HSB
Hochschule Bremen
City University of Applied Sciences

UNDERWATER ACOUSTICS AND SONAR
SIGNAL PROCESSING

SS 2018

ASSIGNMENT 2

SOUND ATTENUATION IN THE OCEAN

Date: 16/04/2018

by

Shinde Mrinal Vinayak(Matriculation No.: 5021349)
Kshisagar Tejashree Jaysinh (Matriculation No.: 5019958)

under the guidance of

Prof. Zimmer

HSB
Hochschule Bremen
City University of Applied Sciences

Contents

Chapter 1

Introduction

Salt water as a medium is dissipative. Due to viscosity of salt water (above 100 kHz), relaxation (i.e. the conversion of acoustic energy into heat) of magnesium sulphate (above 100 kHz or 10 kHz to 100 kHz) and boric acid (above 1 kHz or $\leq$10 kHz) sound waves are attenuated. There are numerous empirical models available for the prediction of attenuation of underwater sound, e.g. Thorp, Schulkin-Marsh and Francois-Garrison. The absorption of low frequency sound is comparably small. However, the magnitude of absorption increases with increasing frequency. In contrast to the sea surface, sound waves can penetrate into soft sea floors and consequently be absorbed significantly. Since the sea bed is a solid, also shear waves can develop. At long distances, losses in acoustic sound intensity result from attenuation and scattering losses. Sound attenuation in underwater environments is dependent on temperature, salinity, pressure and pH. If sound is propagating into the sea bed, it also is attenuated as a function of sediment type.

This assignment consists of three parts: In the first part we develop a MATLAB program that calculates the sound attenuation in seawater by means of Thorp, Schulkin & Marsh and Francois &

Garrison formula. The results of the three approaches are then compared. The next two parts are done for Francois & Garrison formula. The dependency on the frequency, salinity and temperature for a depth of 50m is investigated and finally a graph for attenuation versus frequency is plotted and frequency regions where the different attenuation processes dominate are specified.

Chapter 2

Theory

Variations of the sound velocity in the ocean are relatively small due to low absorption compared to speed of light in water. Blue-green light has the lowest attenuation in water upto 20m while sound can travel upto 100km one way and hence sonar is used to navigate, communicate with or detect objects on or under the surface of the water. Sound velocity in ocean lies between about 1450 m/s and 1540 m/s. Even though the changes of sound speed are small, i.e. $\leq \pm 3\%$, the propagation of sound can be significantly affected.

The sound velocity can be directly measured by velocimeters or calculated by empirical formulae if the temperature, salinity and hydrostatic power or depth are known. However, the error of measurements by modern velocimeters is about 0.1 m/s. The accuracy of calculations by the most complete empirical formulae is about the same. Also formulae providing such high accuracy are very cumbersome.

A less accurate but simpler equation is given by:

$$c = 1449.2 + 4.6T - 0.055T^2 + 0.00029T^3 + (1.34 - 0.01T)(S - 35) + 0.016z$$

where temperature, T in ($^\circ C$), salinity, S in (ppt), depth, z in (m) and sound velocity, c in (m/s)

With the following limits:

$0^\circ C \leq T \leq 35^\circ C$

0 ppt $\leq$S $\leq$45 ppt

0 m $\leq$ z $\leq$1000 m

The function for velocity of sound in ocean and the dependency of nonlinear temperature profile and the depth of the ocean is studied. Two sets of waveforms of c versus z for various sets of T and S are plotted in-order to investigate the dependence of c on T, S and z. Also the waveform for the nonlinear temperature profile and the depth of the ocean is obtained and discussed in the following sections.

Chapter 3

Experimental research

By using the formula for velocity of speed in ocean, the graph of velocity of speed in ocean versus depth is plotted for various values of temperature.

