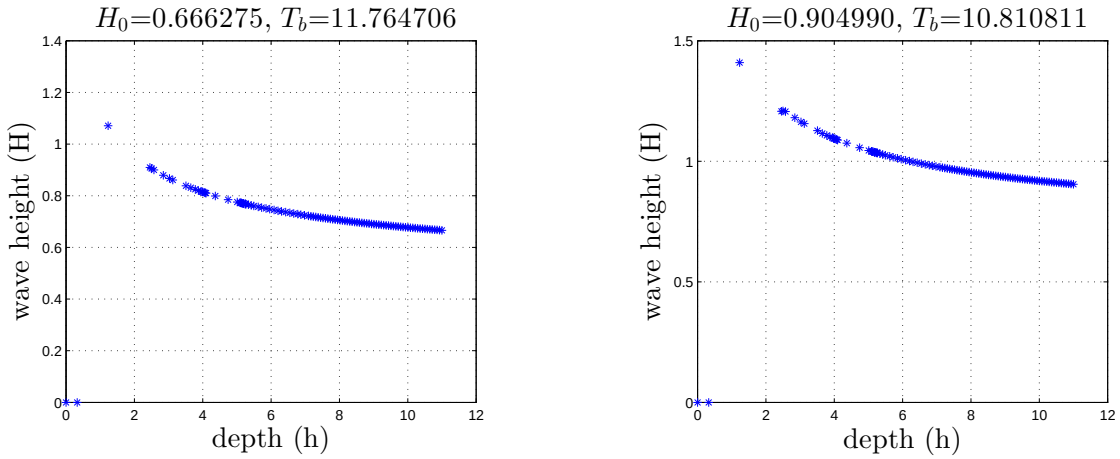

Figure 12: Wave Height(H) varies with Water Depth(h)

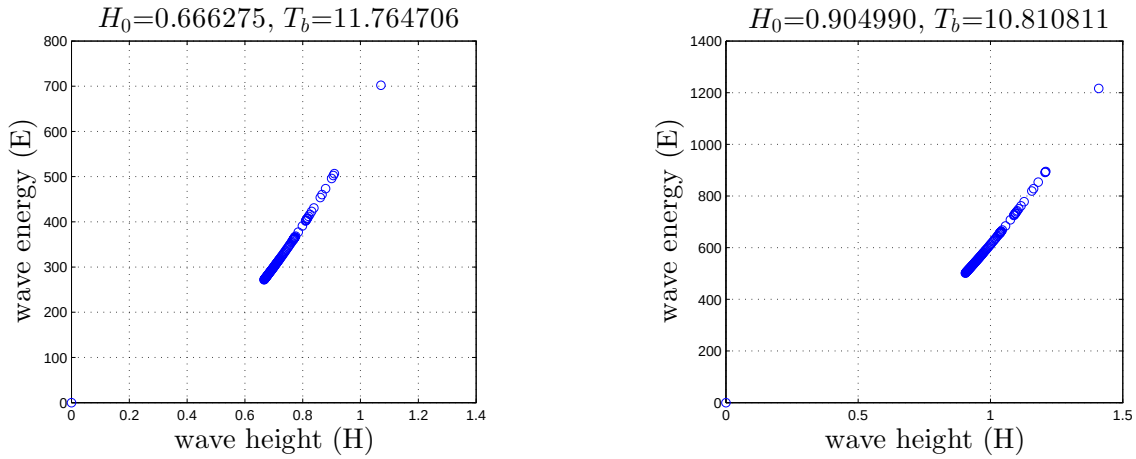

Figure 13: Wave Energy varies with Wave Height(H)

4. Figure 5 is the variation of wave phase speed (celerity) along x-axis. As celerity is a function of wave number and wave period (wave period is assumed to be fixed in our case).

5. Figure 6 shows the variation of wave height along with depth. The data at zeros should not be taken into consideration. We can see that the wave height decreases while the depth increases.

6. Figure 7 shows the variation of wave energy along with wave height. The data at zeros should not be taken into consideration. As the energy is calculated by $E = \frac{1}{8}\rho g H^2$, the curve is a part of $y = x^2$.

7. Figure 8 is the variation of wave energy dissipation along x-axis. Just as the basic assumption of our model, the energy does not dissipate significantly until the depth becomes relatively small (the data after the main peak is not reliable as the depth there

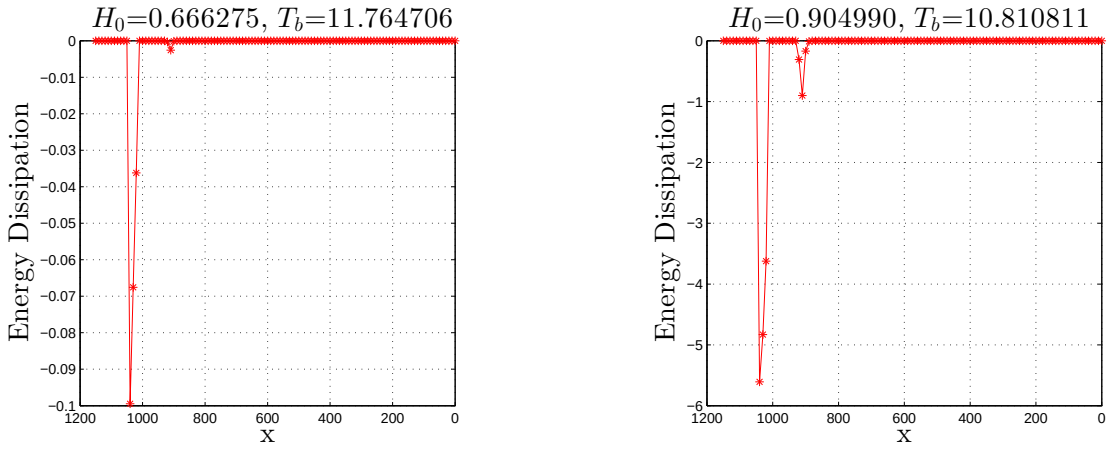

Figure 14: Wave Energy Dissipation varies along x-axis

is too small). The small peak around $x = 900m$ is caused by the peak part of the bathymetry.

4 Inversion Methods

~~Here we perform a preliminary study on potential inversion schemes. We use various inversion~~ schemes to recover an unknown bathymetry using a mock forward model and manufactured data. This preliminary work aided in the selection process of the three inverse methods chosen for this project.

4.1 The Additive Gaussian Noise Model

To accurately mimic the real problem, we create "measurements" that are corrupted by the Gaussian noise; this can be written as

$$\mathbf{d} \sim \text{Gaussian}(\mathbf{A}\mathbf{h}_t), \tag{7}$$

where

$\mathbf{d}$ = a vector of measurements,
$\mathbf{A}$ = a linear forward operator,
$\mathbf{h}_t$ = the true bathymetry (depth).

Defining $\boldsymbol{\epsilon}$ as our Gaussian noise with a standard deviation ν, the corrupted measurements can be represented as

$$\mathbf{d} = \mathbf{A}\mathbf{h}_t + \boldsymbol{\epsilon}.$$

4.2 Ordinary Least-Squares Inversion

To start, we consider the following least-squares problem,

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}\in\mathbb{R}^n} \; f(\mathbf{h}) = \|\mathbf{A}\mathbf{h} - \mathbf{d}\|_2^2, \tag{8}$$

where we minimize the ~~data misfit~~ between the forward predictions and the measurements, ~~in the least-squares sense.~~ In order to test possible Matlab® inbuilt solvers to solve the minimization problem in (8), we have generated dummy measurements with $\nu = 0.1$. In particular, in this test, the forward operator `A = rand(50)`, the true bathymetry `ht = -linspace(-11,0,N)' - 11`, and the Gaussian noise corrupted measurements `b = A * h_t + 0.1 * randn(N,1)`. Recovered bathymetries from different Matlab® ~~solvers~~ are given below. Note ~~that~~ the initial guess for all methods is zero.

