Ultrasound Project Report

Langechuan Liu

October† 2009

This report has two parts.

In Part I, the answers to the questions asked in the project will be answered. For most of the questions, please refer to the commented Matlab code attached to this report. In this report I will provide supplementary explanations and discussions to the questions.

In part II, as a sanity check, ultrasound raw data are generated using a new object comprising of a collection of reflectors located at known positions. (Matlab code for doing so is also provided with comments.) Using the beam-forming and scan-conversion routine developed in part I, reconstructed ultrasound image is obtained, in comparison with the original object.

The attachment list,

<table>
<thead>
<tr>
<th>subroutine</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>beamform_v1.m</td>
<td>beam-forming subroutine</td>
</tr>
<tr>
<td>scancon_v1.m</td>
<td>scan-conversion subroutine</td>
</tr>
<tr>
<td>beamform_v2.m</td>
<td>alternative way of phase correction</td>
</tr>
<tr>
<td>SM_gen.m</td>
<td>subroutine for setting up new reflector positions</td>
</tr>
<tr>
<td>data_gen.m</td>
<td>subroutine for generating ultrasound raw data</td>
</tr>
</tbody>
</table>

Timeline:
October 12, 2009 image reconstruction code finished
October 14, 2009 raw data generation code finished
October 18-22, 2009 answers sorted and \LaTeX ed
1 Part I Answers, explanations & discussions

Question A The `showimage` subroutine can be seen as a more powerful `imagesc`.

![Rawdata wavefield plot](image)

Figure 1: Problem A

Question B Suppose the width of the element is the same as the center-to-center spacing, then citing the equation from lecture notes (A.23), we have
\[ \Delta \sin \theta = \frac{\lambda/2}{D} = \frac{\lambda}{2Nd} = \frac{2d}{2Nd} = \frac{1}{N} = \frac{1}{65} \]  

(1)

For sector angle 90°, \( \theta \in [-45^\circ, 45^\circ] \), therefore \( \sin \theta \in [-\sqrt{2}/2, \sqrt{2}/2] \)

\[ \text{Number of beams} = \frac{\sqrt{2}/2 - (-\sqrt{2}/2)}{\Delta \sin \theta} = 65\sqrt{2} \approx 91.9 \]  

(2)

In order to adequately sample angularly and make number of beams an odd number for \( \theta=0 \) A-mode scan later\(^\dagger\), number of beams is chosen to be \( n_{\text{beam}}=93 \).

**Question C**  After using baseband subroutine, the real part and imaginary part of baseband data is shown in Figure 2 using showimage subroutine.

![Figure 2: wavefield plot of baseband data](image)

\(^\dagger\)Strictly speaking, this is not needed since finally interpolations will be performed. If beam number is odd, the \( \theta=0 \) A-mode scan can be obtained directly even before interpolation. Refer to Question I for detailed discussion.
Then raw data and the absolute value of baseband data are plotted in comparison in Figure 3:

![Figure 3: raw data and baseband data](image)

This is in accordance with the theoretical analysis in the lecture notes on Page A10 and A13 – baseband data is taking the envelope of the absolute values of the original data, turning a fast-varying high frequency signal to a slow-varying low frequency signal. Therefore, when sampling at \( f_s = 16\,MHz = 4f_0 \) (carrier frequency), *i.e.* 4 points per period, neighboring points in original data can be as different as the local amplitude, leading to the irregular “white spot” in gray area. Meanwhile for baseband data, sampling rate \( f_s \) is adequate for depicting the envelope, therefore it has more recognizable pattern than the raw data with no white spots, as displayed in Figure 3.

