

Senior Thesis

Use of Microwave Radar Data to Interpret Physical  
Properties of Ice and Snow on the  
Greenland Ice Sheet

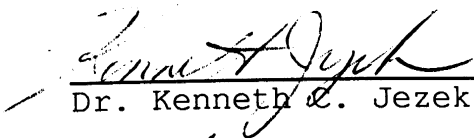
by

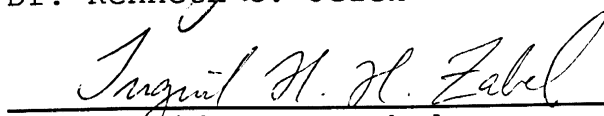
Christopher M. Gero

1995

Submitted as partial fulfillment of  
the requirements for the degree of  
Bachelor of Science in Geology  
at The Ohio State University,  
Spring Quarter, 1995

Approved By:

  
\_\_\_\_\_  
Dr. Kenneth C. Jezek

  
\_\_\_\_\_  
Dr. Ingrid H. H. Zabel

## Table of Contents

1.0 Introduction.....	2
2.0 Approach.....	2
3.0 Equipment.....	4
4.0 Data.....	5
5.0 Analysis.....	7
6.0 Results.....	10
7.0 Summary.....	16
8.0 References.....	17
Appendix I.....	19

### Abstract

In situ observations of microwave radar backscatter were made at Dye-2 in south-central Greenland in 1993. Simultaneous observations of snow and ice physical properties were also acquired. Early in the season, it was found that 20° backscatter from the snow surface was determined mainly by the snow surface roughness. Later in the season the backscatter increases when temperatures are low. We attribute this fact to enhanced scattering from the ice within the radar resolution cell at the surface. This ice formed during the field season as a result of percolation and refreezing of surface meltwater. These observations have a direct implication for interpreting spaceborne SAR data acquired seasonally.

## 1.0 Introduction

The goals of this project are to measure in situ radar signatures of near-surface snow and ice stratigraphy of the Greenland ice sheet and to relate these signatures to the physical properties of the snow. A long term goal is to use this information to interpret spaceborne synthetic aperture radar and radar altimeter data collected over the polar ice sheet.

The objectives of this paper are first, to analyze high-resolution radar data collected on the western flank of the Greenland ice sheet and second, to compare these data with accurate and repeated measurements of near surface snow stratigraphy. These data are unique because of the exceptional range resolution of the radar used and because coincident time series of radar backscatter and ice physical properties were obtained.

This work extends earlier work by Jezek, Gogineni, and Shanableh (1994), who observed and interpreted radar backscatter strengths in early spring from different layers in the firn stratigraphy. Analysis presented here complements work by Fahnestock et al. (1993) who used ERS-1 SAR imagery to reveal the basic zonation of the Greenland ice sheet, that is, the dry snow, percolation, wet snow, and bare ice facies.

## 2.0 Approach

All data used were gathered by I.H.H. Zabel, P.A. Baggeroer, and K.C. Jezek of the Byrd Polar Research Center between June 23

