Delft University of Technology

Wavefield Imaging
EE4595

WFI Final Assignment

Author:
Saurabh Sanjayrao Nerkar (5045274)
S.S.Nerkar@student.tudelft.nl

July, 2020

TU Delft
Delft University of Technology

Contents

1 Problem Statement **2**

2 Problem 1 **2**

3 Problem 2 **3**

4 Problem 3 **3**

5 Problem 4 **3**

6 Problem 5 **4**

7 Problem 6 **4**

8 Problem 7 **5**

9 Problem 8 **6**

10 Problem 9 **7**

11 Problem 10 **7**

12 Problem 11 **8**

13 Problem 12 **8**

14 Problem 13 **9**

15 Problem 14 **9**

16 Problem 15 **10**

17 Problem 16 **10**

18 Problem 17 **11**

19 Problem 18 **11**

20 Problem 19 **11**

21 Problem 20 **12**

References **13**

A Appendix: Matlab Scripts **14**

A.1 Main Script: 14
A.2 Function: Calculation of incident field 17
A.3 Function: Plotting of incident field 18

1 Problem Statement

In this assignment, we consider a two-dimensional linearized inverse scattering problem (inverse medium problem). The goal is to obtain an image of an object by solving the data equation with respect to the contrast function. First, we consider a two-dimensional homogeneous background configuration that is characterized by a constant background wave speed c_b.

The position vector in this 2D configuration is given by $\rho = x\hat{i}_x + y\hat{i}_y$ and a line source with unit amplitude is located at a point with position vector $\rho_s = x_s\hat{i}_x + y_s\hat{i}_y$. The field generated by this source satisfies the Helmholtz equation, $(\partial_x^2 + \partial_y^2 + \partial_z^2)\hat{u}^{inc} = -\delta(x - x_s)\delta(y - y_s)$ where $k_b = \omega/c_b$ is the wave number of the background medium and the delta functions on the right-hand side are Dirac distributions. Using an $exp(j\omega t)$ time convention, the incident field that radiates away from the line source is given by

$$\hat{u}^{inc}(\boldsymbol{\rho}, j\omega) = -\frac{j}{4}H_0^{(2)}(k_b(|\boldsymbol{\rho} - \boldsymbol{\rho_s}|)) \tag{1}$$

In this equation, $H_0^{(2)}$ is the so-called Hankel function of the second kind and order zero.

An object, occupying the bounded domain $\mathbb{D}$, is embedded in this background material and is characterized by the contrast function,

$$\chi(\boldsymbol{\rho}) = [\frac{k(\boldsymbol{\rho})}{k_b}]^2 - 1 \tag{2}$$

where $k(\boldsymbol{\rho}) = \omega/c(\boldsymbol{\rho})$ is the position-dependent wave number of the object. The Green's function for this 2D configuration has the same form as the background field, that is,

$$\hat{G}(\boldsymbol{\rho} - \boldsymbol{\rho'}, j\omega) = -\frac{j}{4}H_0^{(2)}(k_b(|\boldsymbol{\rho} - \boldsymbol{\rho'}|)) \tag{3}$$

and in the Born approximation, the scattered field at location $\boldsymbol{\rho} \notin \mathbb{D}$ is given by,

$$\hat{u}^{sc}(\boldsymbol{\rho}, j\omega) = k_b^2 \int_{\boldsymbol{\rho'}\in\mathbb{D}} \hat{G}(\boldsymbol{\rho} - \boldsymbol{\rho'}, j\omega)\chi(\boldsymbol{\rho'})\hat{u}^{inc}(\boldsymbol{\rho'}, j\omega)dV \tag{4}$$

The goal is to reconstruct $\chi(\boldsymbol{\rho})$ for $\boldsymbol{\rho} \in \mathbb{D}$ from knowledge of $\hat{u}^{sc}(\boldsymbol{\rho}, j\omega)$ at a number of receiver locations $\boldsymbol{\rho} = \boldsymbol{\rho}_m^R \notin \mathbb{D}$, m = 1, 2, ..., M.

2 Problem 1

First, find out what the Hankel function $H_0^{(2)}$ really is. Locate the corresponding function in Matlab and learn how it works.

Solution:

Hankel function is also called the Bessel function of the third kind. Hankel functions are the linear combinations of the two linearly independent solutions of the Bessel's differential equation.

$$H_0^{(1)} = J_\alpha + iY_\alpha \tag{5}$$

$$H_0^{(2)} = J_\alpha - iY_\alpha \tag{6}$$

Where J_α and Y_α are the first and second kind of Bessel functions respectively. The Green's function of the field can be found using the Hankel function. Hankel functions are used in calculation of inward and outward propagating cylindrical-wave solutions of the cylindrical wave equation.[1]

In Matlab the Hankel function can be calculated using the Matlab function 'besselh'[2]. 'H = besselh(nu,K,Z)' gives the Hankel function of 'nu' order and 'k' type of variable 'Z'. 'k' is either 1 or 2.

3 Problem 2

Substitute Eqs. (1) and (2) in Eq. (3) and write down the resulting integral representation explicitly.

Solution:

Equation (3) is given as,

$$\hat{u}^{sc}(\boldsymbol{\rho}, j\omega) = k_b^2 \int_{\boldsymbol{\rho'} \in \mathbb{D}} \hat{G}(\boldsymbol{\rho} - \boldsymbol{\rho'}, j\omega)\chi(\boldsymbol{\rho'})\hat{u}^{inc}(\boldsymbol{\rho'}, j\omega)dV \tag{7}$$

Substituting eq (1) and eq (2) in eq (3) we get,

$$\hat{u}^{sc}(\boldsymbol{\rho}, j\omega) = -\frac{k_b^2}{16} \int_{\boldsymbol{\rho'} \in \mathbb{D}} H_0^{(2)}(k_b(|\boldsymbol{\rho} - \boldsymbol{\rho'}|))H_0^{(2)}(k_b(|\boldsymbol{\rho'} - \boldsymbol{\rho_s}|))\chi(\boldsymbol{\rho'})dV \tag{8}$$

Here, $\boldsymbol{\rho'} \in \mathbb{D}$, $\hat{u}^{sc}(\boldsymbol{\rho}, j\omega)$ is known at a number of receiver locations $\boldsymbol{\rho} = \boldsymbol{\rho}_m^R \notin \mathbb{D}$, m = 1, 2, ..., M.