(1) *Nonnegative least-squares method:* `lsqnonneg(A,b, options)`. This Matlab® function uses the *active-set* algorithm; note that it requires the matrix $\mathbf{A}$ explicitly. The residual norm error for this nonnegativity reconstruction is 8.88×10^{-26} (see Fig. 15).

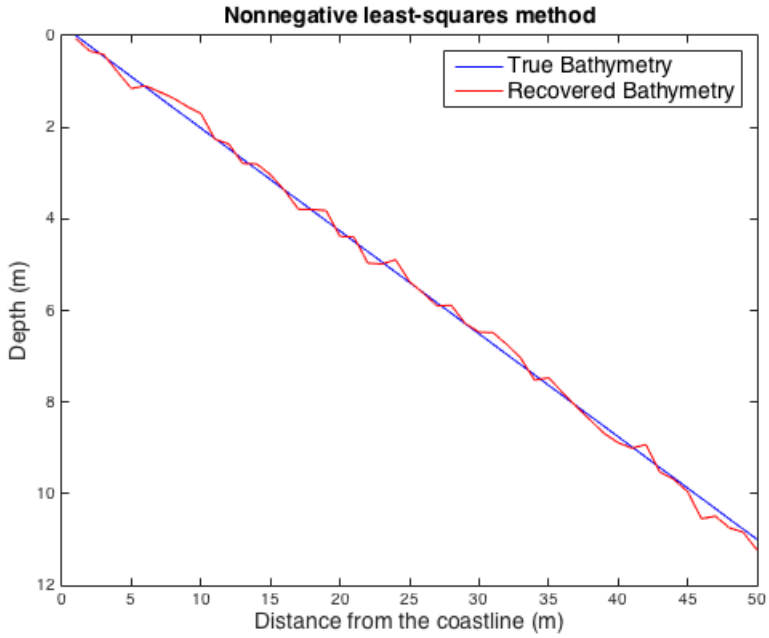

Figure 15: Nonnegative least-squares method reconstruction of depth $\mathbf{h}$ using the dummy dataset.

(2) *Trust-Region-Reflective method:* `lsqnonlin(@(h) A * h - b, zeros(N,1), zeros(N,1),inf(N,1), options)`. ~~Note that~~ the reconstruction of $\mathbf{h}$ is restricted to the positive x-axis using the function arguments. The residual norm error of this reconstruction is 4.32×10^{-10}.

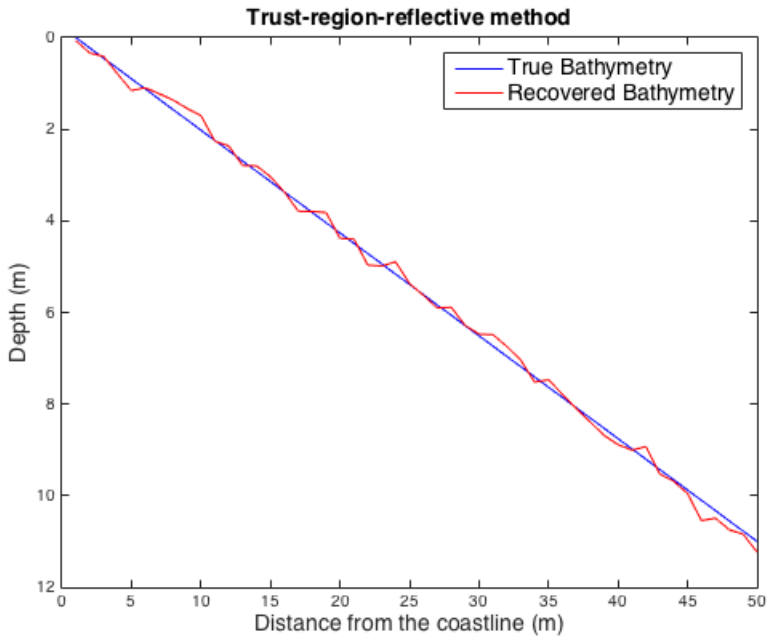

Figure 16: Trust-Region-Reflective method reconstruction of depth $\mathbf{h}$ using the dummy data.

(3) *Levenberg-Marquardt (LM) method:* `lsqnonlin(@(h) A * h - b, 'Algorithm', 'levenberg-marquardt')` Residual norm error for this reconstruction is 6.39×10^{-13}. It should be noted that the Levenberg-Marquardt algorithm does not handle bounded constraints.

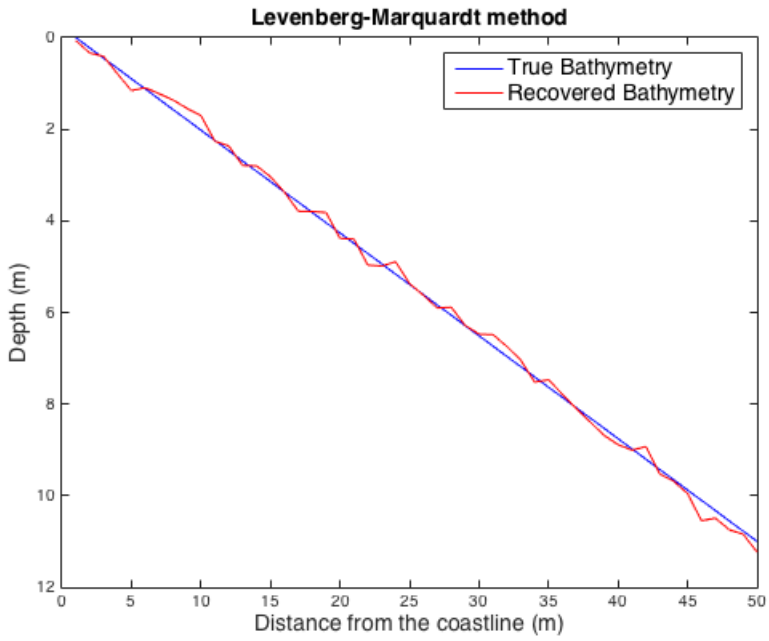

Figure 17: Levenberg-Marquardt (LM) method reconstruction of depth **h** using the dummy data.

(4) *Interior-point method:* `fmincon(f, zeros(N,1), [],[],[],[], zeros(N,1), inf(N,1))`. Residual norm error for the sample dummy data set with $\nu = 0.1$ is 1.29.