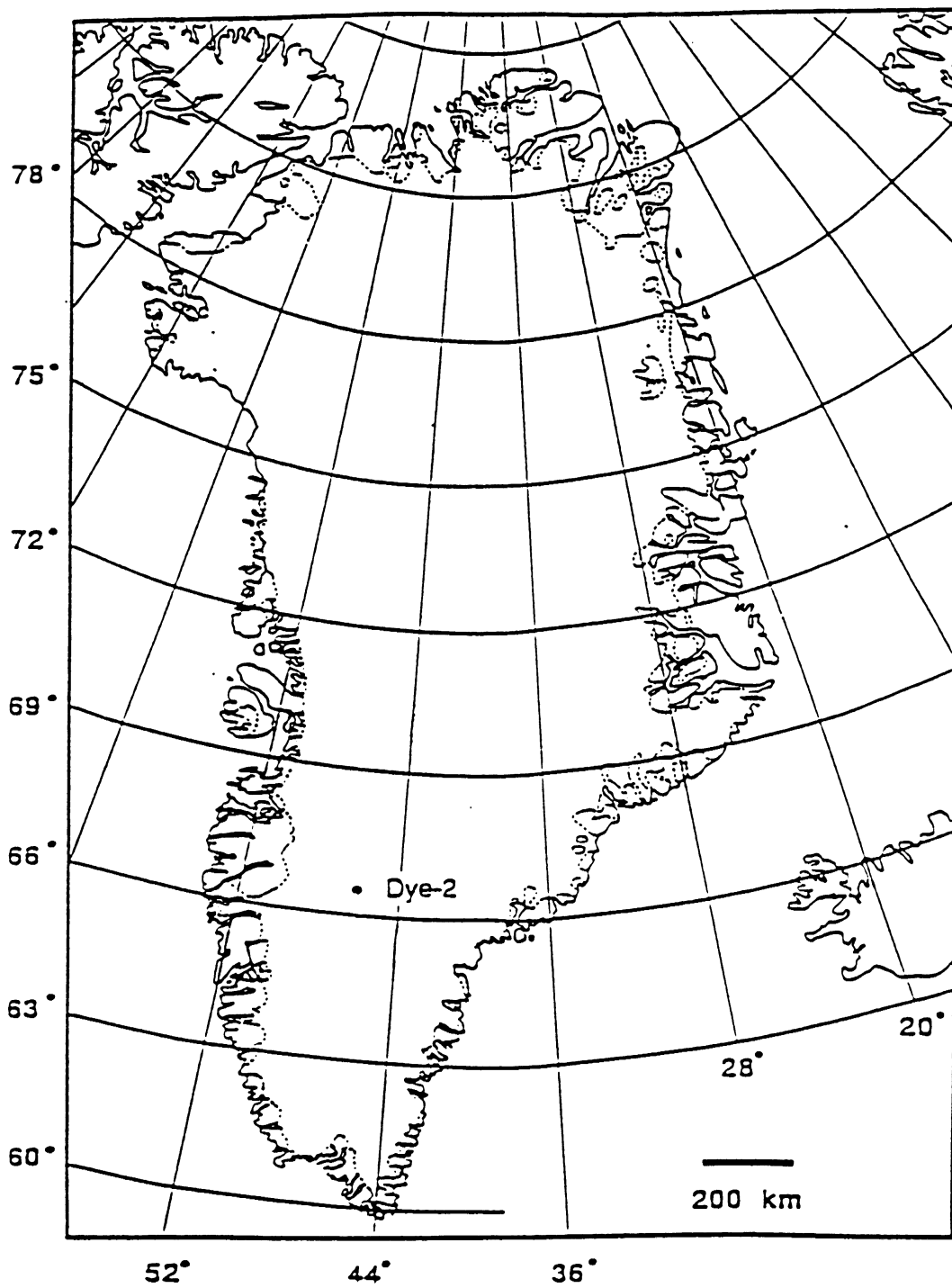


Figure 1: Map of Greenland showing location of the Dye-2 field station.

and July 8, of the 1993 Field season at Dye-2 (see fig.1). The work was conducted in conjunction with the University of Kansas.

The occupied site was in the percolation zone of the ice sheet which lies between the dry-snow facies and the wet-snow facies. Some seasonal melting occurs in the percolation zone, but water redistribution is limited horizontally and vertically. Meltwater percolates down through the snow and refreezes to form ice pipes and massive ice lenses. The percolation zone experiences strong seasonal variations in snow grain size (Jezek, Gogineni, and Shanableh, 1994).

### 3.0 Equipment

The radar, which has a separate transmitting and receiving horn, has a 2 GHz bandwidth centered around 13.5 Ghz. These system parameters result in an 11 cm resolution in the snow. A continuous, frequency modulated wave (FMCW) is emitted with an effective beamwidth of  $10^\circ$ . The radar mounts on a sled, with the transmitting and receiving horns about 1.5 meters above the snow surface. This results in an illuminated area of about  $0.054 \text{ m}^2$  at normal incidence.