**Question D & E** The main equation used for

\[
\hat{R}(r, \theta) = \frac{d_0 d_n}{N} \left| \sum_{n=1}^{N} b_n \left( \frac{d_0 + d_n}{c} \right) e^{-i2\pi f'_0 (d_0 + d_n)/c} \right| 
\]  

(3)
where demodulation frequency $f' \approx$ carrier frequency $f_0$, $N$ is the total number of elements in the array noted as $\text{nelem}$ in Matlab code. The total distance the signal traveled is

$$d_{\text{tot}}(r, \theta) \triangleq d_0 + d_n = \sqrt{(r \sin \theta - x_0)^2 + (r \cos \theta)^2} + \sqrt{(r \sin \theta - x_n)^2 + (r \cos \theta)^2}$$

$$= \sqrt{r^2 + x_0^2 - 2rx_0 \sin \theta} + \sqrt{r^2 + x_n^2 - 2rx_n \sin \theta}$$

where $d_0$ is the transmit distance from transmitter $(x_0, 0)$ to reflector $(r, \theta)$, and $d_n$ is the receive distance from reflector back to different elements of the transducer array $(x_n, 0)$. Thus the total time consumed is

$$t_n(r, \theta) = \frac{d_0 + d_n}{c}$$

where $c$ is the sound of ultrasound, $\text{vsound}$ in Matlab code.

Phase rotation is $\Delta \phi_n(r, \theta) = 2\pi f_0 t_n(r, \theta)$, thus the correction term is

$$\Phi_n(r, \theta) = e^{-i\Delta \phi_n} = e^{-i2\pi f_0 t_n}$$

Since we have only discrete samples, we need to find the correct time delay (“table lookup”) and interpolate. The time interval between two consecutive sample points is $\Delta t = 1/f_s$, thus ideally number of time clicks corresponding to $t_n$ is $\text{tick}_n = t_n/\Delta t = t_n f_s$, which is usually not an integer. Linear interpolation using the weighted sum of two nearest neighbors $\text{floor}(\text{tick}_n)$ and $\text{ceil}(\text{tick}_n)$ is used in $\text{beamform.m}$ (equivalent to interpolate with a triangular function). Single point interpolation with only the nearest neighbor $\text{round}(\text{tick}_n)$ is also explored, with results of no much difference\footnote{the only slightly observable difference is the “trail” artifacts discussed later, so pictures are not shown here.}.

Another Perspective Interestingly, there is another way to interpret and implement phase rotation correction term $\Phi_n(r, \theta)$. Every row in the matrix of $\text{databb}(1400 \times 65$ here) is collected at the same time point, e.g. the $m$th row is collected at the $m$th tick, i.e. the time point $m\Delta t$. This is the time between the sending and receiving of the signals for all channels. Note that for different channels, the information collected at the same time does not come from the same reflecting point.
More specifically, for a single channel, e.g. the \textit{nth channel}, the data collected at time $t$ are the superposed wavefronts reflected back from a certain collection of points\textsuperscript{†}. Thus the distance these signals have travelled before being collected is $mc\Delta t$, the phase rotation at the \textit{n}th channel is

$$\Delta \phi_n(r, \theta) = 2\pi f_0 t_m = 2\pi f_0 m \Delta t$$  \hspace{1cm} (9)

which is used in the phase correction term of $\text{data}_{bb}(m,n)$.

There are two things worth pointing out. First, although one single data point $\text{data}_{bb}(m,n)$ has information from a collection of reflecting points, they share the same phase rotation because they share the same propagation time. Second, this phase rotation depends only on time, \textit{i.e.} on which row the data point is – it does not depend on the channel $n$ nor the reflecting points $(r, \theta)$.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{scan_level_20dB.png}
\includegraphics[width=\textwidth]{scan_level_50dB.png}
\caption{Sector scan images using two ways of phase correction}
\end{figure}

These two facts make it possible to make a more “sorted” phase correction.
tion to the baseband $\text{databb}(\cdot, \cdot)$ before interpolation. The drawback of this method is that we are not sampling fine enough because the phase factor is changing rapidly. It can be estimated that the largest possible phase error is $2\pi f_0 (\Delta t/2) = \pi f_0 / f_s = \pi/4$. Theoretically, if the signal can sampled “quasi-continuously” in time domain, $f_s \to \infty$, this method will produce the same result as the original method employed above.