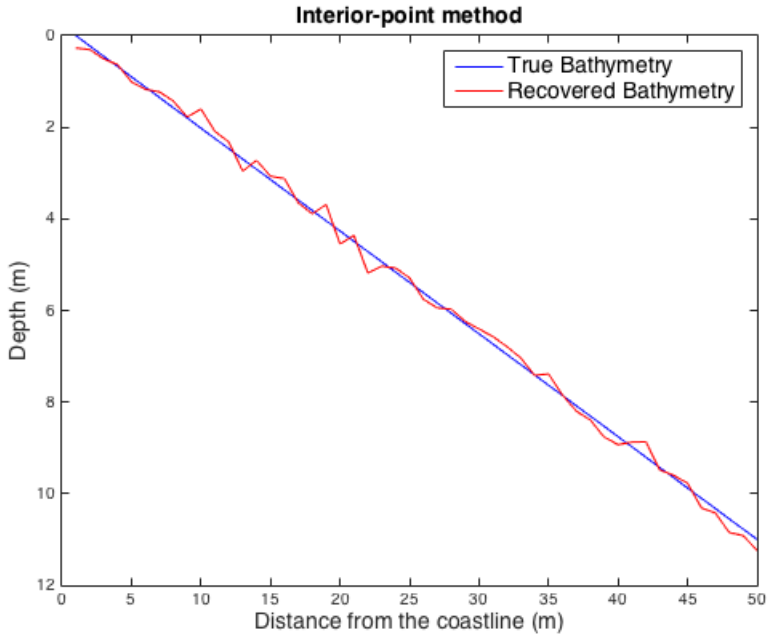

Figure 18: fmincon method reconstruction of depth **h** using the dummy data.

4.3 Tikhonov Regularization

Although the ordinary least squares solution can deal with well-posed problems (i.e. the solution exist and unique), it might yield ~~into a~~ unstable solutions in the presence of ~~a~~ small noise in the measurements. Due to the ill-conditioned nature of the forward operator matrix $\mathbf{A}$, the noise components of the observations can be amplified and ~~leads into~~ a drastic change in the solution [9]. ~~To overcome this problem, we use propose the use *regularization*.~~ The Tikhonov regularization is one of the most widely used regularization techniques in the inverse problems community. ~~It couples the least squares term in (8)~~ with a additional regularization term as defined by

$$\hat{\mathbf{h}} = \underset{\mathbf{h}\in\mathbb{R}^n}{\arg\min} \ \|\mathbf{A}\mathbf{h} - \mathbf{d}\|_2^2 + \alpha\|\mathbf{h}\|_2^2 \tag{9}$$

where α is the regularization parameter ~~(> 0),~~ ~~which balances~~ the trade-off between ~~data fidelity term (i.e.~~ the least squares term~~)~~ and the regularization term $\|\mathbf{h}\|_2^2$. ~~Last but not least,~~ the ordinary least squares method can not incorporate any prior knowledge about the bathymetry, for e.g. any known depths near the shore. However, ~~If~~ we have ~~some~~ prior depth estimate $\mathbf{h}_p$ for the $\mathbf{h}_t$, then the Tikhonov regularized solution with the prior information can be written as

$$\hat{\mathbf{h}} = \underset{\mathbf{h}\in\mathbb{R}^n}{\arg\min} \ \|\mathbf{A}\mathbf{h} - \mathbf{d}\|_2^2 + \alpha\|\mathbf{h} - \mathbf{h}_p\|_2^2, \tag{10}$$

where α is a regularization parameter (> 0). To test this method in (10), Tikhonov method is implemented and tested with sample data set with $\nu = 0.2$. This reconstruction of depth has 0.038 residual norm error. Note that the `tikhonov` Matlab® function requires the matrix $\mathbf{A}$ explicitly in order to solve the regularized minimization problem but the Tikhonov method can be applied through the `fmincon` algorithm when $\mathbf{A}$ is not available.

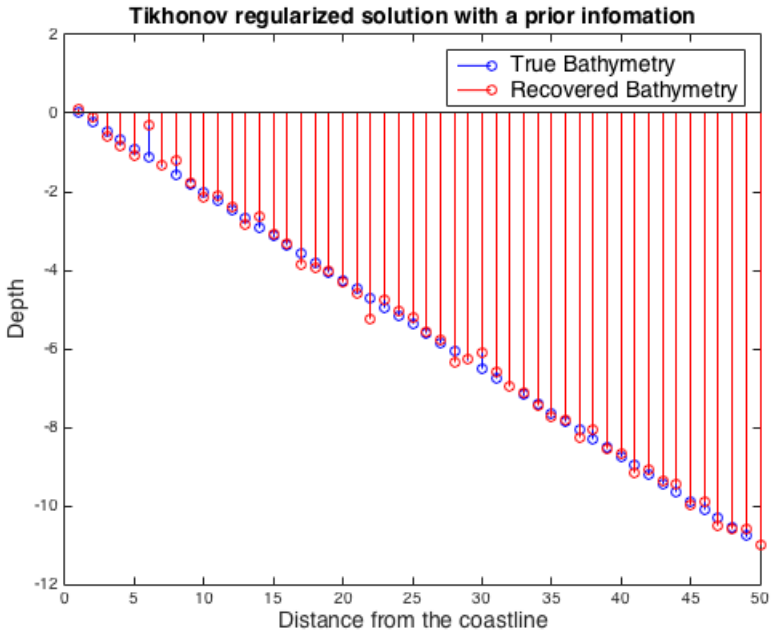

Figure 19: The *tikhonov* Matlab® function reconstruction of the depth using the dummy data.

4.4 Bayesian MCMC Inverse Method

We implement a Bayesian approach because this method uses past information about the parameter, h, to form a prior distribution for analysis and it will sample the posterior probability distribution of "true" h. The recovered distribution of h profiles allows for an intuitive understanding of uncertainty in estimated h.

4.4.1 Bayesian Formulation

Specifically, we use a Bayesian Markov Chain Monte Carlo (MCMC) method to estimate depth, h, given wave number, k. This method computes the posterior probability distribution of h by combining prior information about h with a likelihood function according to Bayes Rule in Equation 11:

$$P(h|k) \propto \Pi(h)L(h|k), \tag{11}$$

where $P(h|k)$ represents the posterior probability function, $\Pi(h)$ is the prior distribution function and $L(h|k)$ is the likelihood function.

We create a random walk Metropolis algorithm [7] which samples proposed depth profiles and updates the posterior probability distribution of h according to the likelihood that the proposed depth profiles describe the true bathymetry.

To estimate depth, we create a prior distribution function of h given 500 realistic approximations of bathymetry derived by the U.S. Army Corps of Engineers. We assume a normal

conjugate prior with mean and standard deviation determined by the distribution of simulated bathymetry as shown in equation 12, this ensures our posterior distribution is normal as well.

$$\Pi(h) \sim N(0, \sigma^2_{sim}) \tag{12}$$

where $\sigma^2_{sim} = 1.01\ m$ is the variance of the simulated h profiles.

Given the sample bathymetry, we evaluate the forward model to compute wave numbers along the 1D profile. A likelihood function (Equation 17) compares the computed k profile to k values observed along the 1D profile.

$$L\left(h|k\right) = \exp\left(-\frac{\sum_{i=1}^{n}(k_{m,i} - k_{d,i})^2}{2\sigma_d^2}\right) \tag{13}$$

Modeled and observed k are $k_{m,i}$ and $k_{d,i}$, respectively, where i corresponds to the points along the 1D profile for which we have inferred k measurements and σ_d^2 is the variance of k values observed along the 1D profile during October 2015. (See section 2.3 for more information about the observations.)

The likelihood uses the sum of square errors between simulated and observed k to quantify the probability that the modeled k represents the true k profile as observed. Inclusion of the σ_d^2 acknowledges that a perfect match between modeled and observed k cannot be expected due to uncertainty in the measurements of k.

