APPENDIX C

C.1 Marconi.m Script

```
% marconi.m
% computes PII, Pc, Pr and plots versus axial diastance,
% given beam axis measurements of a transducer made using a hydrophone
% this script loads both the bin and dat file
%
% written by Adam Wunderlich, Sept. 1998
% using complete5.m as a basis
% modified Dec. 1998 by Jason Sempsrott
% modified Aug. 1999 by Jason Sempsrott
% modified March 2000 by Bill Zierfuss
% modified April 2000 by Bill Zierfuss

clear all;
close all;

n      = input('Enter a filename: ','s');


% **************  IMPORTANT **************************
%
% BE SURE TO CHANGE THESE BEFORE PROCESSING THE DATA FROM EACH TRANSDUCER
%
%
%fc   = input('Enter the center frequency of the transdcuer in MHz ');
fc= 3.03;

%dist= input('Distance of hydrophone from transducer at start  (in cm)');
dist=23.7346;
%
% ******************************************************

R = input('Enter Ritec setting: ');


%date = input('Date of calibration: ','s');
%date='06-19-01';

%whom = input('Person calibrating: ','s');
whom='Stacie Sakai';

%which      = input('Hydrophone that was used (type and Serial #):\n ','s');
which='Marconi EW295';

%factor     = input('Enter the calibration factor in V/MPa: ');
factor=.043;


bin = [n '.bin'];
dat = [n '.dat'];
```

```
fid   = fopen(dat, 'r');
A     = fscanf(fid,'%f',[12,1]);

ymult=A(1,1);            % y-scaling
yzero=A(2,1);            % DC offset
xincr=A(3,1);            % x-scaling, 1/(sampling rate)
number_point=A(4,1);     % # points in waveform
numscan1=A(5,1);  % # of scans for scan axis1
numscan2=A(6,1);  % # of scans for scan axis2
xzero=A(7,1);            % time for start of first wave form
stepsize1=A(8,1); % step size for scan axis 1
stepsize2=A(9,1); % stepsize for scan axis 2
sos=A(10,1);             % speed of sound
timeshift=A(11,1);       % window time shift
numscan3=A(12,1); % number of scan points for beam axis

clear A;

% open and read .bin file
% assume LeCroy oscilloscope was used to take data

status = fclose(fid);
fid=fopen(bin,'rb','b');  % 'b' for LeCroy



%//////////////////////////////////////////////////////////////////////////
% compute PII
%//////////////////////////////////////////////////////////////////////////
PIItemp1=0;
PII=zeros(1,numscan3);
    for(j=1:numscan3)
        B=fread(fid,number_point,'short');  % B is a column vector
        B=B.*ymult;
        B=B-yzero;      % - for LeCroy
        PII(1,j)=xincr*sum(B .* B);
    end

x1 = ((0:numscan3-1)*stepsize1);     % calculates axis for axial distance
x1 = x1/10000;  % convert to cm from um
x = dist + x1;


figure
orient tall

% plot PII versus axial distance
subplot(3,1,1)
% want to apply smoothing function to curve
pPII  = polyfit(x,PII,6);
fPII  = polyval(pPII,x);
plot(x,fPII)
xlabel('Axial distance (mm)')
ylabel('PII')
grid
status = fclose(fid);
[max_fPII,x_fPII] = max(fPII);
```

```
x_fPII=dist+(x_fPII*stepsize1/10000); % axial distance of maximum Pc in cm

% also need derated water value for PII and position of maximum
PII3=fPII.*exp(-.069*fc*x);
[max_PII3,junk] = max(PII3);
x_PII3 = dist+(junk*stepsize1/10000); % axial position of max PII3


%///////////////////////////////////////////////////////////////////////////
%      relevent variables are:
%      x      = axial position from transducer
%      PII    = raw data obtained during scan
%      fPII   = smooth curve of PII
%      PII3   = derated PII values from smooth curve
%      x_PII3  = position of maximum PII3
%////////////////////////////////////////////////////////////////////////////

%////////////////////////////////////////////////////////////////////////////
% plot peak compressional pressure versus axial distance
%////////////////////////////////////////////////////////////////////////////

fid=fopen(bin,'rb','b');    % b for Lecroy

Pc=zeros(1,numscan3);
    for(j=1:numscan3)
        C=fread(fid,number_point,'short');  % C is a column vector
        C=C.*ymult - yzero;  % - for LeCroy
        Pc(1,j) = max(C);
    end

% Need to convert voltages to pressures
Pc=Pc/(factor);

subplot(3,1,2)
pPc   = polyfit(x,Pc,6);
fPc   = polyval(pPc,x);
plot(x,fPc)
xlabel('Axial distance (cm)')
ylabel('Pc (MPa)')
grid
status = fclose(fid);