Beamforming using this method works, although not good, as shown in Figure 4. Reflectivity values become very small due to unsatisfying constructive interference because of phase error. Not until 50 dB does the shape of the mystery object become recognizable. Once going over 80 dB, it is hard to tell the shape any more, thus the observation “dB window” is narrower than using the method with exact phase correction.

![Figure 4: Beamforming results using different radial sampling intervals.](image)

(a) $\Delta r=0.05\text{mm} \ 40\text{dB}$
(b) $\Delta r'=0.5\text{mm} \ 40\text{dB}$

Figure 5: Sector scan images using different radial sampling intervals

**Radial sampling** Besides a careful choice of angular interval to sample adequately, radial interval also need to be determined. Supposedly there
are two reflectors very close to each other, \((r, \theta)\) and \((r + \Delta r, \theta)\). If signals reflected by these two reflectors can be distinguished by detectors, \(2\Delta r/c > \Delta t\), so sampling interval should be

\[
\Delta r \approx \frac{c\Delta t}{2} = \frac{c}{2f_s} \approx 0.05\text{mm}
\]

This is the sampling criterion for a single-element A-mode scanner. Two points with a distance apart smaller than \(\Delta r\) would not be distinguished, limited by finite \(f_s\). For B-mode scan, \(\Delta r\) might be smaller using multi-channel data, but it will not make much more significant improvement of image quality. Larger Sampling using \(\Delta r' = 10\Delta r \approx 0.5\text{mm}\) makes observable but not severe deterioration of image quality, as shown in Figure 5.

![Image of graph](image.png)

Figure 6: \(r - \sin\theta\) buffer

**Question F** \(r - \sin\theta\) buffer is plotted in Figure 6. Images using `showimage(sin_theta, r, abs(rsdata), 0, dbscale), axis normal` can be equally achieved by
rsdata = rsdata / max(max(rsdata));
logrsdata = 20 * log(rsdata) / log(10);
logrsdata( find(logrsdata < -dbscale) ) = -dbscale;
imagesc(sin_theta, r, logrsdata);
colormap(1-gray(256)); colorbar; axis normal;

Question G  To convert data from one coordinate systems to another,

1. establish the grid(matrix of spatial position\textsuperscript{†}) in the target coordinate system(most favorably evenly spaced);

2. then convert the coordinates in this grid to the original coordinate system(where the information of data points are known)

3. interpolate using Matlab subroutines, e.g. interp, interp2\textsuperscript{†}

In order to display the area out of FOV wedge in the same color as the reflector, then it should be padded with 1, which will turn into 0dB after being mapped into logarithm scale.

Question H  Comparison of 20dB and 40dB level scans are compared side by side below. In the 20dB image, it is more neat and easy to observe the whole shape, although some details might be missing.

Description  There are five discrete reflectors altogether, three on/near the z-axis with two of them pretty close to each other, and two others near [±20, 25] mm. There are also two “quasi-continuous” groups of reflectors, forming a square altogether. One group is of a trapezoid shape with a square hole inside, and has higher reflectivities. The other group is of an isosceles right triangle with lower reflectivities.

\textsuperscript{†}Pixel and real spatial distance should be distinguished. Every data point in a matrix is only one pixel(although can be displayed in a larger single-valued area), or the value of one point. It does not have information about real spatial positions itself. In order to know the coordinates of the pixel, other matrices are needed, as are generated by meshgrid or ndgrid.

\textsuperscript{†}First two parameters of subroutine interp2 cannot be changed in order. Both are matrices of the same size generated by duplicating particular vector respectively(1-dim matrices) – a row vector in the case of the first matrix parameter and a column vector for the second.
Artifacts In the section scan, the most striking artifacts is the smearing in the angular direction with lots of arch trails, like a car rain-wiper wiping through the reflecting body. A closer observation of the discrete reflectors nearer to the origin would reveal that there are two arch trails per reflector.