4.4.2 Metropolis Algorithm

The Metropolis algorithm begins with an initial h profile sampled from the simulated distribution of bathymetry. This initial estimate of h is used to compute an initial prior probability distribution. An initial likelihood probability distribution is computed using observed values of k, modeled values of k determined by using the sampled h as input to the forward model (see Section 3.1), and σ_d^2. The prior and likelihood are combined to compute an initial posterior probability distribution of h, shown in equation 14:

$$P(h|k) = log[\Pi(h)] + log[L(h|k)] \tag{14}$$

The algorithm then uses a markov chain random walk to propose new h profiles. Each proposed h is evaluated in the forward model to get a proposed k profile. The proposed k is then compared to observed k using the likelihood function and a proposed posterior distribution is computed.

A unique characteristic of the Metropolis algorithm is the way in which proposed posteriors are accepted or rejected by comparison to the previous step's posterior probability. The probabilty of accepting a proposed posterior is

$$\rho = exp[P(h|k)_{prop} - P(h|k)_{current}] \tag{15}$$

The formulation in equation 15 implies that if the proposed posterior probability is higher than the current posterior probability then the proposed posterior probability replaces the

current posterior probability and the proposed h is accepted as a solution. The algorithm is then incremented and a new proposal is selected and evaluated.

4.4.3 Verification

The Metropolis algorithm used for this project was developed by the authors. To verify the algorithm is working correctly, it was tested on a simple model:

$$y = \alpha x \tag{16}$$

where the algorithm estimates the value of α. The true value is known to be $\alpha_{true} = 2$. Figure 20 shows that our implementation of the random walk Metropolis algorithm is functioning correctly. We are able to accurately estimate the value of the parameter α.

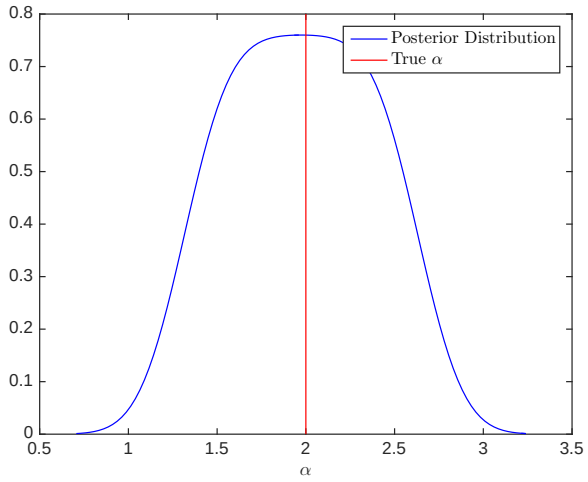

Figure 20: Verification that the Metropolis algorithm has been properly implemented in our code for a simple ODE problem. The red line represents the known value of the parameter α. The blue curve is the posterior probabilty distribution sampled by our algorithm. The posterior we compute accurately estimates the true value of α.

For this simple example we add gaussian noise to a sample dataset. The likelihood function is therefore gaussian to account for the gaussian errors:

$$L\left(h|k\right) = \exp\left(-\frac{\sum(\alpha_{model} - \alpha_{data})^2}{2\sigma_d^2}\right) \tag{17}$$

we use σ_d^2=1 in this case. We use a standard normal distribution for the prior:

$$\Pi(h) \sim N(0, 1) \tag{18}$$

The proposal step size is 0.5 and the algorithm is iterated 10^4 times to arrive at the our example solution, though fewer iterations would have been sufficient.

5 Estimating Bathymetry

In this Section, we describe some experimental results with simulated and real data using few existing nonlinear optimization tools that we used in our preliminarily experiments (see Section 4). Note that our forward model is nonlinear as described in Section 3.1 and it can not be written as a matrix equation system. Therefore, in this Bathymetry inversion, we can not use any inverting method where it needs forward operator as a explicit matrix operator (e.g., `tikhonov` and `lsqnonneg` Matlab® functions need forward operator as a matrix). Moreover, the bounds of the true Bathymetry in the interested near shore region is known to be $[0m, 11m]$. Therefore, we picked the inverse solvers that enable us to incorporate that prior knowledge too.

5.1 Simulated Data

In this study, Gaussian noise corrupted simulated wave numbers (see Section 4.1 for more details) are generated by

$$\mathbf{k}_s = A(\mathbf{h}_t) + \mathcal{N}(0, \nu^2), \tag{19}$$

where $A(\cdot)$ represents the nonlinear forward operator defined in Section 3.1, $\mathbf{h}_t \in \mathbb{R}^n_+$ is the true Bathymetry vector, and $\mathcal{N}(0, \nu^2)$ is a additive Gaussian noise vector with standard deviation ν, generated in the Matlab® by $\nu \cdot$ `randn(n,1)`. we manufactured wave numbers $\mathbf{k}_s$ for two different grid resolutions, i.e., 10m and 25m (see 21 and Fig. 22 respectively), to test and tune our inverse methods before working with actual measurements.

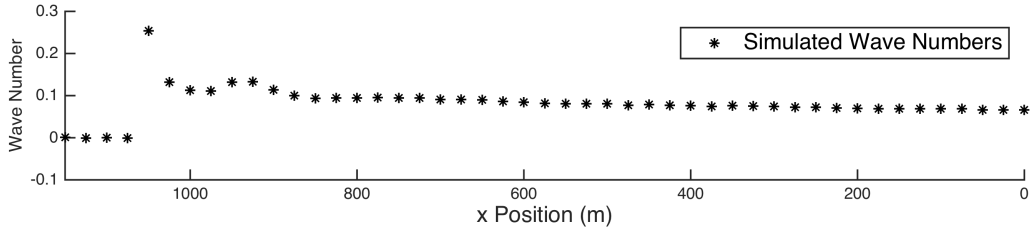

Figure 21: 1% noisy ($\nu = 10^{-3}$) simulated wave numbers ($\mathbf{k}_s$) with 25m grid resolution. Noise (%) $= \|A(\mathbf{h}_t) - \mathbf{k}_s\|/\|\mathbf{k}_s\| \cdot 100$.

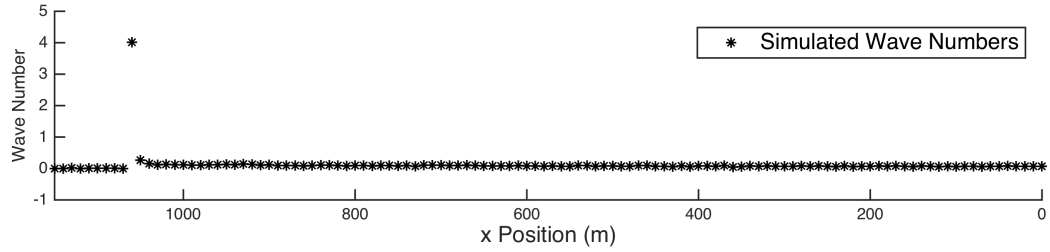

Figure 22: 2.5% ($\nu = 10^{-2}$) noisy simulated wave numbers ($\mathbf{k}_s$) with 10m grid resolution. Noise (%) $= \|A(\mathbf{h}_t) - \mathbf{k}_s\|/\|\mathbf{k}_s\| \cdot 100$.

5.2 Real Data

As discussed in Section 2.3, measured wave numbers, $\mathbf{k}_m$, are provided by the USACE for the month of October 2015 at half hour increments during day light hours. To test our inversion schemes, we choose a wave number profile based off of two constraints; available bathymetry data near the time of the chosen profile and minimal noise in the measured data, which is associated with calmer ocean conditions. According to these constraints, we focus on wave number measured around October 9th as the conditions were relatively calm and a survey executed near that time. Figure 23 shows measured profiles for October 9th, 2015 12:00:00 - 23:00:00 (UTC). It should be noted that multiple wave numbers were provided by the USACE at each measurement location; the dominant wave number were considered in this project. Also note that the measured wave numbers near the shore (x Position > 1050m) and the deep end (x Position < 350m) are not available.