Traditional techniques were used to gather data on the physical properties of the ice and snow. Equipment used included density tubes, photographic equipment, magnifiers and gratitudes for measuring grain size, and comb-gauges for measuring ice-layer roughness.

#### 4.0 Data

The raw radar data were received power as a function of time, incidence angle, and polarization, measured relative to the transmitted power. The data were converted into scattering cross-sections, based on calibration measurements done using a Luneberg lens of known scattering cross-section ( $\sigma_{cal}$ ). The formula used to calculate the scattering cross-section of the snow from a radar measurement is

$$\sigma(R) = (P_r(R)/P_{rcal}) (P_{tcal}/P_t) (R/R_{cal})^4 \sigma_{cal},$$

where  $\sigma$  is the scattering cross-section (unknown),  $P_t$  and  $P_r(R)$  are the transmitted and received power, respectively, for the snow measurement, and  $P_{tcal}$  and  $P_{rcal}$  are the transmitted and received power, respectively, for the calibration experiment.  $R$  is the range to the snow surface,  $R_{cal}$  is the range to the Luneberg lens, and  $\sigma_{cal}$  is the scattering cross-section of the lens.

The radar data were converted to volume backscatter coefficients with the equation,

$$\sigma_{vol}^o(R) = (\sigma(R)/V(R)),$$

where  $\sigma_{vol}^o(R)$  is the volume backscattering coefficient,  $\sigma(R)$  is the scattering cross-section, and  $V$  is the volume illuminated by one range cell at range  $R$ .

Air temperatures at the time of the radar measurements (June 23 through July 8, 1993, usually between 8 AM and 8 PM) ranged from about +1° C to -6° C. Nighttime lows varied between about -5° and -9° C (Zabel, et al., 1994) (see fig. 2). The snow density to a depth of 2 meters varied between 0.3 and 0.5 g/cc. The snow