A minor feature is that the arch trails are not continuously diminishing; rather they become more like dotted lines after some distance away from the reflector points. This is very obvious on both sides of the reflector blocks in the lower part of the image. †

Initial Explanation, Guesses & Trials My initial guess was‡

†This is not exactly caused by the same mechanism of either sidelobes or grating lobes. Discussed later in Question H.
‡I wrote this part down just for the record of my thinking process. For logical continuity, please skip to the next part, Final Explanation. Unless specified, all the numerical values of length in this part is in mm.
Radially, the envelope of the signal determines the radial spatial resolution. If the envelope is a Gaussian shaped pulse containing 2 or 3 periods, then radial PSF is a shifted Gaussian function, rapidly from the location of the reflector with increasing or decreasing radius. Angularly, it is a slowly modulated sinc function, with sidelobes and grating lobes stretching very far away compared with radial PSF. That can explain why the trail become dotted or dashed lines some distance away from the reflector point.

Actually the far-field criterion is

\[ r \gg D^2/\lambda \approx (65 \cdot 0.1925)^2/(0.1925 \cdot 2) \approx 406 \]  \hspace{1cm} (11)

obviously the object we scan is in the near field because the furthest point is only about 70mm away from the origin. Thus we cannot qualitatively explain why there are two trails per reflector using the far-field approximation PSF derived in lecture notes Page A17.

Let’s take some wild guesses here. Take the point nearest to the origin and on z-axis for example. One of the trail is nearly horizontal, the other is more tilted towards right. Parameters known:

- size of the array \( D = 65 \cdot 0.1925 \approx 12.5 \)
- positions of receiving array \( x_n \in [-x_{n \text{max}}, x_{n \text{max}}] = [-6.25, 6.25] \)
- position of transmitter is \([x_0, 0] = [10, 0] \)

Suppose the two trails of one single reflector are two arc of two circles. The center of the two circle should be around \([0, 0]\) and \([7, 0]\) (estimated by naked eye). One intuitive guess would be that the two centers of circles are \([\pm x_{n \text{max}} + x_0]/2, 0]\). Then why just the boarders of the array? One intuitive guess would be

All the contributions in the middle are “somehow” cancelled out by the contribution from their neighbors, thus leaving the two ends “uncancelled” (more exactly only partially cancelled by its only neighbor on one side).

To prove the guess, two trials are taken:
Figure 8: Results of the several trials. (a) and (b) are in $x-sin\theta$ coordinates, (a) being the image reconstructed with only the central (33rd) channel, (b) the image using the full array except the central channel. In (b), a third new “trail” appears the same place where the four parallel lines are in (a). Image converted to (c) and (d) respectively. In (a) and (c), reflectivity values are normalized to make the maximum value unity in convenience of display.
1. Images of $r - \sin \theta$ buffer and sector scan using only the data from the elements on both ends of the array are plotted. They are collections of parallel lines, either image with the uppermost trail in the same position as the above-mentioned two trails. This shows that the two trails are indeed from the contributions of the two elements on either side of the array.

2. One image using all elements except the middle one at the origin (the 33rd one here) and the other using only the middle one is plotted. The first image now has its characteristic “three-trail” feature on the uppermost reflector, and the position of the third trail is in the middle of the previous two, and agrees with the position of that trail in the second image obtained only with the middle element. This shows that the contribution of a certain array element is indeed cancelled out by other elements, mostly arguably neighboring ones because of limited span of radial PSF.

These two trials altogether prove in a empirical way that the guess is right. The cancel-out between neighboring channels is due to destructive interference results from the difference in phase rotations.

Another question arises after the image is reconstructed with the data from only one channel. Why the image of a single reflector is an arc in the sector scan? It means that without constructive inter-channel interference, one single channel cannot distinguish all the points in that arc tail. Then how about magnitude on that trail? A contour image generated by command `contour` shows clearly that all the points in that trail have the same reflectivity value, showing that it is not diminishing with angles (now the $x$-axis). The reason why it is unlike the PSF of a single-element scanner with a diminishing side-lobe is that the wave sent out is a isotropic spherical wave instead of being laterally limited by an aperture function.