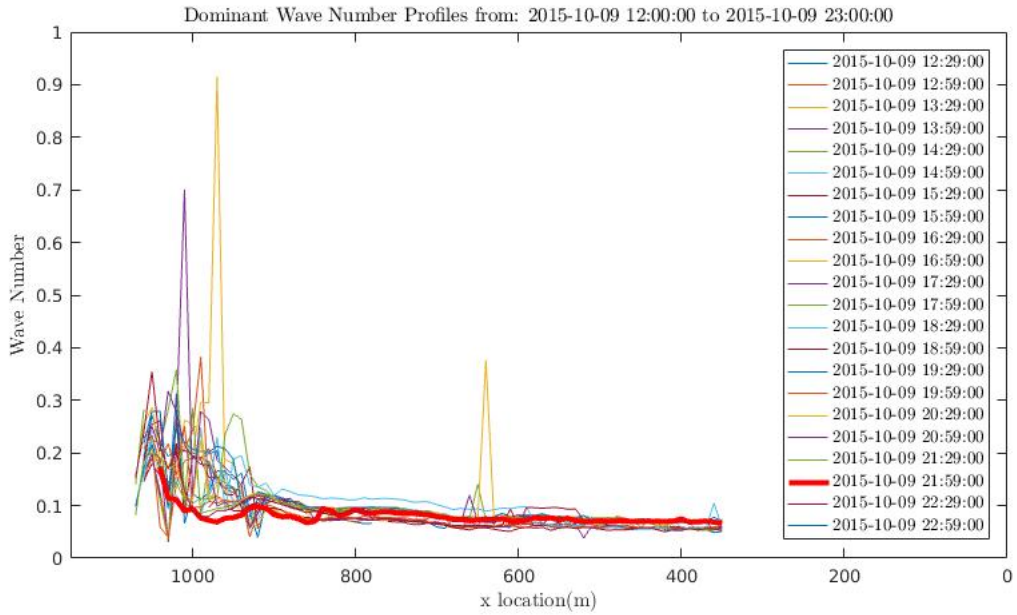

Figure 23: Real measurements of wave numbers (k) on the 09th October 2015.

Upon inspection of each wave number profile from this time period, it is evident that the profile corresponding to 21:59:00 UTC (highlighted in red) contains little noise relative to the others; this profile will be used for our inversion schemes.

5.3 Ordinary Least-Squares Fitting Results

Here, we approximate the unknown bathymetry by solving the following optimization problem:

$$\hat{\mathbf{h}} = \underset{\mathbf{0} \preceq \mathbf{h} \preceq \mathbf{11}}{\arg\min} \ \|A(\mathbf{h}) - \mathbf{k}\|_2^2. \tag{20}$$

In particular, we modify the ordinary least squares problem in (8) to a bounded constrained optimization problem to incorporate the known bathymetry bounds. Based on the preliminary

studies and referring the Matlab® documentation, we decided to use the Matlab's `lsqnonlin` function to solve this problem (20) for the simulated and real $\mathbf{k}$ measurements. According to the Matlab® documentation, here we have two algorithm choices: *trust-region-reflective* (default) or *Levenberg-Marquardt*. However, the Levenberg-Marquardt algorithm does not handle bound constraints. Therefore, we use the trust-region-reflective algorithm (See [5] for more details regarding the algorithm) inside the `lsqnonlin` function to recover the unknown depth parameter. Rather than input the sum of squares of the objective function in (20) to the `lsqnonlin` function, it requires a user-defined vector-valued function $f(\mathbf{h}) = A(\mathbf{h}) - \mathbf{k}$ as the function argument along on the boundary conditions.

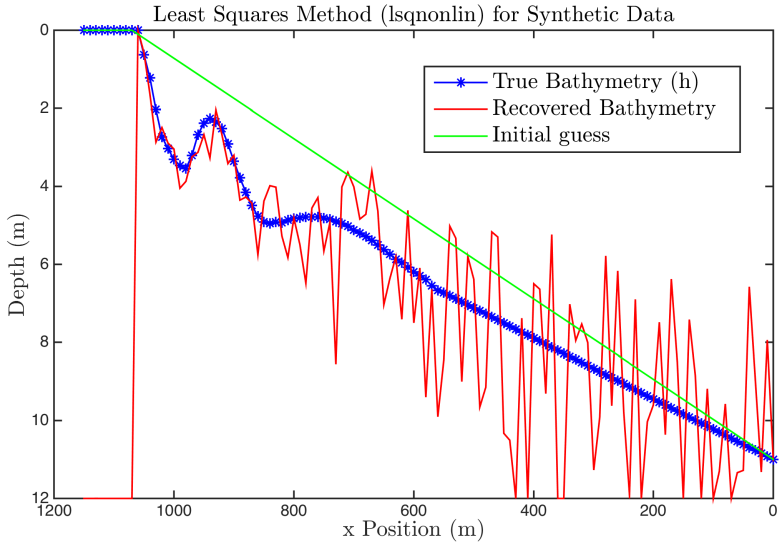

Figure 24: Ordinary least-squares reconstruction of depth $\mathbf{h}$ using the simulated data.

First, we run the `lsqnonlin` function on the synthetic data $\mathbf{k}_s$ with 10m grid resolution. The least-squares method captured the most interesting and important sand bar near the shoreline (see Fig. 24 from 800m to 1000m area). However, the recovered depth is unstable from 600m to the deep-end due to the ill-conditioned nature of the nonlinear forward operator.

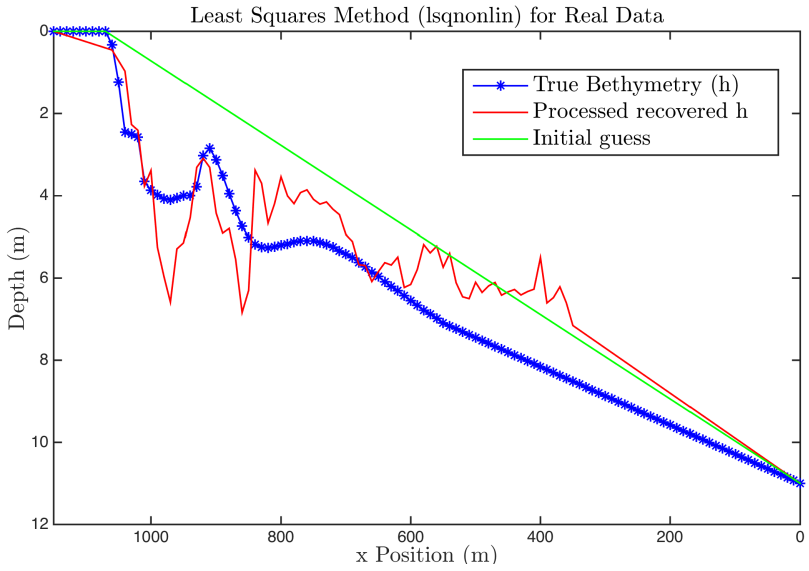

Figure 25: Ordinary least-squares reconstruction of the depth $\mathbf{h}$ using the real data.