%      need to find maximum Pc value and its position

[max_Pc,x_Pc]=max(fPc);
x_Pc=dist+(x_Pc*stepsize1/10000); % axial distance of maximum Pc in cm

% derated water value for Pc

Pc3   = fPc.*exp(-.0345*fc*x);

%      max value of Pc3 and its position
[max_Pc3,x_Pc3]=max(Pc3);
x_Pc3=dist+(x_Pc3*stepsize1/10000); % axial distance of maximum Pc3

Pc3_max_PII3=Pc3(1,junk); % derated Pc at the position of max PII3
```

```
%///////////////////////////////////////////////////////////////////////////
%     relevant variables include:
%           x     = axial distance
%           Pc    = original max compressional values
%           fPc   = smooth curve of Pc
%           x_Pc  = position of maximum Pc (cm)
%           max_Pc      = value of maximum Pc (MPa)
%           Pc3   = derated water value for fPc (MPa)
%           x_Pc3 = position of maximum Pc3 (cm)
%           max_Pc3 = value of maximum Pc3 (MPa)
%           Pc3_max_PII3 = value of Pc3 located at max_PII3
%/////////////////////////////////////////////////////////////////////

%///////////////////////////////////////////////////////////////////////////
% plot peak rarefractional pressure versus axial distance
%///////////////////////////////////////////////////////////////////////////

fid=fopen(bin,'rb','b');    % b for Lecroy

Pr=zeros(1,numscan3);
    for(j=1:numscan3)
        D=fread(fid,number_point,'short');  % D is a column vector
        D=D.*ymult - yzero;  % - for LeCroy
        Pr(1,j) = abs(min(D));
    end

% Need to convert voltages to pressures
Pr=Pr/(factor);

subplot(3,1,3)
pPr=polyfit(x,Pr,6);
fPr=polyval(pPr,x);
plot(x,fPr)
xlabel('Axial distance (cm)')
ylabel('Pr (MPa)')
grid
status = fclose(fid);

%     need to find maximum Pr value and its position

[max_Pr,x_Pr]=max(fPr);
x_Pr=dist+(x_Pr*stepsize1/10000); % axial distance of maximum Pr


%     derated water value for Pr
Pr3   = fPr.*exp(-.0345*fc*x);
%     max value of Pr3 and its position
[max_Pr3,x_Pr3]=max(Pr3);
x_Pr3=dist+(x_Pr3*stepsize1/10000); % axial distance of maximum Pr3

Pr3_max_PII3=Pr3(1,junk); % derated Pr at the position of max_PII3

MI = Pr3_max_PII3/(fc^.5);

%/////////////////////////////////////////////////////////////////////
%     relevant variables include:
```

```
%           x     = axial distance (cm)
%           Pr    = original max rarefactional values (MPa)
%           fPr   = smooth curve of Pr (MPa)
%           x_Pr  = position of maximum Pr (cm)
%           max_Pr      = value of maximum Pr (MPa)
%           Pr3   = derated water value for fPr (MPa)
%           x_Pr3 = position of maximum Pr3 (cm)
%           max_Pr3 = value of maximum Pr3 (MPa)
%        Pr3_max_PII3 = value of Pr3 at position of max_PII3
%           MI    = mechanical index
%///////////////////////////////////////////////////////////////////////

%///////////////////////////////////////////////////////////////////////
% plot the time domain waveform at the maximum point in the PII plot
% (plots waveform at focus)
%///////////////////////////////////////////////////////////////////////

fid=fopen(bin,'rb','b');   % b for Lecroy

[max_PII,x_PII]=max(PII);     % finds axial location of focus
junky=dist+(x_PII*stepsize1/10000); % axial distance of maximum Pc in cm

temp=0;
for(j=1:x_PII-1)
    [wave]=fread(fid,number_point,'short'); % wave is col. vector

end




% check for zero points at end of data set, and count them
z=0;
for (i=1:number_point)
    if (wave(i,1) == 0)
        z = z+1;
    end
end


% plot the waveform at the focus centered around 0 V
figure

g = number_point-z;      % g is an intermediate variable

%twave=wave-mean(wave);
wavescaled=wave.*ymult - yzero;

% Want to develop the waveform for PII vs. Time
simple=0;
for(j=1:number_point)
      tempwave=wave(j,1).*ymult;
      tempwave1=tempwave.*tempwave;
      tempwave2=tempwave1+simple;
      simple=tempwave2;
      newwave(j,1)=tempwave2;
end
```

```
%twavescaled= wkeep(wavescaled,g,'l');  % truncate to get rid of zeros

timeaxis=(0:number_point-1)*xincr*1e6;
plot(timeaxis,wavescaled)
title('Waveform at Focus')
xlabel('Time (microseconds)')
ylabel('Amplitude (V)')
grid
focuspc=max(wavescaled);   %This gives the pc voltage value at the focus.