All the points on that trail must have a certain same property that makes the single channel incapable of telling them apart. The wave reflected by a single reflector and collected by one channel itself contains gives information on the position of the reflector, but only about “distance”, more exactly the sum of the distances back and forth, $d_{n}^{\text{tot}} = d_0 + d_n$, as shown in Equation 3. Therefore the points on the
Figure 9: (a) and (b) are the plot reconstructed with only the element on either end. In order to see that the reflectivity on the same trail is the same, contours are plotted in (c). To get better view, (d) is the enlarged plot of the boxed area in (c)
trail should have the same sum of distance to the transmitter and receiver channel. This brings everything to light – the trail in the sector scan is not an arc from a circle, but part of an ellipse. In Figure 10, \( T \) is the transmitter \([10,0] \text{mm}\), \( R \) is the receiver of the first channel \([-D/2,0] = [-6.16,0] \text{mm}\).

![Figure 10: the formation of elliptical trail](image)

**Final Explanation** The double-trail blur image of a discrete single point reflector is the total PSF, which can be seen as the superposed single PSF of every channel (similar to the expression of \( \hat{R}(r, \theta) \)).

\[
PSF = \sum_{n=1}^{N} PSF_n \tag{12}
\]

\[
PSF_n = \frac{d_0 d_n}{N} a_n \left( t - \frac{d_0 + d_n}{c} \right) e^{-i2\pi f_0^0(d_0+d_n)/c} \tag{13}
\]
For one single reflector, there are two **arc-like trails**. The trail is not exactly arc from a circle, but part of an **ellipse** with one focus located at the transmitter \([x_0, 0]\) and the other located at the element at either end of the phased array \([\pm x_{\text{max}}, 0]\). If the wavepacket of the signal \(a_n(t)\) is an ideal Dirac delta function, then

\[
PSF_{n}^{\text{ideal}} = \frac{d_0d_n}{N} \delta(r - (d_0 + d_n))e^{-i2\pi f_0(d_0+d_n)/c}
\]

which is an equation of ellipse. Since the major axis is much larger than the focal distance\(^1\), the part of the ellipse can be approximated by an arc with center of the circle located at the center of the ellipse \([[(\pm x_{\text{max}}^n + x_0)/2, 0]\). Also the central angle of the arc \(\theta \ll 1\), thus can be further approximated by

\[
r = a + \frac{(\pm x_{\text{max}}^n + x_0)\sin\theta}{2}
\]

\(a\) is the radius of the circle used to approximate the exact ellipse. \(^2\)

In \(r - \sin\theta\) coordinate system, Equation (11) gives two straight lines, which agree with Figure(9).

\[PSF = \sum_i PSF_i,\] in other words, every channel would have its own contribution of the two-trail pattern(PSF), but when added through all the channels, the contributions(waves, which can have negative value) are added up destructively due to difference of phase rotation term, except the position near the location of the reflector, where all channels interfere constructively. The destructive interference mainly happens between neighboring channels, because the radial span of PSF is very limited compared with angular direction. This leaves the two elements

\(^1\)i.e. the eccentricity \(e \approx 1.\)

\(^2\)\(a\) can be approximated by

\[a \approx ct/2 \approx c \cdot tick \Delta t/2 = c \cdot tick/(2 f_s)\]

where \(tick\) is the average of the row number where first none-zero elements appear across all channels. For example, from Figure 3(b), \(tick\) for the uppermost point is approximately 450, leading to \(a \approx 21\). Then Equation(11) is the equation for those two trails in polar coordinate systems.
at both ends of the array standing out of the group which only have neighbor on one side, marking the PSF pattern with **two trails**.

**Question I** There are actually two slightly different ways to do this. The first is cut the central piece located at z-axis out of the sector scan image, plotted in Figure 5(a) below. Since $\theta = 0$ A-scan is pretty special, it can be directly obtained from $\sin \theta = 0$ column of $\text{rsdata}$ even before “bilinear” interpolation, as plotted in Figure 5(b). The peak is a slightly a little lower after interpolation.