After considering the synthetic data, the real wave number measurements $\mathbf{k}_m$ discussed in Section 5.2 are used with the same Matlab's `lsqnonlin` optimizer. In this reconstruction, we capture the near-shore topography including the sand bar with 53% root mean square error (see Fig. 25); errors sea-ward of the sand bar are again attributed to the ill-conditioned nature of this problem in combination with noise in the measured wave number. It should be noted that we have interpolated and processed the depth reconstruction based on the missing measurement locations.

5.4 Tikhonov Regularized Results

In order to get more stable solution than the ordinary least squares solution in Fig. 24, we decided to solve the following bounded-constrained Tikhonov regularization optimization problem:

$$\hat{\mathbf{h}} = \underset{\mathbf{0} \preceq \mathbf{h} \preceq \mathbf{11}}{\arg\min} \ \|A(\mathbf{h}) - \mathbf{k}\|_2^2 + \|\boldsymbol{\alpha} \cdot \mathbf{h}\|_2^2, \tag{21}$$

with the hope of getting more stable solution. Specially, here we use a vector of regularization parameters to balance the trade-off between the data-fidelity and the regularization term according to the x Position. Based on the Matlab's optimization decision table [6], there are only two solvers that we can apply to solve this bounded-smooth-nonlinear problem, i.e., `fmincon` and `fseminf`. In this study, we decided to use most popular `fmincon` solver, which uses *interior-point* (by default) algorithm internally. According to the Matlab® documentation, *interior-point* algorithm uses special techniques for large-scale problems and it satisfies bounds at all iterations.

In this experiment, we heuristically tuned the regularization parameters and recovered the bathymetry with only with 15% root mean square error (see Fig. 26) for the synthetic measurements with 25m grid resolution. Note that the recovered bathymetry is much stable and has less fluctuations than the ordinary least squares solution in Fig. 24.

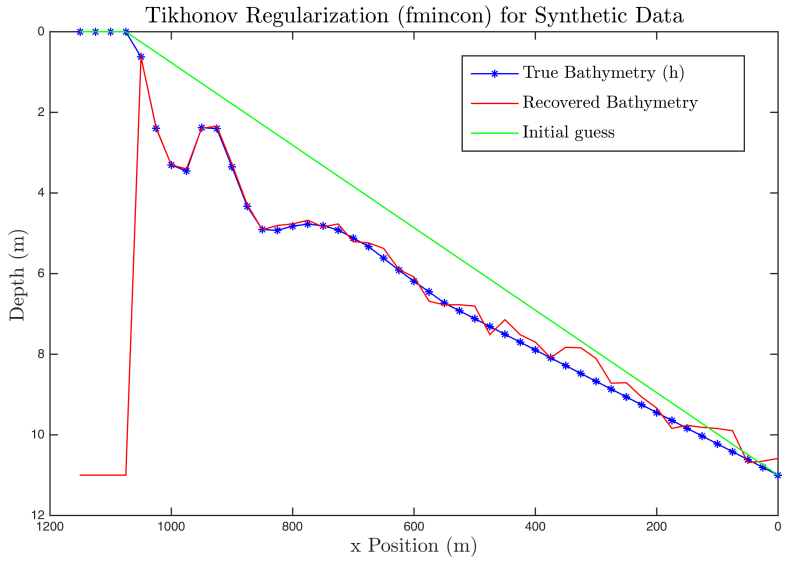

Figure 26: Tikhonov Regularized reconstruction of the depth **h** using the simulated data.

Similarly, we run the Matlab's `fmincon` solver to minimize the Tikhonov regularized objective with the real dataset $\mathbf{k}_s$. In this experiment, we were able to recover the sand bar area much accurately with less artifacts (see Fig. 27) but again deviations are noted sea-ward of the sand bar.

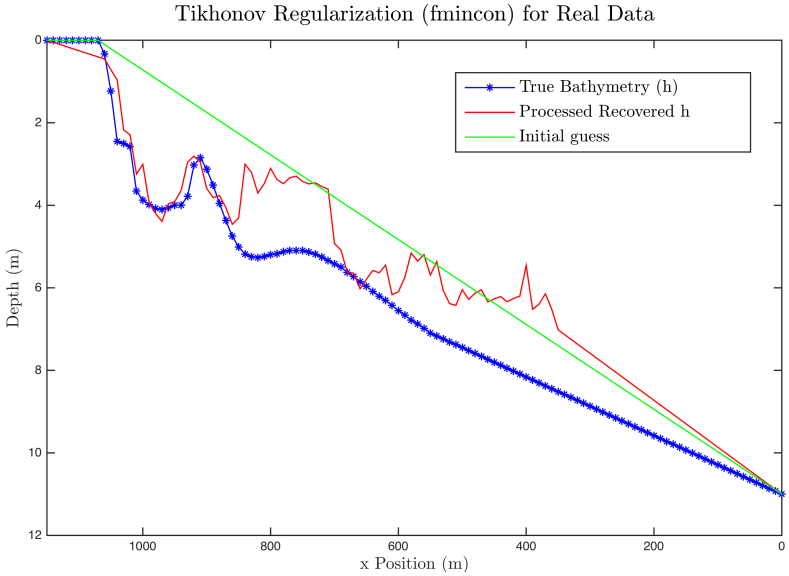

Figure 27: Tikhonov Regularized reconstruction of the depth **h** using the real data.

5.5 Bayesian Inference

The Bayesian inversion approach samples a posterior distribution of depth profiles. For comparison to the other inversion methods, we consider only the depth at each grid point which corresponds to the maximum of the posterior probability distribution at that point. This is

achieved by taking the maximum of the kernel density of the estimated depth distribution at each point along the 1D profile.

5.5.1 Manufactured Data Result

This method is first applied to the synthetic data $\mathbf{k}_s$ and the resulting depth estimate is shown in Figure 28. As for other methods, the synthetic result accurately represents the sandbar located at x position $\sim 950\ m$ along the profile, which is an important feature to recreate. However, at offshore locations (x position$< 600\ m$), the estimation appears to break down. As in other methods, this is expected because of the lower sensitivity of k on h at these depths, a relationship on which our inverse methods rely.

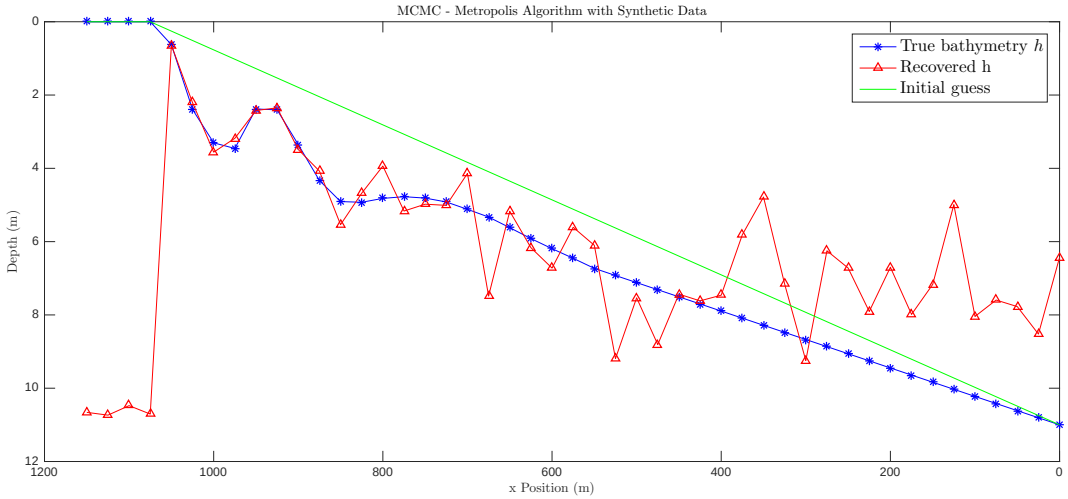

Figure 28: Bathymetry estimate from the Bayesian Markov Chain Monte Carlo (MCMC) approach using the synthetic wave numbers with 25m grid resolution. The initial depth guess is shown in green, the true depth is shown in blue, and the derived estimate of depth is shown in red.