4 Problem 3

For simplicity, take $k_b = 1$ and introduce a Cartesian coordinate system with x-axis horizontal and pointing to the right, y-axis vertical and pointing down. Take the object domain $\mathbb{D}$ to be a rectangle such that its upper left corner is at the origin of the coordinate system and its lower right corner is located at the point with coordinates (λ, λ), where $\lambda = 2\pi/k_b$ is the wavelength of the background field. The source is located at the point $\rho_s = (\lambda/2)\hat{i}_x + 10\lambda\hat{i}_y$. Make a sketch of the configuration (this will be Figure 1 in your report). Consider using the line command of Matlab for this sketch.

Solution:

Given rectangular domain was drawn using line command[3]. The domain and the source at $\rho_s = (\lambda/2)\hat{i}_x + 10\lambda\hat{i}_y$, is located in figure 1 below.

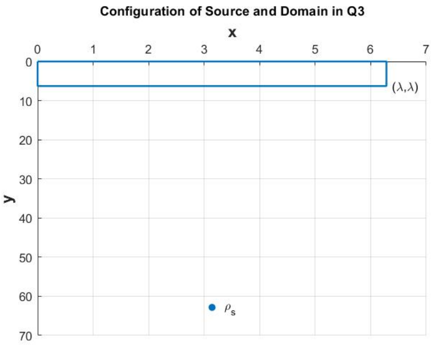

Figure 1: Configuration of the source and the object domain in given question

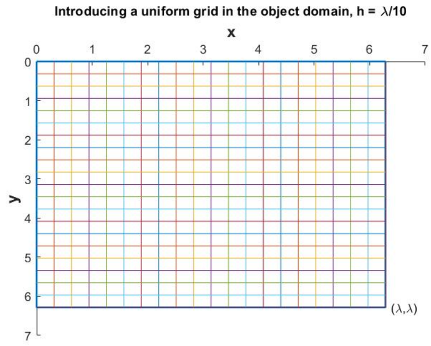

Figure 2: Object Domain after introduction of uniform grid of step size $h = \lambda/20$

5 Problem 4

Create a Matlab script file to store your work.

Solution:

Matlab script of the given assignment is showcased in appendix 1.

6 Problem 5

Introduce a uniform grid (same increment in the x- and y-directions) on $\mathbb{D}$ with a step size $h = \lambda/20$. Compute the total number of grid points N.

Solution:

A uniform grid was introduced in the object domain with a step size of $h = \lambda/20$. Figure 2 shows the required object domain after introduction of the uniform grid.

Total number of grid points were found to be $N = 441$.

7 Problem 6

Compute the incident field on the grid and store it as a 2D array in Matlab. Make a 2D image of the real and imaginary parts and the absolute value of $\hat{u}^{inc}$. Make sure that the orientation of the image corresponds to the orientation of your coordinate system. Use axis equal tight command to keep the aspect ratio.

Solution:

Figures 3, 4 and 5 give the required images of the absolute value of the incident field on the grid along with it's real and imaginary parts respectively. Command 'imagesc'[4] in Matlab was used to get the images.

It can be seen clearly from the images that the field is stronger near the source than away from the source. The colorbar in the figures indicate the strength of the respective quantities in the given images.

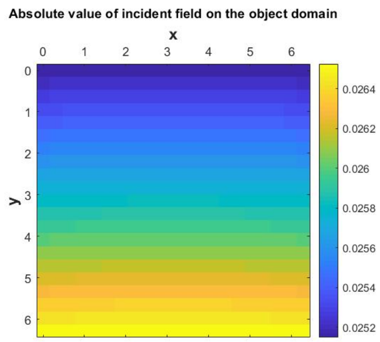

Figure 3: Absolute value of the incident field on the object domain

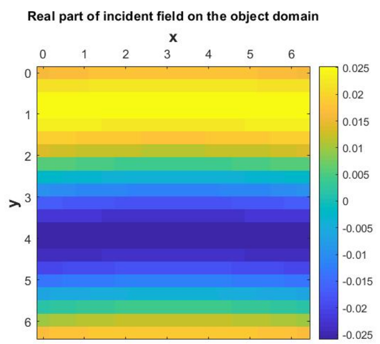

Figure 4: Real part of the incident field on the object domain

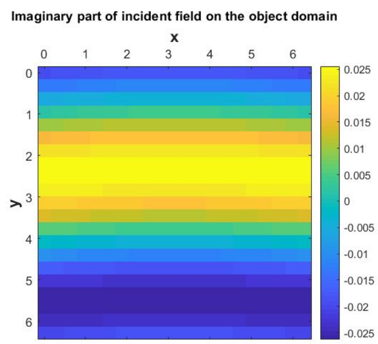

Figure 5: Imaginary part of the incident field on the object domain

8 Problem 7

Change the location of the source (move it closer to the object domain $\mathbb{D}$, for example) and repeat the calculations. Does the background field distribution change? Change the wave number to $k_b = 2$ and repeat the calculations. Explain the changes in the background field. When you are done, return to the case where the source is located at $\rho_s = (\lambda/2)\hat{i}_x + 10\lambda\hat{i}_y$ and $k_b = 1$.

Solution:

In this question, first the location of the source was changed and it was made far from the object domain ($\rho_{snew} = (\lambda/2)\hat{i}_x + 20\lambda\hat{i}_y$). Figures 6, 7 and 8 represent the incident field on the object domain from the source which is farther than in the previous question. It is known than the intensity of the field is inversely proportional to the distance square and this can be validated from the aforementioned figures. The field indeed decreases when the source is away from its actual location.