- After interpolation $\text{plot}(z, \text{image(:,21)})$
- Before interpolation $\text{plot}(r, \text{rsdata(:,33)})$

![A-scan image at theta=0](image1.png) ![reflectivity before interpolation](image2.png)

(a) post-interpolation plot using $\text{image}$  (b) pre-interpolation plot using $\text{rsdata}$

Figure 11: A-Mode scan at $\theta=0$

The maximum value\(^\dagger\) is 0.92, which is almost 1, representing the reflector at about $[0,20]mm$, nearest to the origin. In Figure 5(a), after $r \approx 60mm$,

\(^\dagger\)before interpolation the maximum is 0.98, even closer to 1
the reflectivity goes up to and stays at 1 because of the data padding outside the POV wedge.

**Question J** After we know the approximate configuration of the reflectors in the POV field, a *post hoc* examination of the wavefield plot in Question A can reveal more information (shame that we are not smart enough to realize it sooner :-/ ).

In both images, left side of the image correspond to the first transducer located at \([-D/2, 0]\). There are four obvious stripes before the more messy pattern in the wavefield plot. The first stripe is symmetric around the central element, indicating the symmetric geometry in the reflector’s real position. It should be nearest to the origin, namely the reflector located at around \([x, z] = [0, 20]mm\). As a side-proof or double check, the stripe is around the \(350 + 100 = 450th\) row, thus the distance from the origin should be

\[
r \approx c \cdot \text{tick}\Delta t/2 \approx 1.54 \cdot 450/(2 \cdot 16) \approx 21.6mm
\]  

\(16\)
which agrees with the reasoning above. The arc is bent upward in the middle part because the reflected sound wave need a little more time to reach the element in the ends of the array than the middle element. Similar reasoning can apply to the second stripe at around $350 + 150 = 500th$ row, corresponding to the reflector at about $[0, 23]mm$.

The third stripe is tilted toward left, $i.e.$ to the first element, indicating that the $65th$ element receives the signal earlier than the $1st$ one. Therefore it should be nearer to the $65th$ element, and the reflector around $[13, 24]mm$ satisfies this. From another perspective, the second and third stripe almost join together for the $65th$ channel. If we use what we learnt in Question $H$, the relevant two reflectors should sit on one ellipse with center at around $(13 + 0)/2 = 6.5mm$ on $x$ axis. This largely agrees with $(x_0 + x_mA)/2(10 + 6.25)/2 \approx 8mm$. The fourth one can be reasoned in the similar way, mapping to the $[-14, 22]mm$ reflector in the sector scan.

Interestingly, in the sector scan we can see there is a reflector at around $[x, z] = [0, 58]mm$, leading us to expect one “arc-like” stripe in the wavefield plot at around

$$\text{tick} \approx \frac{2r}{c\Delta t} = \frac{2rf_s}{c} = \frac{2 \cdot 58 \cdot 16}{1.54} \approx 1200th \text{ row}$$

Looking closely at the fourth column of the wavefield plot of either raw data or baseband data, at around $1200 - 350 \cdot 3 = 150th$ row, which is at the end of the blurring block, there is indeed a barely recognizable “arc-like” stripe, like the first and second ones discussed above. This seems a little far-fetched, but since the stripe is still recognizable, this little discovery is attributed to the virtue of $post \, hoc$ analysis.

Then the first blurring block begins at around $2 \times 350 + 50 = 750th$ row, corresponding to a distance of around $36mm$ from the origin, agrees with the upper boarder of the first trapezoid reflector group. The blurring is becoming dimmer because further away from the origin, the triangle group has a larger proportion of contribution, resulting in a weaker reflection of signals. The end of the blurring ends almost the same position as the very vague stripe corresponding to the lowermost discrete reflector at $[0, 58]mm$. 

19
2 Part II Generation of ultrasound data

This part is about the sanity check as recommended by the professor.

2.1 positioning of reflector points

Out of curiosity of how good this whole system of data-generation and reconstruction works for more complex shapes, I decided to use a traditional Chinese character 劉 (yes, my family name Liu in Regular Script :P ) as the scattering object.