The Markov Chain Monte Carlo (MCMC) results for the synthetic data $\mathbf{k}_s$ are generated using 20,000 iterations of the Metropolis algorithm after 1000 "burn in" iterations were discarded. The first iterations are considered to be a transient and unreliable portion of the solution so extra iterations, known as "burn in" iterations are included to account for this. Proposal h_{i+1} profiles are computed by adding a randomly generated step to h_i and a proposal posterior distribution is computed for the $(i+1)th$ step. An arbitrary scale factor of 0.5 is used to moderate the step size between iterations.

5.5.2 Real Data Result

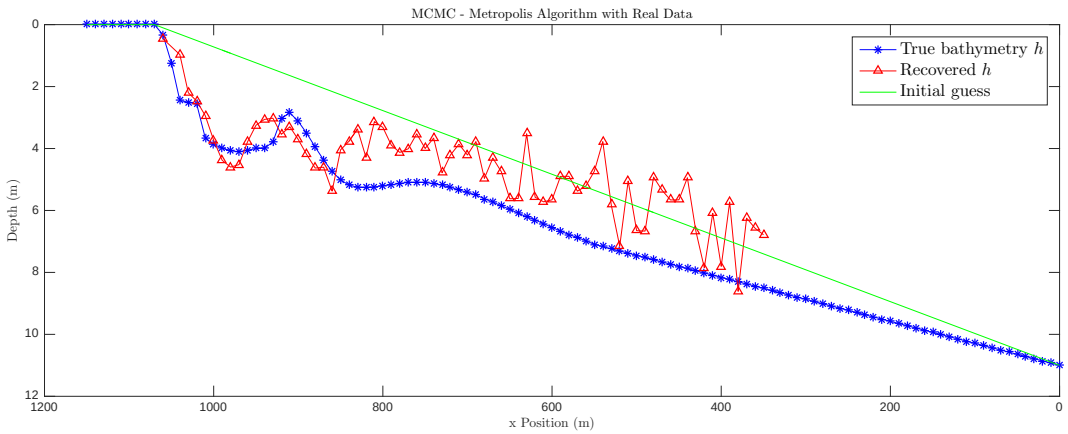

Figure 29: Bathymetry estimate from the Bayesian Markov Chain Monte Carlo (MCMC) approach using measured wave numbers as the objective in the likelihood function. The initial depth is shown in green, the true depth is shown in blue, and the derived estimate of depth is shown in red.

Real $\mathbf{k}_m$ data is used to estimate the same bathymetry profile (see Figure 29). As with the synthetic case, we found a good match to the sandbar region around x position$\sim$ 900 m. However, the estimated depth decreases just seaward of the sand bar and does not correlate well with the true bathymetry for the remainder of the seaward domain. We expect the noisy spike in $\mathbf{k}$ seaward of the sandbar seen in Figure 31 may be responsible for errors in our estimate of bathymetry in this area. The same issue occurs in the other inversion depth estimates. Note the real observed wave numbers are unavailable seaward of $\sim$ 350 m, so we do not estimate a depth for this region.

Overall, the MCMC result performs better for the synthetic case, as may be expected given increased noise in real observations of $\mathbf{k}_m$. We compute this depth estimate using 1000 iterations with 250 burn in iterations and a proposal step size of 0.1. It it possible that inclusion of the wave height, H, as an additional objective in the likelihood function would improve estimates seaward of the sandbar where wave number becomes less sensitive to depth, h, however we have yet to test this with the MCMC method. We also expect that additional scrutiny of our choices for the prior, proposal step size, and number of iterations could result in an improved bathymetry estimate given observed wave number, $\mathbf{k}_m$.

5.5.3 Posterior Probability Distributions

Unlike the other methods, the Bayesian approach results in a distribution of depth profiles which optimize bathymetry given uncertain wave number. Measurements of wave number, $\mathbf{k}_m$, are not perfect and our resulting distribution of depth profiles translates that uncertainty into an ensemble of depth estimates which are consistent with the $\mathbf{k}_m$ data to within uncertainty. Figures 30 and **??** show the resulting posterior depth distributions.

As expected, the posterior distribution shows confidence in the location and shape of the true bathymetry in the area of the sandbar. However, the posterior also seems to show confidence in the high variability area seaward of the sandbar. As discussed previously, we expect our estimates to perform poorly in the offshore region, which would suggest our posterior distribution demonstrate less confidence in this area. Further tests are needed to thoroughly examine this issue, but it is possible our Metropolis algorithm is heavily dependent on choices of the prior distribution and proposal function.

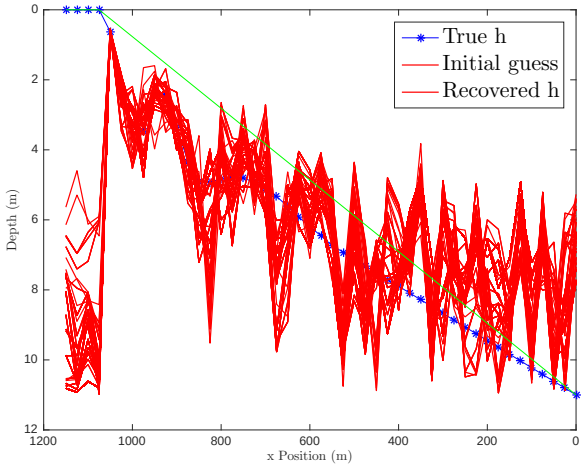

Figure 30: Posterior distribution of depth, **h**, estimates for the manufactured data result. The distribution largely dithers around the result shown in Figure 28. The posterior accurately predicts the location of the sandbar but appears to give false confidence to our bathymetry estimate elsewhere.

The posterior distribution resulting from use of real k values leaves the position of the sandbar ambiguous because the posterior distribution does not appear to converge to a single solution. This suggests less confidence in our estimate of the sandbar bathymetry, though the maximum of our posterior density appears to converge to the true sandbar location. The lack of confidence in the real versus manufactured data results may be due to discrepancies in the number of iterations for each. Because the real data result is derived from only 1000 iterations as opposed to 20,000 for the manufactured data result, it is possible the solution has not converged sufficiently to the true posterior disribution of depth, h. Increasing the number of iterations and burn in may therefore improve the posterior distribution and therefore confidence in our estimate of the true bathymetry.

5.6 Effect of Noise

As mentioned in Sections 5.3 - 5.5, all three inversion methods perform well around the sand bar at $x \approx 900$ m but problems are noted as one moves sea-ward. This deviation between the approximated solution and the true solution is attributed to noise in the measured wave number profile; to see why, consider the difference between the calculated wave profile and the measured profile (Figure 31). As can be seen, the calculated wave number exhibits a smooth

shape similar to the true bathymetry profile. Since we are using the same forward model to invert for the bathymetry we expect the calculated bathymetry to behave similarly to the measured wave number. At $x \approx 800$ m, Figure 31 shows a distinct jump in the measured wave number which accounts for the mirrored behaviour in the real data results for all three methods.