focuspc=focuspc/(factor);      % Converted volts to MPa
focuspr=abs(min(wavescaled));   %This gives the pr voltage value at the
focus.
focuspr=focuspr/(factor);      % Converted volts to MPa


%///////////////////////////////////////////////////////////////////////////
% Code to determine the frequency of the signal
%
% wavefft=abs(fft(wavescaled(1:8192)));
% freq=(0:8191)/8192/xincr;
% plot(freq,wavefft);
% [a,b]=max(wavefft);
% freq(b)
%
%///////////////////////////////////////////////////////////////////////////




%///////////////////////////////////////////////////////////////////////////
%            relevant variables include:
%                        junky   = position of max_PII
%                        focuspc = Pc located at max_PII
%                        focuspr = Pr located at max_PII
%///////////////////////////////////////////////////////////////////////////

%//////////////////////////////////////////////////////////////////////////
% A plot of PII vs Time
%//////////////////////////////////////////////////////////////////////////
figure
plot(timeaxis,newwave);
xlabel('Time (microseconds)');
ylabel('PII Value');
grid

% Need to find limits for calculating the pulse duration
% This calculation uses the 90% and 10% points of the PII curve
% First find the upper and lower limits according to FIgure 3

upperlimit=max(newwave)*.9;
lowerlimit=max(newwave)*.1;

% Second, find the time value at the upper and lower limits
```

```
for(j=1:number_point)
      if(newwave(j,1) > (upperlimit))
                  timeaxisupper=timeaxis(1,j-1);
                  break;
      end
end
for(j=1:number_point)
      if(newwave(j,1) > (lowerlimit))
                  timeaxislower=timeaxis(1,j-1);
                  break;
      end
end

% Third, calculate a time for the pulse duration
pulse=(timeaxisupper-timeaxislower)*1.25;

%/////////////////////////////////////////////////////////////////////////////



rawdata           = [x;PII;Pc;Pr];
smoothdata  = [x;fPII;PII3;fPc;Pc3;fPr;Pr3];
interests1  = [max_Pc;x_Pc;max_Pr;x_Pr];
interests2  = [max_Pc3;x_Pc3;max_Pr3;x_Pr3];
interests3  = [junky;focuspc;focuspr];
interests4  = [x_PII3;Pc3_max_PII3;Pr3_max_PII3];
interests5  = [MI;pulse;R];
everything  =
[R;pulse;max_Pc;x_Pc;max_Pr;x_Pr;max_Pc3;x_Pc3;max_Pr3;x_Pr3;junky;focuspc;fo
cuspr;x_PII3;Pc3_max_PII3;Pr3_max_PII3;MI];


%n=input('Enter filename to save all data: ','s');

if (R < 1)
     n = [num2str(R,2)];
     n1 = [num2str(R,2)];
     name = [num2str(R,2),'-1.jpg'];
elseif (R < 10)
     n = ['0',int2str(R)];
     n1 = ['0',int2str(R)];
     name = ['0',int2str(R),'-1.jpg'];
else
     n = [int2str(R)];
     n1 = [int2str(R)];
     name = [int2str(R),'-1.jpg'];
end


txtfile=[n,'-data.txt'];
fid=fopen(txtfile,'a');
fprintf(fid,'\n/////\n');
fprintf(fid,'Raw\n');
fprintf(fid,'x(cm)\t PII\t Pc(MPa)\t Pr(MPa)\n');
