APPENDIX B

B.1 Complete5.m Script

```
% complete5.m
% written by Susan Mui
% modified by Yang Zheng
% May, 1998
%
% the complete program, for almost everything you ever wanted to know about
% a transducer
%
% this version is for data taken after July 1997, since Karen modified 2d.c
% to record both the speed of sound (sos) and the window time shift
% (timeshift)
%
% initial start position of scan is taken into account
% there is no filtering of the waveform

%
********************************************************************************
%
% pii_log.m
% calculates PII-matrix of 2-D data set, then takes 10*log10
%
%
********************************************************************************

clear all;
close all;

n=input('Enter a filename : ','s');
bin=[n '.bin'];
dat=[n '.dat'];

fid=fopen(dat,'r');
A=fscanf(fid,'%f',[11,1]);

ymult=A(1,1);           % y-scaling
yzero=A(2,1);           % DC offset
xincr=A(3,1);           % x-scaling
number_point=A(4,1);    % # pts in waveform
numscan1=A(5,1);  % # lateral scans (y)
numscan2=A(6,1);  % # axial scans (x)
xzero=A(7,1);           % time for start of first waveform
stepsize1=A(8,1); % lateral stepsize
stepsize2=A(9,1); % axial stepsize
sos=A(10,1);            % speed of sound
timeshift=A(11,1);      % window time shift

status=fclose(fid);

clear A
```

```
% opens and reads filename.b

tek=input('Was a Tektronix used to take this data? (y/n): ','s');
if (tek == 'n')
    fid=fopen(bin,'rb','b');  % 'b' for the LeCroy
else
    fid=fopen(bin,'rb','g');  % 'g' for the Tek
end

disp('Enter transducer position at start of scan')
startpos=input('c for close to wire, f for far from wire: ','s');

PII=zeros(numscan2,numscan1);
for(i=1:numscan2)
    for(j=1:numscan1)
        B=fread(fid,number_point,'short');
        B1=B.*ymult;
        if (tek == 'n')
            B2=B1-yzero;      % - for LeCroy
        else
            B2=B1+yzero;      % + for Tek
        end
        PII(i,j)=xincr*sum(B2 .* B2);
    end

end

PII = PII';                     % transpose to get axial direction
horizontally

if (startpos == 'f')           % flips the matrix so that the transducer is
    PII=fliplr(PII);           % located at the left of the plot
end

dbPII = 10*log10(PII);          % converts to decibel

x = (0:numscan2-1)*stepsize2;           % calculates axes for lateral and
x = x/1000;  % convert to mm from um    % axial distance
y = (0:numscan1-1)*stepsize1;
y = y/1000;

% plots the beam plot

plotitle=input('Enter title of plot: ','s');
figure
orient tall
subplot(311)
imagesc(x,y,dbPII)
axis('xy')
colormap('hot')
xlabel('Axial distance (mm)')
ylabel('Lateral distance (mm)')
title(plotitle)

%
********************************************************************************
%
```

```
% pii_sum.m
% calculates and produces the graph of the sum of PII versus axial distance
%
%
*******************************************************************************

subplot(312)
piisum=sum(PII);
plot(piisum)
ymax=input('input ymax: ');
xmax=x(numscan2);
plot(x,piisum)
axis([0 xmax 0 ymax])
axis('xy')
xlabel('Axial distance (mm)')
ylabel('sum of PII')

%
*******************************************************************************
%
% pii_contour.m
% makes a contour plot of the pii_log plot, plots 0,-6,-9 dB lines and
% gives the location of the maximum point in the contour plot
%
%
*******************************************************************************

dbpii=dbPII-max(max(dbPII));    % makes the maximum value of plot 0
subplot(313)

% contour(x,y,dbpii,[0,-6,-9])  % to make the axes on this plot the same
                                % as on the other two plots

contour(dbpii,[0,-6,-9])        % plots the contour
xlabel('Axial scans')
ylabel('Lateral scans')
%gtext('-6 dB')
%gtext('-9 dB')
[ypt,xptmax]=max(max(dbpii));            % calculates the location of the
[dblevel,yptmax]=max(dbpii(:,xptmax));   % maximum point on the plot
xptmax
yptmax

%
*******************************************************************************
%
% waveform.m
% finds the time waveform of the maximum point in the contour plot
%
%
*******************************************************************************

if (tek == 'n')
    fid=fopen(bin,'rb','b');   % b for Lecroy
else