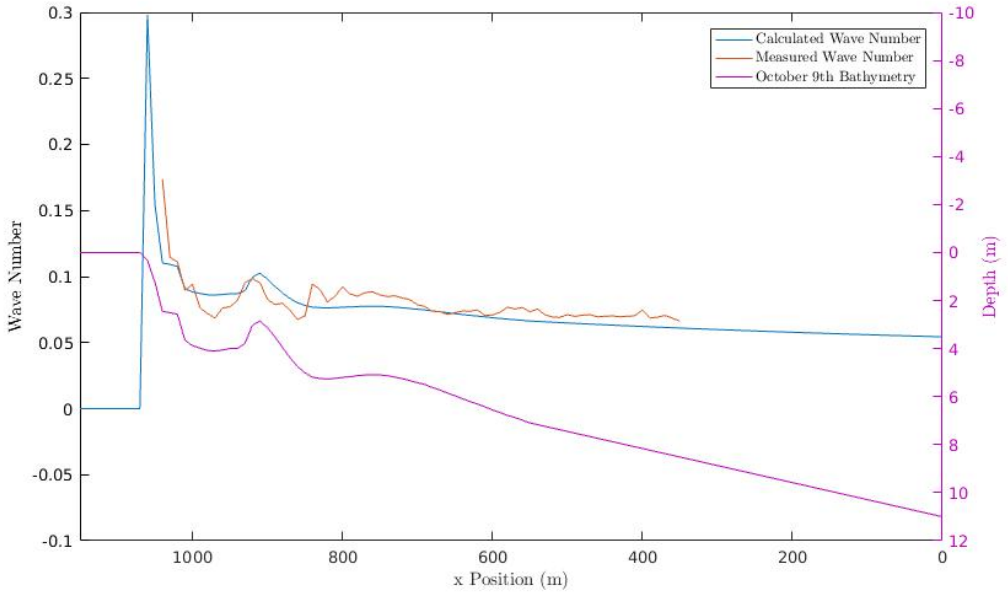

Figure 31: Noise in Measured Wave Number

5.7 Normalized Root Mean Square Error (NRMSE)

We evaluate the NRMSE of our methods to measure the statistical differences between the estimated depths and the depths which were actually observed on October 9th (see section 5.2). The error along our main interest, the sandbar, is relatively low due to there being less noise within the parameters of the sandbar compared to the error along the 1D transect.

Depth Prediction Error Along 1D Transect			
Method	**NRMSE Simulated k**	**Method**	**NRMSE Real k**
Nonlinear LS	53.81%	Nonlinear LS	53.89%
Tikhonov	46.40%	Tikhonov	15.70%
MCMC	50.35%	MCMC	29.58%

Depth Prediction Error Along Sandbar ($600m \leq x \leq 1000m$)			
Method	**NRMSE Simulated k**	**Method**	**NRMSE Real k**
Nonlinear LS	21.90%	Nonlinear LS	21.75%
Tikhonov	2.04%	Tikhonov	21.72%
MCMC	14.62%	MCMC	17.82%

Figure 32: Depth Prediction Error for Estimated Depths

6 Summary and Future Work

Historically, the standard approach for resolving coastal hydrodynamics has been to directly measure bathymetry and use the data as input in numerical models to determine coastal conditions. However, direct measurements of bathymetry are known to be costly in both a temporal and financial sense requiring heavy project hours and/or sophisticated equipment. In recent years, remote sensing and photogrammetry have seen rapid advancement, facilitating the easy measurement of surface properties of the ocean. This availability of data has motivated the scientific community to try to flip problem and estimate bathymetry using surface properties.

For this project we propose three different inversion methods for estimating bathymetry from wavenumber data; a standard Non-linear Least Squares approach, a Bayesian MCMC statistical method, and a Tikhonov Regularized Non-Linear Least Squares method. All three methods perform well in the dynamic portion of the near shore (the region of highest interest) and accurately predict the location of the sand bar at approximately $x = 900$ m. As we move further off shore we see notable deviations from the known bathymetry. This deviation has been attributed to the fact that as depth increases, wave numbers become less dependent on water depth due to limited interaction of surface waves with the sea bed. This phenomenon leads to an ill-conditioning of our forward operator and thus small deviations in the measured wavenumber lead to large fluctuations predicted bathymetry below. Of the three methods tested, the most promising to deal with this difficulty is the Tikhonov Regularization scheme as the use of a regularizer is well equipped to handle the instabilities introduced by this ill-conditioning, as shown in both the synthetic and real data tests. That being said, all three methods perform well in the region of highest interest and further work should be undertaken to test to suitability of each.

To expand on this work, there are many options for future work. In our formulation of the forward model, we assume the dispersion relationship given by linear wave theory but

formulations that involve wave height, such as those in [2], should be tested to assess the validity of the linear model in this context. Additionally, the inversion methods outlined in this report will allow for the use of other measured parameters, such as wave height or known bathymetry in the assimilation scheme. The use of multiple measurement types may reduce the sensitivity to noise in a single variable. A method of including a quantification of "noise" in these schemes should also be explored. In Section 5.4 we discuss the effect of noise in the wavenumber data on the estimated bathymetry. If an uncertainty measurement is applied to each wave number and we assume bathymetry is stable on timescales relative to the temporal spacing of wavenumber measurements, a method of applying multiple wavenumber profiles through time could be used to better estimate the bathymetry by applying additional weight to reliable data in each wavenumber profile. Another promising route for research would be the use of different regularization schemes for the Tikhonov method. Due to time constraints, our work was limited to heuristic testing of different different regularization parameters but as mentioned Section 4.3, a prior estimate of the bathymetry, such as a stable beach profile, could be implemented into the regularization. Lastly, we suggest the implementation of these schemes into higher fidelity models, such as a two dimensional models, to see how they perform in predicting a three dimensional representation of the bathymetry.

7 Acknowledgements

We acknowledge the generous support of SAMSI, NCSU, and the NSF in addition to the workshop organizers including Ilse Ipsen and Thomas Gehrmann. We also appreciate the guidance and patience of our mentors Ty Hesser and Matthew Farthing of the USACE and Lea Jenkins of Clemson University. Anusha Madushani (UC Merced) and Kimberly Kaufeld (NCSU) provided valuable input to the MCMC inversion effort. Finally, we thank the U.S. Army Corps of Engineers for providing a novel problem and interesting data from which to start.

References

[1] Alex Apotsos, Britt Raubenheimer, Steve Elgar, and R.T. Guza. Testing and calibrating parametric wave transformation models on natural beaches. *Coastal Engineering*, 55:224–235, 2007.

[2] Patricio A. Catalan and Merrick C. Haller. Remote sensing of breaking wave phase speeds application to non-linear depth inversions. *Coastal Engineering*, 55:93–111, 2008.

[3] Robert George Dean and Robert A Dalrymple. Water wave mechanics for engineers and scientists. 1991.

[4] Rob Holman, Nathaniel Plant, and Todd Holland. cbathy: A robust algorithm for estimating nearshore bathymetry. *Journal of Geophysical Research: Oceans*, 118(5):2595–2609, 2013.

[5] The MathWorks Inc. Least-squares (model fitting) algorithms, 2016.

[6] The MathWorks Inc. Optimization decision table, 2016.

[7] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.

[8] Hilary F Stockdon and Rob A Holman. Estimation of wave phase speed and nearshore bathymetry from video imagery. *Journal of Geophysical Research: Oceans*, 105(C9):22015–22033, 2000.

[9] A. Tuysuzoglu. Robust inversion and detection techniques for improved imaging performance. *Dissertation*, page 14, 2014.