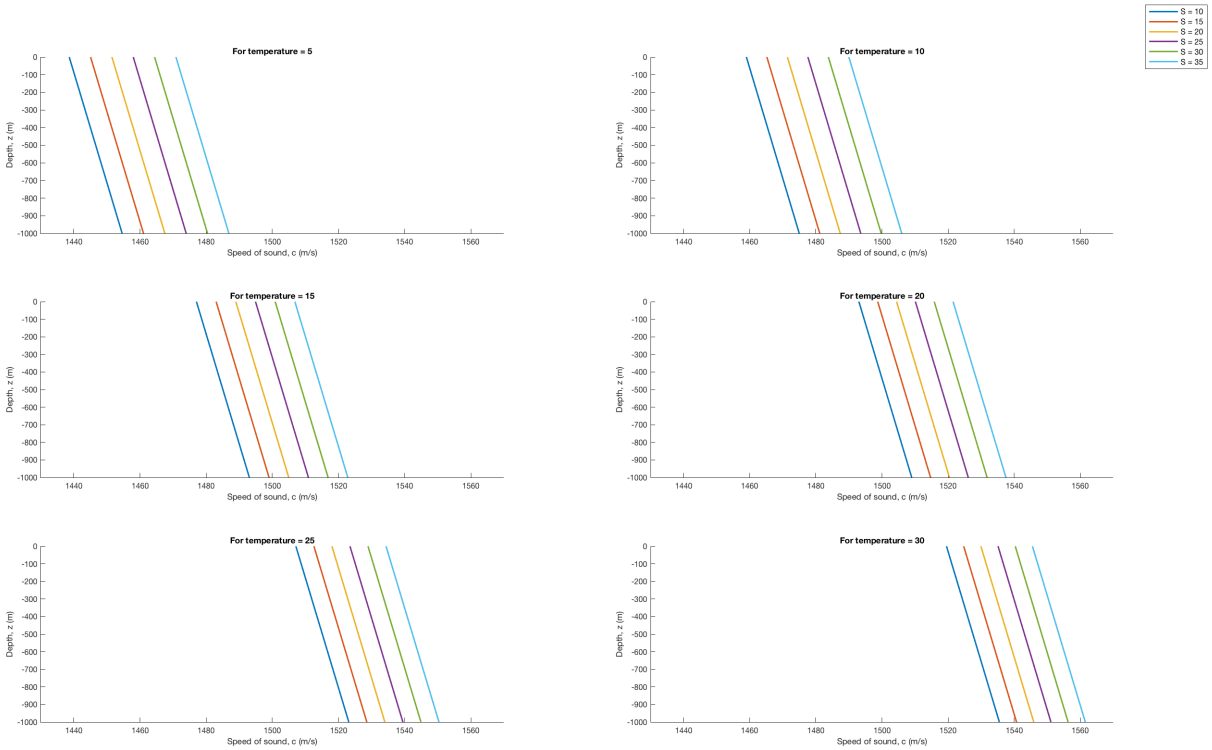

Figure 3.1: c versus z for various sets of T

By using the formula for velocity of speed in ocean, the graph of velocity of speed in ocean versus depth for various values of salinity is obtained.