Afterwards, the source was moved nearer to the object domain ($\rho_{snew} = (\lambda/2)\hat{i}_x + 4\lambda\hat{i}_y$) than it's actual position and as it was expected the field strength increased at the nearer location. Figures 9, 10 and 11 represent the absolute value, the real part and the imaginary part of the incident field on the object domain when the source is nearer than it's actual location.

The third variation that was applied was changing the value of the wavenumber k_b to be equal to 2 from its previous value 1. It was seen that the value of the field decreases in this scenario when compared to the case where, $k_b = 1$. Figures 12, 13 and 14 represent absolute value, the real part and the imaginary part of the incident field on the object domain in this scenario where, k_b is changed to 2 from 1.

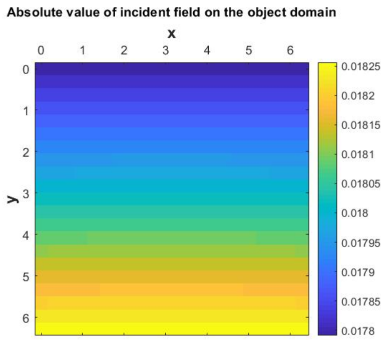

Figure 6: Absolute value of the incident field on the object domain (Far Source)

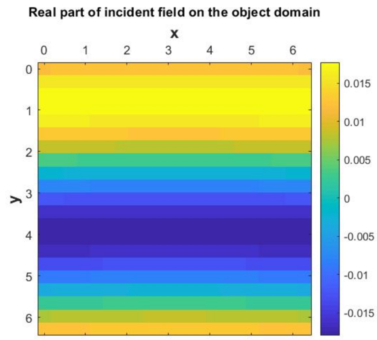

Figure 7: Real part of the incident field on the object domain (Far Source)

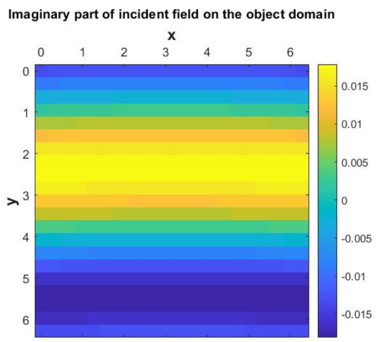

Figure 8: Imaginary part of the incident field on the object domain (Far Source)

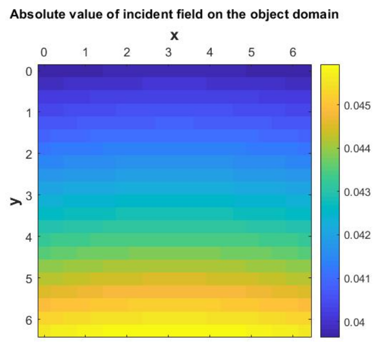

Figure 9: Absolute value of the incident field on the object domain (Near Source)

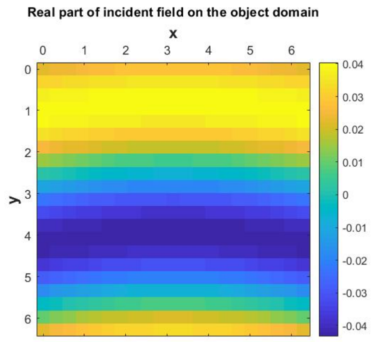

Figure 10: Real part of the incident field on the object domain (Near Source)

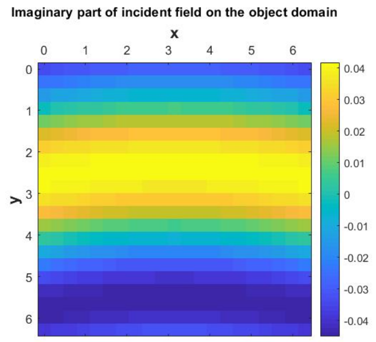

Figure 11: Imaginary part of the incident field on the object domain (Near Source)

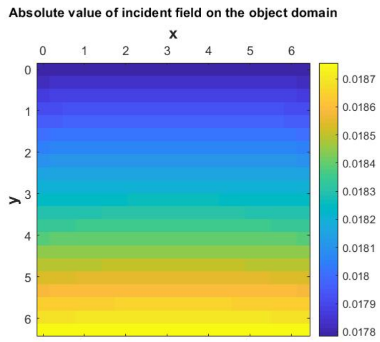

Figure 12: Absolute value of the incident field on the object domain ($k_b = 2$)

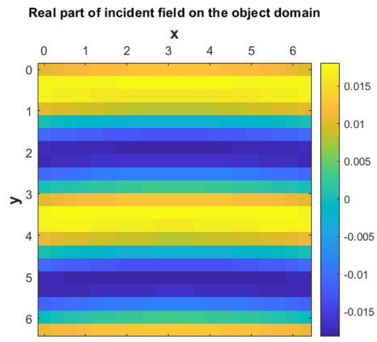

Figure 13: Real part of the incident field on the object domain ($k_b = 2$)

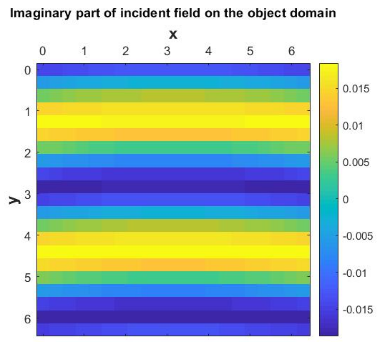

Figure 14: Imaginary part of the incident field on the object domain ($k_b = 2$)

9 Problem 8

Introduce a contrast function $\chi(\boldsymbol{\rho}) \geq 0$ of your object (your choice!) with $\boldsymbol{\rho}$ on the grid and store it as a 2D array in Matlab.