    fid=fopen(bin,'rb','g');   % g for Tek
end
```

```
if (startpos == 'f')
    xptmax=numscan2-(xptmax-1);
end

for(j=1:xptmax-1)
   for(i=1:numscan1)
      wave=fread(fid,number_point,'short');
   end
end

for (i=1:yptmax)
   [wave,count]=fread(fid,number_point,'short');
end

% plots the waveform at the focus centered around 0 V

figure
orient tall
subplot(211)
twave=wave-mean(wave);
twave=wave-mean(wave);
twavescaled=twave*ymult;
timeaxis=(0:number_point-1)*xincr*1e6;
plot(timeaxis,twavescaled)
title(['Waveform at Focus for ', num2str(plotitle)])
xlabel('Time (microseconds)')
ylabel('Amplitude (V)')
grid
zoom         % Added to zoom waveform. SBS

%
*************************************************************************
%
% powerspec.m
% gives power spectrum of a time waveform
%
%
*************************************************************************

fftwave=fft(twave);
twavefft=fftwave.*conj(fftwave)/number_point;
twavefft=sqrt(twavefft);
pspec=20*log10(twavefft);
Pspec=pspec-max(pspec);

sampf=(1/xincr)/1e6;                    % sampling frequency
nyq=sampf/2;                            % nyquist frequency

freq=(0:(number_point/2)-1)*nyq/(number_point/2);

figure
plot(freq(1:number_point/2),Pspec(1:number_point/2))
title('Power Spectrum of Waveform at Focus')
xlabel('Frequency (MHz)')
ylabel('Magnitude (dB)')
grid
```

```
db3=-3*ones(number_point/2,1);
hold on
plot(db3(1:100),'r')
zoom

db3ref=input('Do you need to re-orient the 0 dB reference line? (y/n):
','s');
if (db3ref == 'n')
    figure(2)
    subplot(212)
    plot(freq(1:number_point/2),Pspec(1:number_point/2))
    title('Power Spectrum of Waveform at Focus')
    xlabel('Frequency (MHz)')
    ylabel('Magnitude (dB)')
    grid
    db3=-3*ones(number_point/2,1);
    hold on
    plot(db3(1:100),'r')
    zoom
end

if (db3ref == 'y')
    figure
    plot(Pspec(1:number_point/2))
    title('Power Spectrum with Frequency Index')
    xlabel('Frequency Index')
    ylabel('Magnitude(dB)')
    grid
    db3=-3*ones(number_point/2,1);
    hold on
    plot(db3(1:number_point/4),'r')
    zoom
    disp('Enter the frequency index of the point that you want to')
    db3index=input('re-reference the 0 dB line to: ')
    Pspecmax=Pspec(db3index);
    newPspec=Pspec-Pspecmax;
    figure(2)
    subplot(212)
    plot(freq(1:number_point/2),newPspec(1:number_point/2))
    title('Power Spectrum of Waveform at Focus')
    xlabel('Frequency (MHz)')
    ylabel('Magnitude (dB)')
    grid
    db3=-3*ones(number_point/2,1);
    hold on
    plot(db3(1:50),'r')
    zoom
end


%
********************************************************************************
%
% freqchar.m
% gives frequency characteristics such as center frequency, wavelength,
% bandwidth, and fractional bandwidth given the -3db frequency index points
% obtained from powerspec.m
```

```
%
%
*****************************************************************************

f1=input('input the 1st -3dB frequency (MHz): ');
f2=input('input the 2nd -3dB frequency (MHz): ');

center_frequency=(f1+f2)/2                          % in MHz
bandwidth=f2-f1                                     % in MHz
fractional_bandwidth=(bandwidth/center_frequency)*100  % in percent
wavelength=sos/center_frequency                          % in microns
f1                                          % in MHz
f2                                          % in MHz

%
*****************************************************************************
%
% pulsedurab.m
%
% plots absolute value of waveform so that the pulse duration can be
% found and finds the -20dB level pulse duration
%
%
*****************************************************************************

abs_wave=20*log10(abs(twave));
Abs_wave=abs_wave-max(abs_wave);

figure
plot(timeaxis,Abs_wave)
title('Absolute value of waveform at focus')
xlabel('Time (microseconds)')
ylabel('Magnitude (dB)')
db20=-20*ones(size(Abs_wave));
hold on
plot(timeaxis,db20,'r')
zoom
grid

time1=input('input 1st -20dB time (microseconds): ');
time2=input('input 2nd -20dB time (microseconds): ');
pulse_duration=(time2-time1)*1000                % in ns


%
*****************************************************************************
%
% ellipse.m
% assuming that the -6 dB contour line of a transducer beam plot forms an