fprintf(fid,'%1.4f\t %2.4e\t %2.4f\t  %2.4f\n',rawdata);
fprintf(fid,'/////\n');
```

```
fprintf(fid,'Best Fit \n');
fprintf(fid,'x(cm)\t PII\t PII.3\t
Pc(MPa)\tPc.3(MPa)\tPr(MPa)\tPr.3(MPa)\n');
fprintf(fid,'%1.4f\t %2.4e\t %2.4e\t
%2.4f\t%2.4f\t%2.4f\t%2.4f\n',smoothdata);
fprintf(fid,'/////\n');
fprintf(fid,'Points of Interest\n');
fprintf(fid,'\nGlobal Information from Best Fit Data:\n');
fprintf(fid,'\nMax Pc(MPa)\tPosition(cm)\t Max Pr(MPa)\tPosition(cm)\n');
fprintf(fid,'%2.4f\t\t %1.4f\t\t %2.4f\t %1.4f\n\n',interests1);
fprintf(fid,'Max Pc.3(MPa)\tPosition(cm)\t Max Pr.3(MPa)\tPosition(cm)\n');
fprintf(fid,'%2.4f\t\t %1.4f\t\t %2.4f\t\t %1.4f\n\n',interests2);
fprintf(fid,'Information from waveform at focus (max of PII):\n');
fprintf(fid,'Position of max PII\t Pc (MPa)\t\t Pr (MPa)\n');
fprintf(fid,'%1.4f\t\t\t%2.4f\t\t\t%2.4f\n\n',interests3);
fprintf(fid,'Position, Pc.3, and Pr.3 at maximum of PII.3:\n');
fprintf(fid,'Position of max PII.3\t Pc.3 (MPa)\t\t Pr.3 (MPa)\n');
fprintf(fid,'%1.4f\t\t\t%2.4f\t\t\t%2.4f\n\n',interests4);
fprintf(fid,'Mechanical Index\t Pulse Duration (microsecs)\t RITEC
Setting\n');
fprintf(fid,'%1.4f\t\t\t%1.4f\t\t\t\t%1.2f\n\n',interests5);
fprintf(fid,'Date of calibration:\n');
fprintf(fid,date);
fprintf(fid,'\nPerson who calibrated:\n');
fprintf(fid,whom);
fprintf(fid,'\nHydrophone that was used (type and serial #):\n');
fprintf(fid,which);
fprintf(fid,'\nHydrophone conversion factor is (V/MPa):\n');
fprintf(fid,'%1.4f',factor);
fprintf(fid,'\nTransducer frequency(MHz)   %1.4f\t',fc);
fprintf(fid,'\n/////\n');

status=fclose(fid);

%n1=input('Enter filename for points of interest only: ','s');


txtfile1=[n1,'-intdata.txt'];
fid=fopen(txtfile1,'a');
fprintf(fid,'%1.2f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%
2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\t%2.4f\n',everything);

status=fclose(fid)

disp('Max Pc in MPa is:');
max_Pc
disp('Max Pr in MPa is:');
max_Pr


%name = [int2str(R),'-1.jpg'];
figure(1);
eval (['print -djpeg ',name]);

% To plot the 6th degree polyfit coefficients, use: `pPr' at the
% Matlab prompt.  The polynomial is of the form:
%
```

```
%     y =  x^n + x^(n-1) + ... + x^1 + x^0
%
% For example, from a run of the program...
%
% >> pPr
%
% pPr =
%
%   1.0e+05 *
%
%    0.0000   -0.0011    0.0233   -0.2318    1.2176   -3.2768    3.5726
%
% In this example, the 6th degree polynomial would expand to:
%
%   1.0e05 [ 0x^0 + (-0.0011)x^5 + ... + (-3.2768)x^1 + (3.5726)x^0
%

disp('The data was written to the file:');
disp(txtfile);
disp(' ');
disp('and the interesting data was written to the file:');
disp(txtfile1);
disp(' ');
disp('A jpeg image of figure 1 was written to the file:');
disp(name);
disp(' ');
disp('*******************************************************************');
      disp(' ');
```