Solution:

A contrast function of a circular (disc) object placed at $(\lambda/2, \lambda/2)$ was found. The radius of the object was defined to be $\lambda/6$. The values of $k(\rho)$ were defined to achieve $\chi(\rho) \geq 0$. Section-Q8 in the main code mentioned in the appendix-1 has the required Matlab code. Figure 15 shows the image of the defined contrast function.

10 Problem 9

Make a 2D image of your contrast using the imagesc command. Add a colorbar to it.

Solution:

Figure 15 shows the image of the contrast function defined in question 8. One thing to notice here is that, even though the expression used was of a circular object, the object that we see in the figure is different from a circular object is because the pixels in the image are larger in size (less resolution).

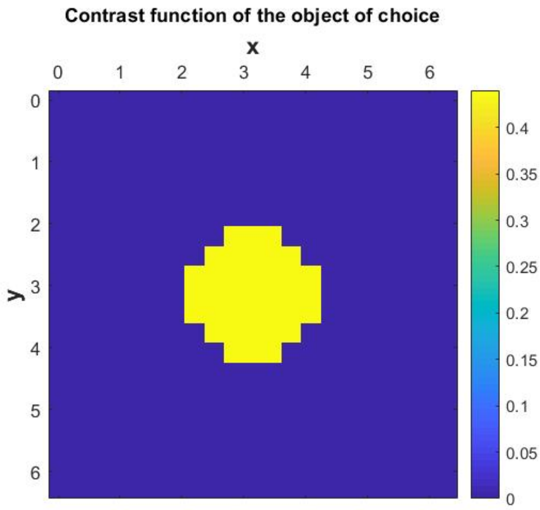

Figure 15: Image of the contrast function of the choice

11 Problem 10

Define the receiver domain $\mathbb{D}^{rec}$ as a line segment of length $L = 3\lambda$ with end points at $(-\lambda, 1.5\lambda)$ and $(2\lambda, 1.5\lambda)$. Draw it in Figure 1.

Solution:

The receiver domain $\mathbb{D}^{rec}$ was defined. $\mathbb{D}^{rec}$ is a line segment of length $L = 3\lambda$ with end points at $(-\lambda, 1.5\lambda)$ and $(2\lambda, 1.5\lambda)$.

Figure 16 represents the receiver domain drawn along with the object domain and the source.

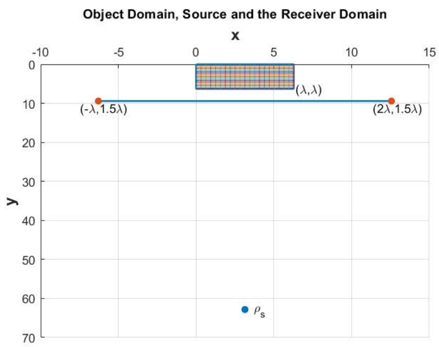

Figure 16: Object Domain, Receiver Domain and the Source

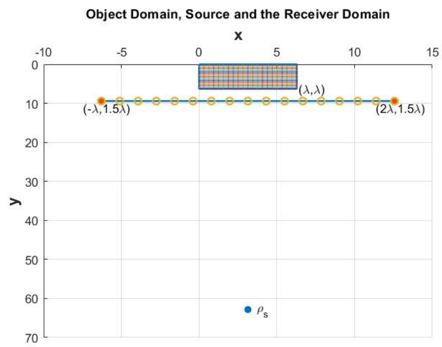

Figure 17: Object Domain, Receiver Domain (with grid M = 17) and the Source

12 Problem 11

Introduce a grid on $\mathbb{D}^{rec}$ using a uniform step size so that there are M uniformly spaced grid points on the line segment, where $M > 0$ is an integer parameter of your problem that you will investigate later.

Solution:

A uniform grid with (M = 17, M uniformly spaced grid points on the line segment) was introduced on the receiver domain $\mathbb{D}^{rec}$. Figure 17 represents the object domain, source and the receiver domain with the uniform grid (M = 17).

13 Problem 12

Discretize the data equation (3) by approximating the integral via a sum (use the composite midpoint rule, for example). Write down the discretized equation

Solution:

Equation (3) is given as,

$$\hat{u}^{sc}(\boldsymbol{\rho}, j\omega) = k_b^2 \int_{\boldsymbol{\rho'}\in\mathbb{D}} \hat{G}(\boldsymbol{\rho} - \boldsymbol{\rho'}, j\omega)\chi(\boldsymbol{\rho'})\hat{u}^{inc}(\boldsymbol{\rho'}, j\omega)dV \tag{9}$$

Substituting eq (1) and eq (2) in eq (3) we get following equation, which we have already derived in question 2.

$$\hat{u}^{sc}(\boldsymbol{\rho}, j\omega) = -\frac{k_b^2}{16} \int_{\boldsymbol{\rho'}\in\mathbb{D}} H_0^{(2)}(k_b(|\boldsymbol{\rho} - \boldsymbol{\rho'}|))H_0^{(2)}(k_b(|\boldsymbol{\rho'} - \boldsymbol{\rho_s}|))\chi(\boldsymbol{\rho'})dV \tag{10}$$

Here, $\boldsymbol{\rho'} \in \mathbb{D}$, $\hat{u}^{sc}(\boldsymbol{\rho}, j\omega)$ is known at a number of receiver locations $\boldsymbol{\rho} = \boldsymbol{\rho}_m^R \notin \mathbb{D}$, m = 1, 2, ..., M.

The object domain in the question is a 2D space. Hence, the discretization will happen in 2D[5]. Equation (10) can be re-written as below:

$$\hat{u}^{sc}(x_m, y_m, j\omega) = -\frac{k_b^2}{16} \int_{x'=0}^{x'=\lambda} \int_{y'=0}^{y'=\lambda} H_0^{(2)}(k_b * f(x_m, y_m, x', y'))H_0^{(2)}(k_b * f(x', y', x_s, y_s))\chi(x', y')dx'dy' \tag{11}$$

Here (in equation 11), $x', y' \in \mathbb{D}$, $x_m, y_m \in \mathbb{D}^{rec}$ and x_s, y_s is the location of the source.