% ellipse, then the depth of focus and the beamwidth can be found by
% measuring the major and minor axes of the ellipse
%
%
*****************************************************************************

C=contourc(dbpii,[0,-6]);
sizeC=size(C);
```

```
C=C(:,8:sizeC(1,2));
db6x=C(1,:);

clear i
sizeC=size(C);
pos6db=zeros(size(db6x));

for (i=1:sizeC(1,2))
     if (db6x(i) == -6)
         pos6db(i)=i;
     end
end

pos6dbpts=pos6db(:,:) > 0;
pos6db=pos6db(:,pos6dbpts);
C(:,[pos6db])=[];               % deletes columns of C that contain -6

db6x=C(1,:);
db6y=C(2,:);

figure
plot(db6x,db6y)
title('-6dB contour')
xlabel('x value')
ylabel('y value')
hold
grid
zoom

clear i j

sizeC=size(C);
distance=zeros(1,sizeC(1,2));

if (startpos == 'f')
    xptmax=numscan2-(xptmax-1);
end

for (i=1:sizeC(1,2))
     X=xptmax-db6x(i);
     Y=yptmax-db6y(i);
     x2=X^2;
     y2=Y^2;
     distance(i)=sqrt(x2+y2);
end

minvalue=min(distance);
maxvalue=max(distance);

for (j=1:sizeC(1,2))
     if (distance(j) == maxvalue)
         majaxindex=j;
     elseif (distance(j) == minvalue)
         minaxindex=j;
     end
end
```

```
xmaj=db6x(majaxindex);
ymaj=db6y(majaxindex);
xmin=db6x(minaxindex);
ymin=db6y(minaxindex);

clear i

slopemaj=(ymaj-yptmax)/(xmaj-xptmax);
bmaj=((xmaj*yptmax)-(xptmax*ymaj))/(xmaj-xptmax);

slopemin=(ymin-yptmax)/(xmin-xptmax);
bmin=((xmin*yptmax)-(xptmax*ymin))/(xmin-xptmax);

xaxismin=min(db6x)-50;
xaxismax=max(db6x)+50;
yaxismin=min(db6y)-5;
yaxismax=max(db6y)+5;

xline=[xaxismin:xaxismax];
m=length(xline);

for (i=1:m)
     ymajline(i)=(slopemaj*xline(i))+bmaj;
     yminline(i)=(slopemin*xline(i))+bmin;
end

plot(xline,ymajline,'r')
plot(xline,yminline,'c')
plot(xptmax,yptmax,'o')
axis([xaxismin xaxismax yaxismin yaxismax])

maja1=input('input the x coordinate of the 1st point on the major axis: ');
majb1=input('input the y coordinate of the 1st point on the major axis: ');
maja2=input('input the x coordinate of the 2nd point on the major axis: ');
majb2=input('input the y coordinate of the 2nd point on the major axis: ');

mina1=input('input the x coordinate of the 1st point on the minor axis: ');
minb1=input('input the y coordinate of the 1st point on the minor axis: ');
mina2=input('input the x coordinate of the 2nd point on the minor axis: ');
minb2=input('input the y coordinate of the 2nd point on the minor axis: ');

maja=(maja1-maja2)*stepsize2/1000;      % in mm
majb=(majb1-majb2)*stepsize1/1000;
mina=(mina1-mina2)*stepsize2/1000;
minb=(minb1-minb2)*stepsize1/1000;

beam_width=sqrt((mina^2)+(minb^2))
depth_of_focus=sqrt((maja^2)+(majb^2))

%
*************************************************************************
%
% foclength.m
% finds the focal length of a transducer
%
%
*************************************************************************
```

```
% sos = speed of sound (m/s)
% t = temperature of water (Celcius)
% xzero = time to first waveform

abstwave=abs(twave);
abstwavemax=max(abstwave);

clear i
for (i=1:number_point)
     if (abstwave(i) == abstwavemax)
         maxtwavepos=i;
     end
end

total_shift=xptmax*timeshift;                          % in seconds

if (startpos == 'c')
    totaltime=xzero+total_shift+(xincr*maxtwavepos);    % adds time shift
else
    totaltime=xzero-total_shift+(xincr*maxtwavepos);    % subtracts time
shift
end

focal_length=(totaltime/2)*sos*1000                     % in mm


% Adding session capture to a file.

%sessionlog = fopen('session.log','w');
%fprintf(sessionlog,'1st -3 dB = %4.2f\r',f1);
%fprintf(sessionlog,'2nd -3dB = %4.2f\r',f2);
%fprintf(sessionlog,'Center frequency = %6.2f\r',center_frequency);
%fprintf(sessionlog,'Bandwidth = %6.2f\r',bandwidth);
%fprintf(sessionlog,'Fractional Bandwidth = %7.2f\r',fractional_bandwidth);
%fprintf(sessionlog,'Wavelength = %6.2f\r',wavelength);
%fprintf(sessionlog,'Pulse duration = %6.2f\r',pulse_duration);
%fprintf(sessionlog,'-6 dB Beam width = %6.2f\r',beam_width);
%fprintf(sessionlog,'-6 dB Focus depth = %6.2f\r',depth_of_focus);
%fprintf(sessionlog,'Focal length = %5.2\r',focal_length);
%fclose (session-log);


%
**************************************************************************
%
% THE END
%
%
**************************************************************************
```