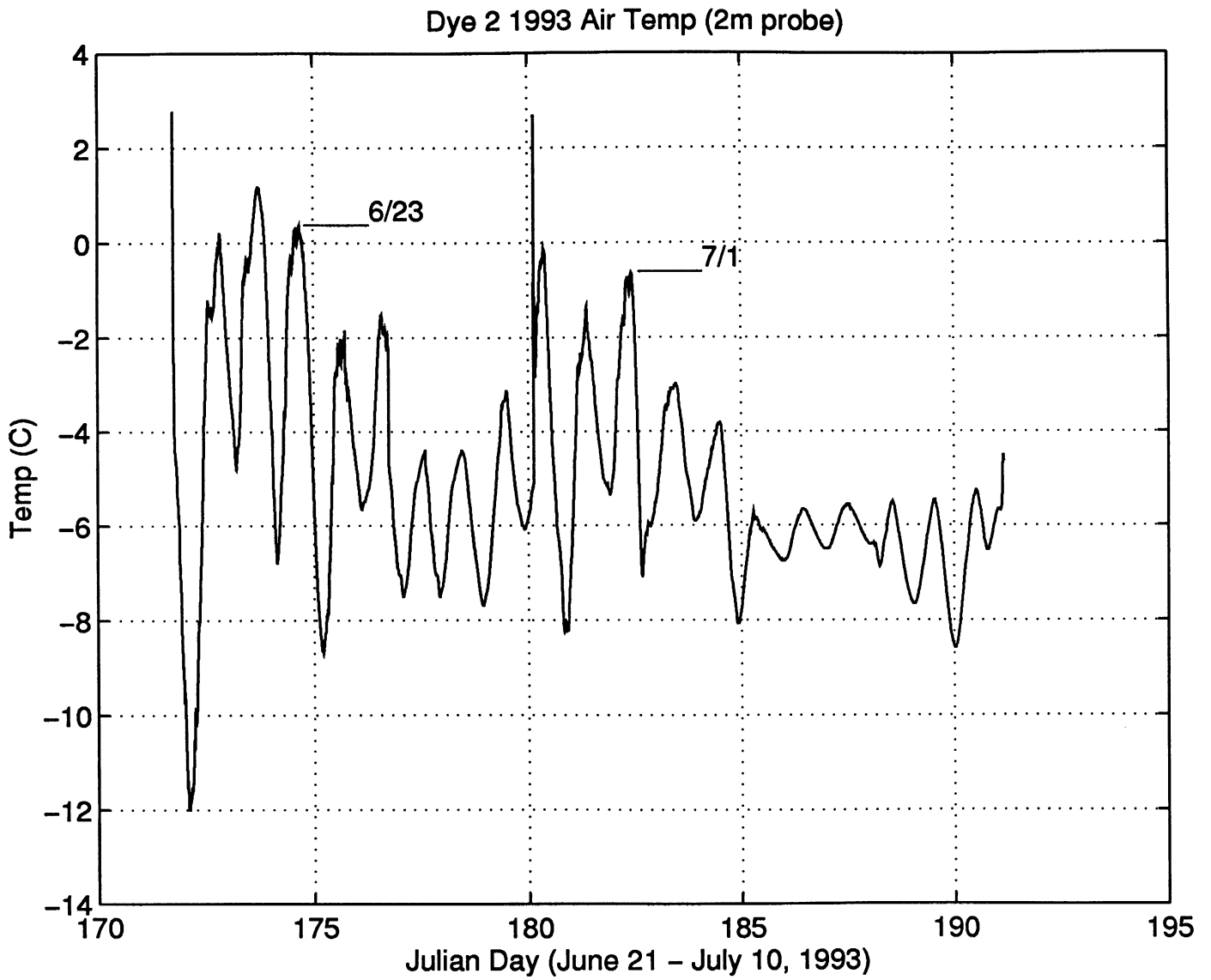


Figure 2: Air temperature record. Peak on 6/29 is an anomaly.

temperature ranged from about  $-2^{\circ}$  C at the surface to about  $-18^{\circ}$  C at 2 meters depth (Zabel, et al., 1994).

## 5.0 Analysis

"Sig.m" was the first of several MATLAB programs used to convert the raw data to volume backscattering coefficient as a function of range (see Appendix I). Samples of plots produced by sig.m using data gathered on June 23 and July 8, 1993 are shown in figures 3 and 4. Both plots show data taken at  $20^{\circ}$  incidence. The June 23 plot shows a very prominent peak ( $\sim 10$  dB higher than the nearby high peaks) at about 1.4 m. The July 8 plot is more ambiguous, with several high peaks close to the range to the snow surface.

After all the data were converted to the form shown in figures 3 and 4, they were entered into the program "test.m" (see Appendix I). Test.m creates a window extending .5 meters above and below a prescribed range-to-snow-surface value. This value is computed using the known height and pointing geometry of the radar. An initial way of choosing the snow surface correctly (most of the time) is to designate the highest peak within the window as being the backscatter from the snow surface. The highest peak, however, is not always from the surface. Often, shallow, subsurface ice layers give a stronger return than the snow surface. It was therefore necessary to go through all the test.m output plots and compare known range-to-surface values with the range of the chosen peak. In the cases where the wrong peak was chosen (as fig. 4