Also the general definition of $f(x, y, x', y')$ is given as below:

$$f(x, y, x', y') = \sqrt{|(x - x')|^2 + |y - y'|^2} \tag{12}$$

Equation (11) can then be discretized as below: (Using composite mid-point rule)

$$\hat{u}^{sc}(x_m, y_m, j\omega) = -\frac{k_b^2}{16}\Delta x\Delta y \sum_{i=1}^{i=n_x} \sum_{j=0}^{j=n_y} H_0^{(2)}(k_b * f(x_m, y_m, x_i, y_j))H_0^{(2)}(k_b * f(x_i, y_j, x_s, y_s))\chi(x_i, y_j) \tag{13}$$

Here, $n_x = \lambda/\Delta x$ and $n_y = \lambda/\Delta y$, $\Delta x = \Delta y = \lambda/20$, and $x_i = (i - 1/2) * \Delta x$ and $y_j = (j - 1/2) * \Delta y$. i is varying from 1 to n_x and j from 1 to n_y.

14 Problem 13

Study the reshape command of Matlab. Reshape the 2D array of the contrast function into a column vector $x \in \mathbb{C}^{N\times 1}$.

Solution:

Reshape command in Matlab 'B = reshape(A, sz)' reshapes a matrix A into B according to the size vector sz[6]. The total number of elements before and after the reshape should remain the same.

Q13 section in the main code in appendix-1 reshapes the 2D array of the contrast function into a column vector. Here N = 441 as found in Q5.

```
%Reshaped contrast function
siRs = reshape(si_rho, [N, 1]);
```

15 Problem 14

Build the system matrix $A \in \mathbb{C}^{M\times N}$ of your discretized data equation.

Solution:

The data equation equation (3) or (8) was discretized in equation (14). This discretized equation represents a system of linear equations of form,

$$B = Ax \tag{14}$$

Comparing equation (15) with (14) it can be found that, in this case, $\chi(x_k, y_k)$ is the x, $\chi(x_k, y_k)$ is reshaped to be $x \in \mathbb{C}^{N\times 1}$. $\hat{u}^{sc}(x_m, y_m, j\omega)$ is the B. Similarly, the system matrix A is found to be as follows:

$$A = -\frac{k_b^2}{16}\Delta x \Delta y \sum_{i=1}^{i=n_x}\sum_{j=0}^{j=n_y} H_0^{(2)}(k_b * f(x_m, y_m, x_i, y_j))H_0^{(2)}(k_b * f(x_i, y_j, x_s, y_s)) \tag{15}$$

Here, $f(x, y, x', y')$ is given by equation (12).

$$\hat{u}^{sc} = [\hat{u}^{sc}(x_1, y_1, j\omega), \hat{u}^{sc}(x_2, y_2, j\omega)...\hat{u}^{sc}(x_M, y_M, j\omega)] \tag{16}$$

$\hat{u}^{sc} \in \mathbb{C}^{M\times 1}$ and the system matrix $A \in \mathbb{C}^{M\times N}$.

Section-Q14 in the Matlab main code given in appendix-1 builds the system matrix using the discretized data equation.

16 Problem 15

Compute the singular value decomposition of the system matrix using the svd command. Plot the singular values. Change M and investigate the dependence of the singular values on M.

Solution:

The singular values of the system matrix A were computer using the svd[7] command. See, section Q15 in the main code given in appendix-1 for the implementation.

It was observed that the as M increases, the number of singular values above zero along with the zero singular values increase. This is a trade off, more non-zero singular values the better the reconstructed image is going to be. However, the zero singular values contribute to amplifying the noise. So, more number of zero singular values the reconstructed image (case: with noise) will be more distorted.[5]

Figure 18, 19 and 20 show the graphs of singular values when M = 3, M = 8 and M = 16 respectively. It can be seen that, M = 16 has the highest number of zero and non-zero singular values. In later questions, it can be seen that, M = 16 has the best reconstruction of the image when compared to the other two.

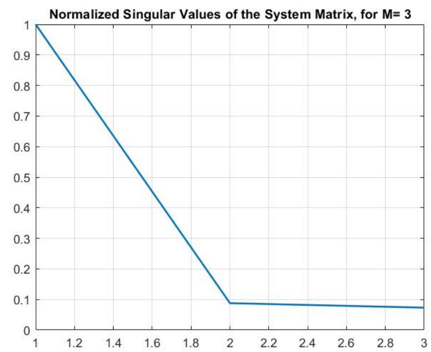

Figure 18: Singular values of system matrix when M = 3

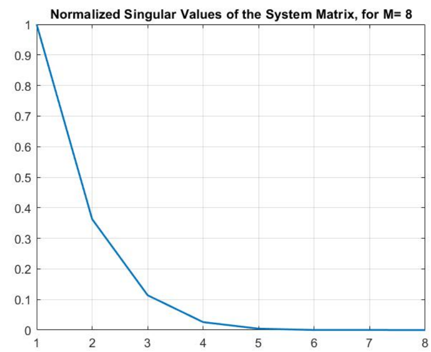

Figure 19: Singular values of system matrix when M = 8

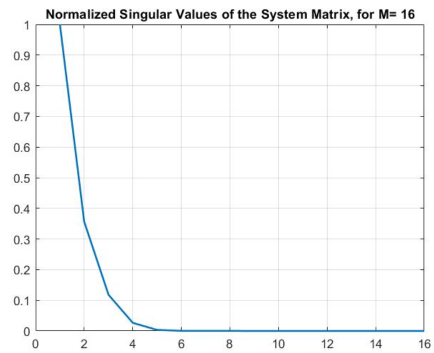

Figure 20: Singular values of system matrix when M = 16

17 Problem 16

Compute the scattered field as $u^{sc} = Ax$ in Matlab.