![Figure 13: preparation of collection of reflectors](image)

First one 300×300 PNG picture of the character is obtained using graphic software and then loaded it into Matlab, generating one 300×300 matrix. It is then relocated and resized into a 200×200 area in the lower-middle part of a 401×500 matrix, with value 1 inside the character, and 0 outside. This is used to sift randomly generated points inside the big matrix with reflectivity values assigned to 1.0 on the left, and 0.5 on the right (as shown in Figure 13).

Finally 3000 random points inside the character shape are generated. In order to observe behaviors of single point reflectors, four point reflector are
placed at [180, 200], [420, 200], [300, 100] and [350, 300]. As for parameters used for generation of wavefield raw data, $f_s = 4f_0 = 16Hz$ is used here, same as what is used in this project Part I. The signal is sent in Gaussian wavepacket,

$$a(t) = e^{-(t\omega_0-2)^2}$$

which contain about three periods of oscillation.

![Gaussian signal wavepacket](image)

Figure 14: Gaussian signal wavepacket (dashed line) and modulated signal (solid line). Note that center is not located at $t = 0$.

### 2.2 Method used in data generation

In raw data generation, every $\Delta t$, the data across all channels are collected. Every channel should have the superposition of wave pressure contribution from each of the 3000 reflectors, the signal received by the $n$th channel at time $t$ is

$$v_n(t) = \sum_{i=1}^{N_{re}} \frac{R}{d_0^i d_n^i} \alpha(t - \frac{d_0^i + d_n^i}{c}) \cos(2\pi f_0(t - \frac{d_0^i + d_n^i}{c}) + \phi_n)$$

these are row/column indices, not spatial positions
where
\[
d_0^i(r_i, \theta_i) \triangleq \sqrt{r_i^2 + x_0^2 - 2r_ix_0\sin\theta_i}
\]
\[
d_n^i(r_i, \theta_i) \triangleq \sqrt{r_i^2 + x_n^2 - 2r_ix_n\sin\theta_i}
\]
are the distance from transmitter to the \(i\)th reflector point, and the distance from the \(i\)th reflector point back to the receiver of the \(n\)th channel. \(N_{re}\) is the total number of reflector point, therefore 3000 in this project, \(\theta_n\) is set to zero across all channels.

Then raw data is beam-formed and scan converted to the sector scan, as plotted below:

![Scan level 30 dB](image1)

![Scan level 60 dB](image2)

(a) 30 dB scan  
(b) 60 dB scan

Figure 15: reconstruction of reflecting object

Things worked out pretty good, except that generating data for the 3000 points takes quite a while (around a minute on my laptop). There are approximately

\[
\text{ntime} \times \text{nelem} \times N_{re} \approx 1500 \times 65 \times 3000 \approx 3 \times 10^8 \text{ FLOPs}
\]

in the whole calculation process, each FLOP being one evaluation of the expression inside the summation of Equation (16).
In Equation (19), once the geometry of reflectors and the equipment is chosen, the number of transducer elements $nelem$ and number of reflectors $Nre$ is fixed, and

$$ntime = 2z_{\text{max}}/(c\Delta t) = 2z_{\text{max}}f_s/c$$

(23)

If sampling rate $f_s$ increases, the calculation time needed would increase linearly.

### 2.3 further exploration

As pointed out in **Part I Question D&E**, there is an alternative way of phase correction, which is not accurate generally, but can lead to sufficiently good quality once the sampling rate is large enough. Increasing sampling rate $f_s$ to $2f_s$ and $4f_s$, raw data are generated again, and then reconstructed images using these new data are compared with the image using exact phase correction under original sampling rate $f_s$ as plotted in Figure 16. For (d) where $f'_s = 4f_s = 16f_0$, there is hardly any visible difference from (a), supporting the reasoning before.
Figure 16: reconstruction using different ways of phase correction under 30dB

(a) exact phase correction with $f_s$

(b) alternative phase correction with $f_s$

(c) alternative phase correction with $2f_s$

(d) with alternative phase correction with $4f_s$