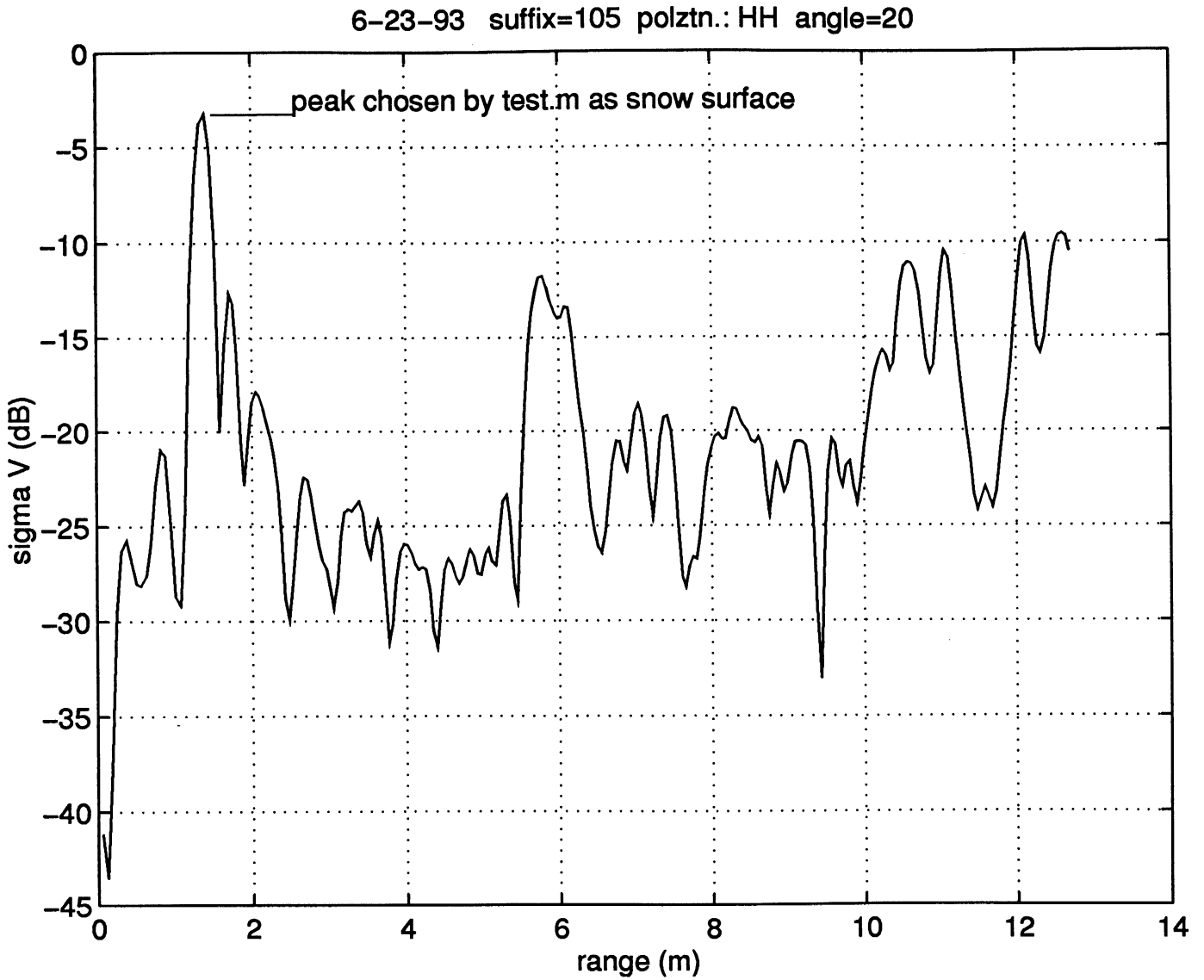


Figure 3: Output from the MATLAB program sig.m, showing the prominent peak in backscatter from the snow surface.