Solution:

```
usc = A*siRs;
```

Above code snippet was implemented in section Q16 of the main Matlab code that is given in the appendix-1. $\hat{u}^{sc} \in \mathbb{C}^{M\times 1}$ as discussed in question 14.

18 Problem 17

Consider now u^{sc} as given and determine the minimum norm solution x_{mn} using the singular value decomposition or the pinv command of Matlab

Solution:

```
%Using pinv
xmn = pinv(A)*usc;

%Using SVD (min norm calculation)
xmn1 = lsqminnorm(A,usc);
```

Above code snippet mentioned in section Q17 in the main Matlab code (appendix-1) calculates the minimum norm solution x_{mn} using singular value decomposition and pinv command[8][9].

19 Problem 18

Reshape the solution vector x_{mn} into a 2D array and make an image with imagesc. Compare this image with the image of the original contrast.

Solution:

Figure 21 shows the reconstructed image found after solving the data equation assuming given $\hat{u}^{sc}$. Here, M = 16. Figure 22 is the original contrast function shown again for comparison.

As it can be seen here, the reconstructed image is not exactly same as that of the contrast function however, it is similar. This gives an idea how the imaging happens through inverse scattering problem and what ambiguities could appear.

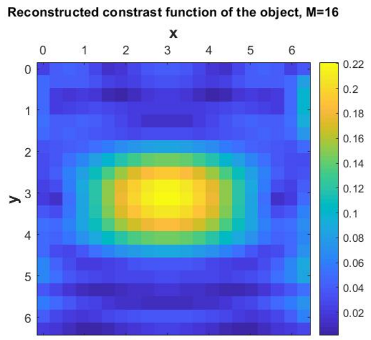

Figure 21: Reconstructed contrast function with M = 16

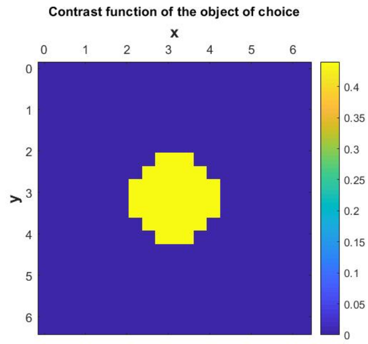

Figure 22: Actual contrast function of the object

20 Problem 19

Repeat the previous step with a different number of receivers M. Make an image.

Solution:

It was previously mentioned question 15 that changing M has an impact on the reconstruction of the image. M is a trade-off parameter, it improves the reconstruction of the image, at the same time, it can also amplify the noise (if any noise is present).

If the inter-receiver distance is greater than $\lambda/2$, the images (of contrast function) reconstructed are distorted enough that the position of object is difficult to identify. As M increases, the position gets clearer.

For M = 6, the contrast function is decently reconstructed without noise and has a better reconstruction with noise than the cases with higher or lower values of M.[5]

Figure 23, 24, 25 and 26 show the reconstructed contrast function when M = 3, 6, 8 and 16 respectively. (This case has no noise in $\hat{u}^{sc}$.)

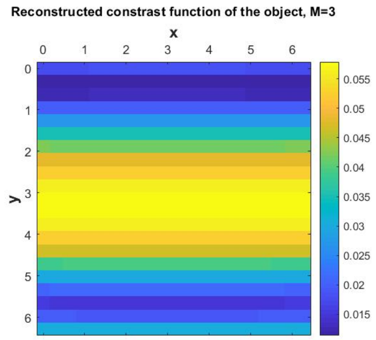

Figure 23: Reconstructed contrast function with M = 3

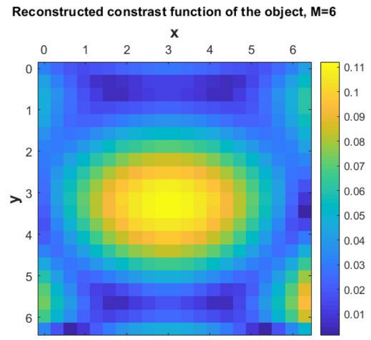

Figure 24: Reconstructed contrast function with M = 6

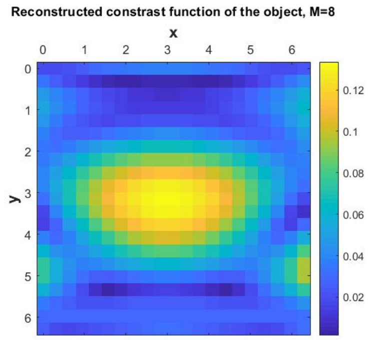

Figure 25: Reconstructed contrast function with M = 8

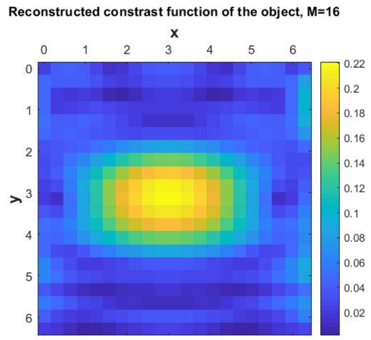

Figure 26: Reconstructed contrast function with M = 16

21 Problem 20

Add noise to the data vector u^{sc} using the rand command. Investigate the influence of noise on the reconstruction results.

Solution:

Noise amplitude of $u^{sc} * 0.01$ was taken and then using rand[10] command the noise was generated and added to the signal u^{sc}. The detailed code can be found under section Q20 of the main Matlab code mentioned in the appendix-1.

As discussed earlier, M is a trade-off parameter, larger M reconstructs the contrast function better than the smaller M if there is no noise. However, with larger M, the number of zero singular values increases and this leads to amplification of noise and that causes an extremely distorted reconstruction of the contrast function.