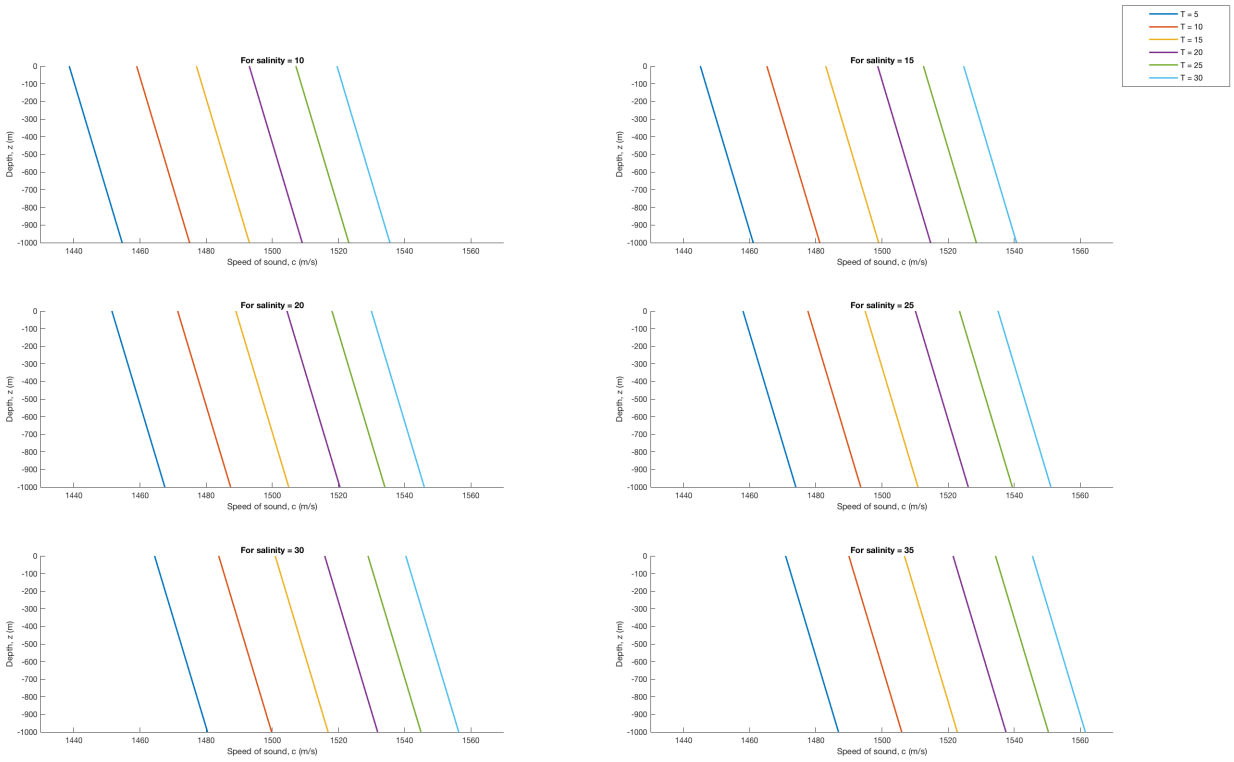

Figure 3.2: c versus z for various sets of S

The dependency of non-linear temperature profile and depth is plotted using an exponential function for temperature. The plot contains of three curves: temperature versus depth, sound velocity (having constant salinity and temperature) versus depth and sound velocity (having expoenential temperature and constant salinity) versus depth.

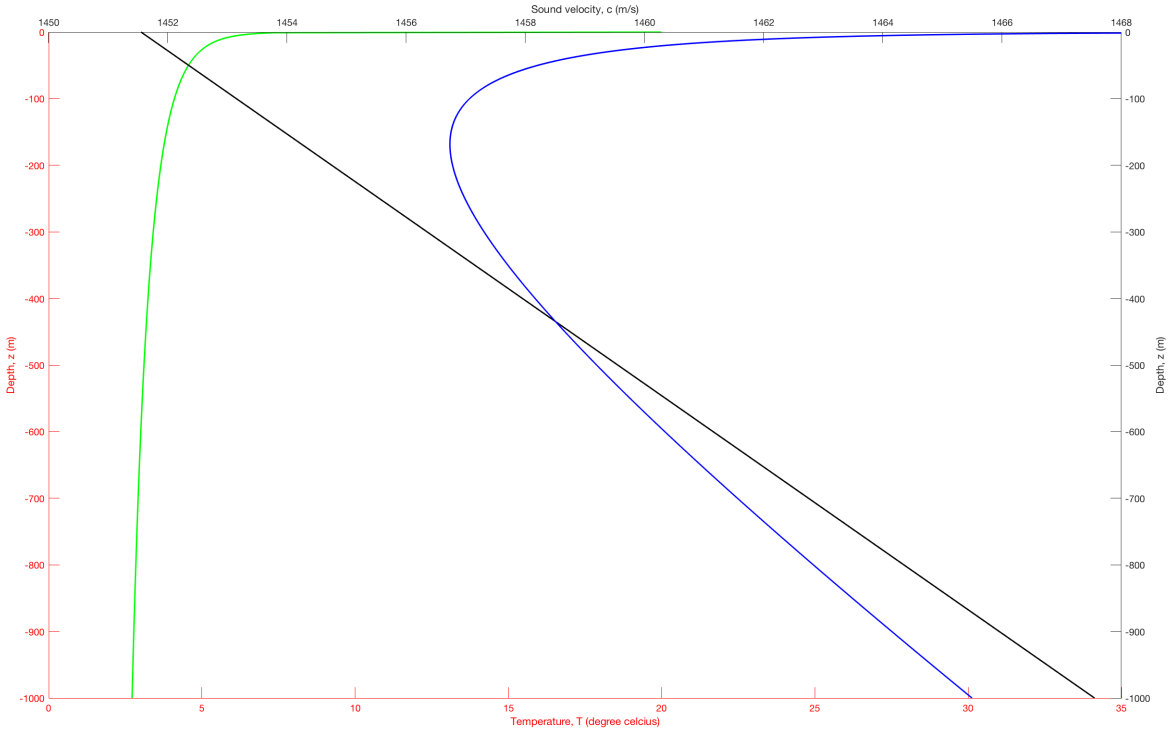

Figure 3.3: Dependency of non-linear temperature profile and depth

Chapter 4

Conclusion

It can be concluded that depth and salinity of the ocean are linearly dependent on sound velocity in the ocean while temperature has non linear dependance on speed of sound in the ocean