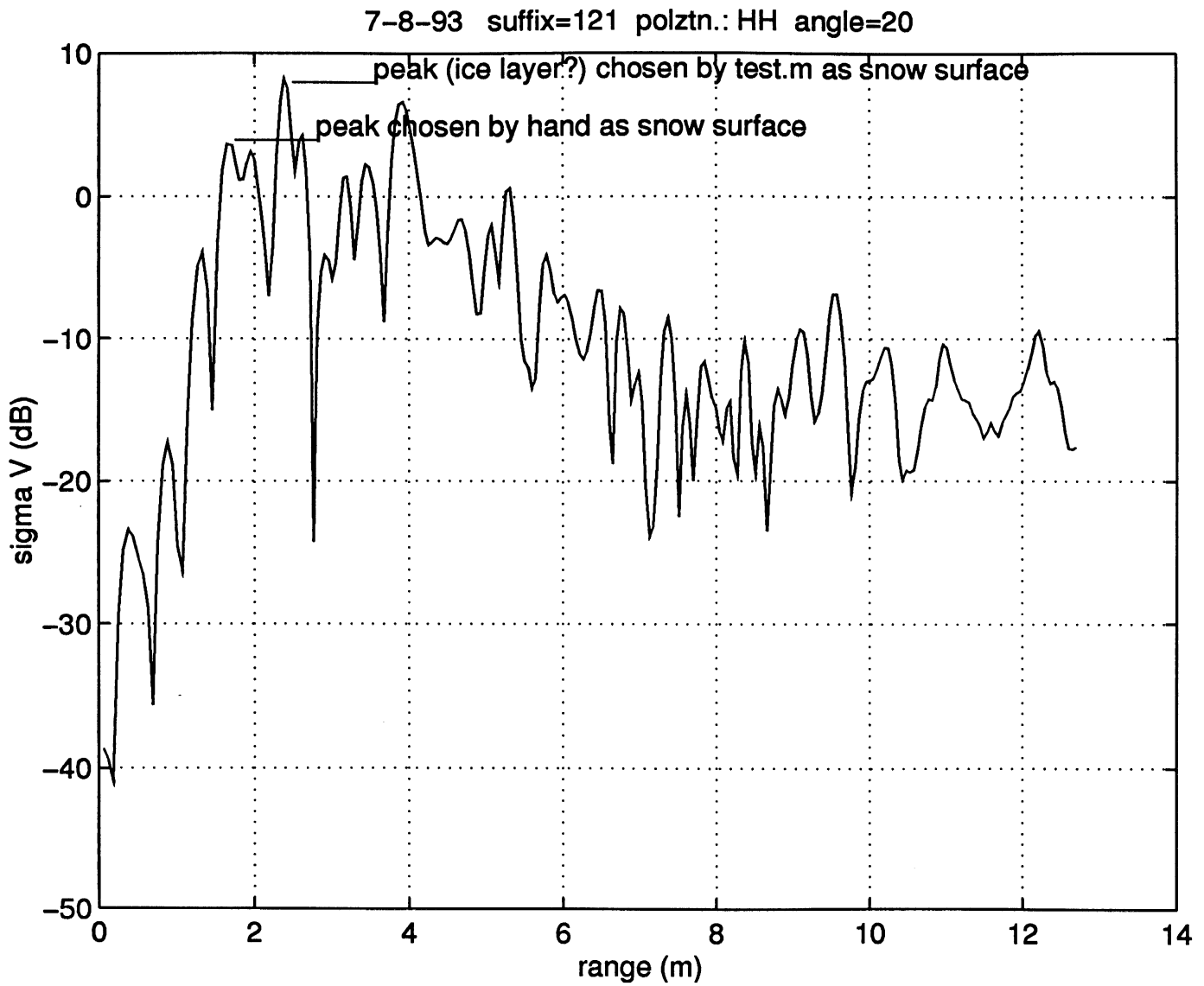


Figure 4: Output from the MATLAB program sig.m, showing how a shallow, subsurface ice layer can give a stronger return than the snow surface.

illustrates), the range and backscatter values for the peak thought to represent the actual snow surface were recorded in place of the incorrect values.

The output of test.m is a plot of the volume backscatter from the snow surface versus angle of incidence. As the incidence is increased from 0° to 60°, an expected decrease in backscatter occurs (see figs. 5 and 6).

As figures 5 and 6 show, the data are somewhat erratic. In order to smooth out the data and account for heterogeneities in the firn, an additional program, "avplot.m," was written (see Appendix I). Avplot.m averages the backscatter values of each of the angles in a set of measurements for two or more measurement locations. The output is average backscatter plotted against angle of incidence (see fig. 7). The almost 40 dB decrease in signal strength with incidence angle is indicative of rough surface scattering (Ulaby et al., 1982).

In addition to looking at the snow surface, test.m can also be used to get backscatter versus incidence angle data for a subsurface ice layer or from a specific depth (to about 11 m).