Figure 27 and 28 represent the reconstruction of the contrast function when the u^{sc} is noisy for M = 6 and M = 16 respectively.

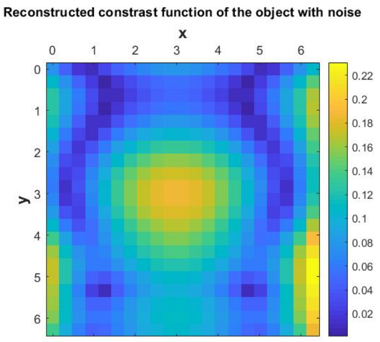

Figure 27: Reconstructed contrast function with M = 6 and noisy $\hat{u}^{sc}$

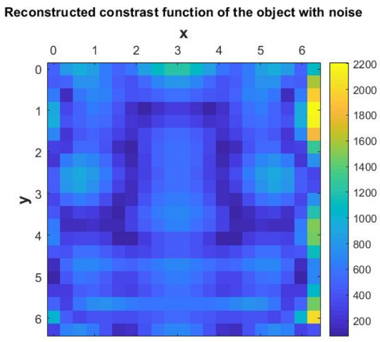

Figure 28: Reconstructed contrast function with M = 16 and noisy $\hat{u}^{sc}$

References

[1] (2020, Jun) Bessel function. Accessed: 2020-07-18. [Online]. Available: https://en.wikipedia.org/wiki/Bessel_function

[2] besselh. Accessed: 2020-07-18. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/besselh.html?s_tid=srchtitle

[3] line. Accessed: 2020-07-18. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/line.html

[4] imagesc. Accessed: 2020-07-18. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/imagesc.html

[5] R. Remis, "Wavefield imaging lecture notes."

[6] reshape. Accessed: 2020-07-19. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/reshape.html

[7] svd. Accessed: 2020-07-19. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/double.svd.html

[8] pinv. Accessed: 2020-07-19. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/pinv.html?s_tid=srchtitle

[9] lsqminnorm. Accessed: 2020-07-19. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/lsqminnorm.html

[10] rand. Accessed: 2020-07-19. [Online]. Available: https://nl.mathworks.com/help/matlab/ref/rand.html?s_tid=srchtitle

A Appendix: Matlab Scripts

This appendix contains the Matlab scripts written to complete the given assignment.