Chapter 5

Appendix

5.1 MATLAB code for dependence of c on T, S and z

5.1.1 Function to compute velocity of sound

```
% Function saved as speedOfSound.m and is called from other MATLAB files
function c = speedOfSound(S, T, Z)
c = 1449.2 + 4.6*T - 0.055*T.^2 + 0.00029*T.^3 ...
                + 1.34*S - 46.9 - 0.01*T.*S + 0.35*T + 0.016*Z;
end
```

5.1.2 For various values of temperature

```
%file saved as UCP1_1.m
temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
```

```
3  salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
4  depth = 0:200:1000; % depth varies in steps of 200 from 0 to 1000
5  for i = 1:6
6      u = temperature(i);
7      for j = 1:6
8          v = salinity(j);
9          c = speedOfSound(v, u, depth); % calling the function
10         subplot(3,2,i);
11         hold on;
12         plot(c,-depth,'LineWidth',1.5);
13         title(['For temperature = ',num2str(temperature(i))])
14         ax = gca; % current axes
15         ax.FontSize = 8;
16         ax.XLim = [1430 1570];
17
18        % labelling of axes
19         ylabel('Depth, z (m)')
20         xlabel('Speed of sound, c (m/s)')
21     end
22 end
23 % labelling of waveforms
24 hL = legend('S = 10', 'S = 15', 'S = 20', 'S = 25','S = 30', 'S = 35');
25 newPosition = [0.85 0.85 0.2 0.2];
```

```
newUnits = 'normalized';
set(hL,'Position', newPosition,'Units', newUnits);
```

5.1.3 For various values of salinity

```
% file saved as UCP1_2.m
temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
depth = 0:200:1000; % depth varies in steps of 200 from 0 to 1000
for i = 1:6
    u = salinity(i);
    for j = 1:6
        v = temperature(j);
        c = speedOfSound(u, v, depth); % calling the function
        subplot(3,2,i);
        hold on;
        plot(c,-depth,'LineWidth',1.5);
        title(['For salinity = ',num2str(salinity(i))])
        ax = gca; % current axes
        ax.FontSize = 8;
        ax.XLim = [1430 1570];
        ylabel('Depth, z (m)')
        xlabel('Speed of sound, c (m/s)')
    end
```

```
20 end
21 hL = legend('T = 5', 'T = 10', 'T = 15', 'T = 20','T = 25', 'T = 30');
22 newPosition = [0.85 0.85 0.2 0.2];
23 newUnits = 'normalized';
24 set(hL,'Position', newPosition,'Units', newUnits);
```

5.2 MATLAB code for dependence of non linear-temperature profile and depth

```
1 temperature = 5:5:30; % temperature varies in steps on 5 from 5 to 30
2 salinity = 10:5:35; % salinity varies in steps of 5 from 10 to 35
3 depth = 0:1000; % depth varies in steps of 1 from 0 to 1000
4 u = 4; % salinity
5 v = 10; % temperature
6 expo_temp = 20*exp(-depth.^.1);
7 line(expo_temp,-depth,'color','g','LineWidth',1.5);
8 ax1 = gca; %current axis
9 ax1.XColor = 'r';
10 ax1.YColor = 'r';
11 ax1.XLim = [0 35];
12 ylabel('Depth, z (m)')
13 xlabel('Temperature, T (degree celcius)')
14 ax1_pos = ax1.Position; % position of first axes
```

```
15 ax2 = axes('Position',ax1_pos,...
16     'XAxisLocation','top',...
17     'YAxisLocation','right',...
18     'Color','none');
19 ax2.XLim = [1450  1468];
20 ax2.YLim = [-1000  0];
21 ylabel('Depth, z (m)')
22 xlabel('Sound velocity, c (m/s)')
23 c = speedOfSound(u, v, depth); % calling the function
24 line(c,-depth,'Parent',ax2,'Color','k','LineWidth',1.5)
25 c = speedOfSound(26, expo_temp, depth); % calling the function
26 line(c,-depth,'Parent',ax2,'Color','b','LineWidth',1.5)
```