A.1 Main Script:

```
%Final project WFI
clear;
close all;

%% Q1 and Q2 Learning about Hankel function and deriving usc
%Hankel is a third order Bessel Function given by hankle command in
%Matlab. Second question is done by substituting eq(1) and (2) in (3).

%Taken from Matlab Documentation
k = 2;
nu = 0;
z = linspace(0.1,25,300);
H = besselh(nu,k,z);
figure(1);
plot(z,real(H),z,imag(H))
grid on
hold on
M = sqrt(real(H).^2 + imag(H).^2);
plot(z,M,'--')
legend('$J_0(z)$', '$Y_0(z)$', '$\sqrt{J_0^2 (z) + Y_0^2 (z)}$','interpreter','latex');
hold off;

%% Q3 Sketching the configuration

kb = 1;
lam = 2*pi/kb;
src = [lam/2, 10*lam];
domX = [0, 0, lam, lam, 0];
domY = [0, lam, lam, 0, 0];
figure(2);
line(domX, domY, 'LineWidth', 1.5); text(lam+0.1,lam,'(\lambda,\lambda)')
hold on;
scatter(src(1), src(2), 'filled'); text(lam/2+0.1,10*lam,'  \rho_s');
set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
grid on;
title('Configuration of Source and Domain in Q3');
xlabel('x','FontSize',13,'FontWeight','bold');
ylabel('y','FontSize',13,'FontWeight','bold');

%% Q5 Defining grid and the number of grid points
%Introducing grid
step = lam/20;
x_vec = 0:step:lam;
y_vec = 0:step:lam;
[x, y] = meshgrid(x_vec, y_vec);

%Numer of points
N = length(x).*length(y);

%Plotting grid
figure(2);
line(domX, domY, 'LineWidth', 1.5); text(lam+0.1,lam,'(\lambda,\lambda)')
line(x, y);
line(y, x);
set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
```

```
title('Introducing a uniform grid in the object domain, h = \lambda/10');
xlabel('x','FontSize',13,'FontWeight','bold');
ylabel('y','FontSize',13,'FontWeight','bold');

%% Q6 Calculation of incident field on the grid
uincInit = calcUinc(x, y, src, kb);
createImage(x_vec, y_vec, uincInit);

%% Q7 Variation in background field due to change in the location of the src and kb
%Changed position of source and changed kb
srcFar = [lam/2, 20*lam];
srcNear = [lam/2, 4*lam];
kbNew = 2;

%Field calculations - Far
% uincFar = calcUinc(x, y, srcFar, kb);
% createImage(x_vec, y_vec, uincFar);

%Field calculations - Near
% uincNear = calcUinc(x, y, srcNear, kb);
% createImage(x_vec, y_vec, uincNear);

% %Field calculations - Changed kb
uincKb = calcUinc(x, y, src, kbNew);
createImage(x_vec, y_vec, uincKb);

%% Q8 Introducting a contrast function of the object of my choice:
% I chose circle!! So, x^2 + y^2 = 0; Disc shape, however, due to the fact
% that the number of points in the grid are comparitively very few the
% circle won't be of a perfect shape.

center = [lam/2, lam/2];
expression = (x-center(1)).^2 + (y-center(2)).^2-(lam/6).^2;
%To make si >= 0
k_rho = ones(size(x)).*kb;
k_rho(expression <=0) = (1.2).*kb;
si_rho = (k_rho./kb).^2 - 1;

%% Q9 Making an image of the contrast function

figure(3);
imagesc(x_vec, y_vec, si_rho);
set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
colorbar;
axis equal tight;
title('Contrast function of the object of choice');
xlabel('x','FontSize',13,'FontWeight','bold');
ylabel('y','FontSize',13,'FontWeight','bold');

%% Q10 Defining the receiver domain L = 3*lam from (-lam, 1.5lam) to (2lam, 1.5lam)
%Adding the Rec domain to figure 1
drecX = [-lam 2*lam];
drecY = [1.5*lam 1.5*lam];

figure(2);
line(x, y);
line(y, x);
line(drecX, drecY, 'LineWidth', 1.5); hold on;
scatter(drecX, drecY, 'filled');
text(-lam-1.2,1.5*lam+1.8,'(-\lambda,1.5\lambda)');
text(2*lam-1.2,1.5*lam+1.8,'(2\lambda,1.5\lambda)');
```

```
title('Object Domain, Source and the Receiver Domain');

%% Q11 Introducing grid on Drec M > 0

Mdiv = 15; %Number of receivers
M = Mdiv + 1;
DrecX = -lam:3*lam/Mdiv:2*lam;
DrecY = 1.5*lam.*ones(size(DrecX));

figure(2);
scatter(DrecX, DrecY, 'LineWidth', 1.5);

%% Q12 Discretize data equation (appox. integral via sum)
%Mentioned in the report under question 12.

%% Q13 Reshape 2D array of contrast function; study reshape function

%Reshaped contrast function
siRs = reshape(si_rho, [N, 1]);

%% Q14 Building System Matrix A from discretized data equation
%Step size is same for x and y
delX = step;
delY = step;

%Covering the domain
nX = length(x_vec);
nY = length(y_vec);

%Defing the kth x
kX = 1:nX;
kY = 1:nY;
xK = (kX-1/2).*delX;
yK = (kY-1/2).*delY;

%Mid point meshgrid
[xK1, yK1] = meshgrid(xK, yK);

%Constant in front of sum in the discretized equation
const = -((kb^2)/16)*delX*delY;

%Finding system matrix
A = zeros(M, N);
for ind = 1:M
    xm = DrecX(ind);
    ym = DrecY(ind);
    rhoMS = sqrt(abs(xm-xK1)^2 + abs(ym-yK1)^2);
    Greq = besselh(nu, k, (kb.*rhoMS));
    Greq = Greq*(calcUinc(xK1, yK1, src, kb).*(-4./1j));
    Greq = reshape(Greq, [1, N]);
    A(ind, :) = const.*Greq;
end

%% Q15 Compute SVD of system matrix
%plot singular values, change M to invesigate dependance of singular values
%on M

%SVD
s = svd(A);
figure(4);
```

```
plot((s./max(s)), 'LineWidth', 1.5);
title(['Normalized Singular Values of the System Matrix, for M= ', num2str(M)]);
grid on;
%s > 0 -> eigen values are non-negative -> solutions available? Read more
%on this..

%% Q16 Computing scattered field

usc = A*siRs;

%% Q17 Considering usc given and calculating min non using pinv of system matrix

%Using pinv
xmn = pinv(A)*usc;

%Using SVD (min norm calculation)
xmn1 = lsqminnorm(A,usc);

%% Q18 Reshaping the solution in 2D

%Found si -> from SVD and given usc
xF = reshape(xmn, [length(x_vec), length(y_vec)]);
figure(5);
imagesc(x_vec, y_vec, abs(xF));
set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
colorbar;
axis equal tight;
title(['Reconstructed constrast function of the object, M=', num2str(M)]);
xlabel('x','FontSize',13,'FontWeight','bold');
ylabel('y','FontSize',13,'FontWeight','bold');

%% Q19 Repeating step 18 with different number of recivers and making images

%% Q20 Add noise to usc using rand and investigating influence of noise on
%reconstruction of results

%Add noise to the usc
noiseAmp = 0.01;
uscn = usc + (((usc).*noiseAmp).*rand(length(usc), 1));

%Using pinv -> is it min norm?
xmnnoise = pinv(A)*uscn;

%Reconstruction with noise
xFn = reshape(xmnnoise, [length(x_vec), length(y_vec)]);
figure(7);
imagesc(x_vec, y_vec, abs(xFn));
set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
colorbar;
axis equal tight;
title('Reconstructed constrast function of the object with noise');
xlabel('x','FontSize',13,'FontWeight','bold');
ylabel('y','FontSize',13,'FontWeight','bold');
```

A.2 Function: Calculation of incident field

```
%Function to calculate the incident field x and y are meshgrid variables
%and src is the location of the src, kb is wavenumber of the background
%medium.
function uinc = calcUinc(x, y, src, kb)
    nu = 0;
```

```
    k = 2;
    x_dash = x-src(1);
    y_dash = y-src(2);
    rho = sqrt(abs(x_dash).^2 + abs(y_dash).^2);
    uinc = (-1j/4).*besselh(nu,k,kb.*rho);
end
```

A.3 Function: Plotting of incident field

```
%Function to create images of fields using vectors and fields
function createImage(x, y, uinc)
    figure();
    imagesc(x, y, real(uinc));
    set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
    colorbar;
    axis equal tight;
    title('Real part of incident field on the object domain');
    xlabel('x','FontSize',13,'FontWeight','bold');
    ylabel('y','FontSize',13,'FontWeight','bold');

    figure();
    imagesc(x, y, imag(uinc));
    set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
    colorbar;
    axis equal tight;
    title('Imaginary part of incident field on the object domain');
    xlabel('x','FontSize',13,'FontWeight','bold');
    ylabel('y','FontSize',13,'FontWeight','bold');

    figure();
    imagesc(x, y, abs(uinc));
    set(gca,'XAxisLocation','top','YAxisLocation','left','YDir','reverse');
    colorbar;
    axis equal tight;
    title('Absolute value of incident field on the object domain');
    xlabel('x','FontSize',13,'FontWeight','bold');
    ylabel('y','FontSize',13,'FontWeight','bold');
end
```