## 6.0 Results

After all the data had been run through test.m, a time series of  $\sigma_{vol}^{\circ}$  was constructed. The 20° incidence, HH-polarized backscatter from the snow surface from each set of measurements and the corresponding temperature were plotted against time (see fig. 8). Note that these values are for single samples, i.e., they have

6-30-93 suffix=100-109 polztn.: HH

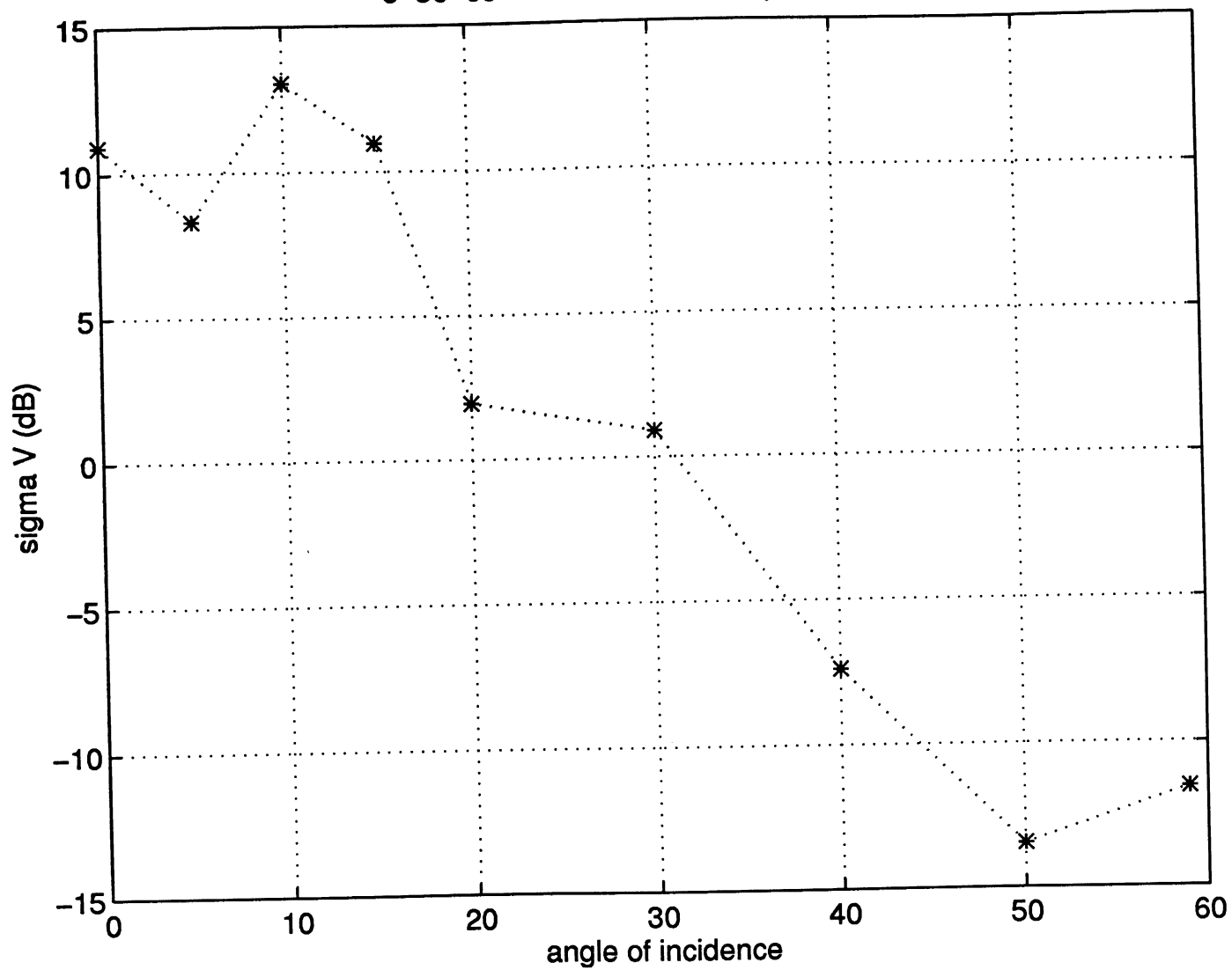


Figure 5: Output plot from test.m with HH polarization.

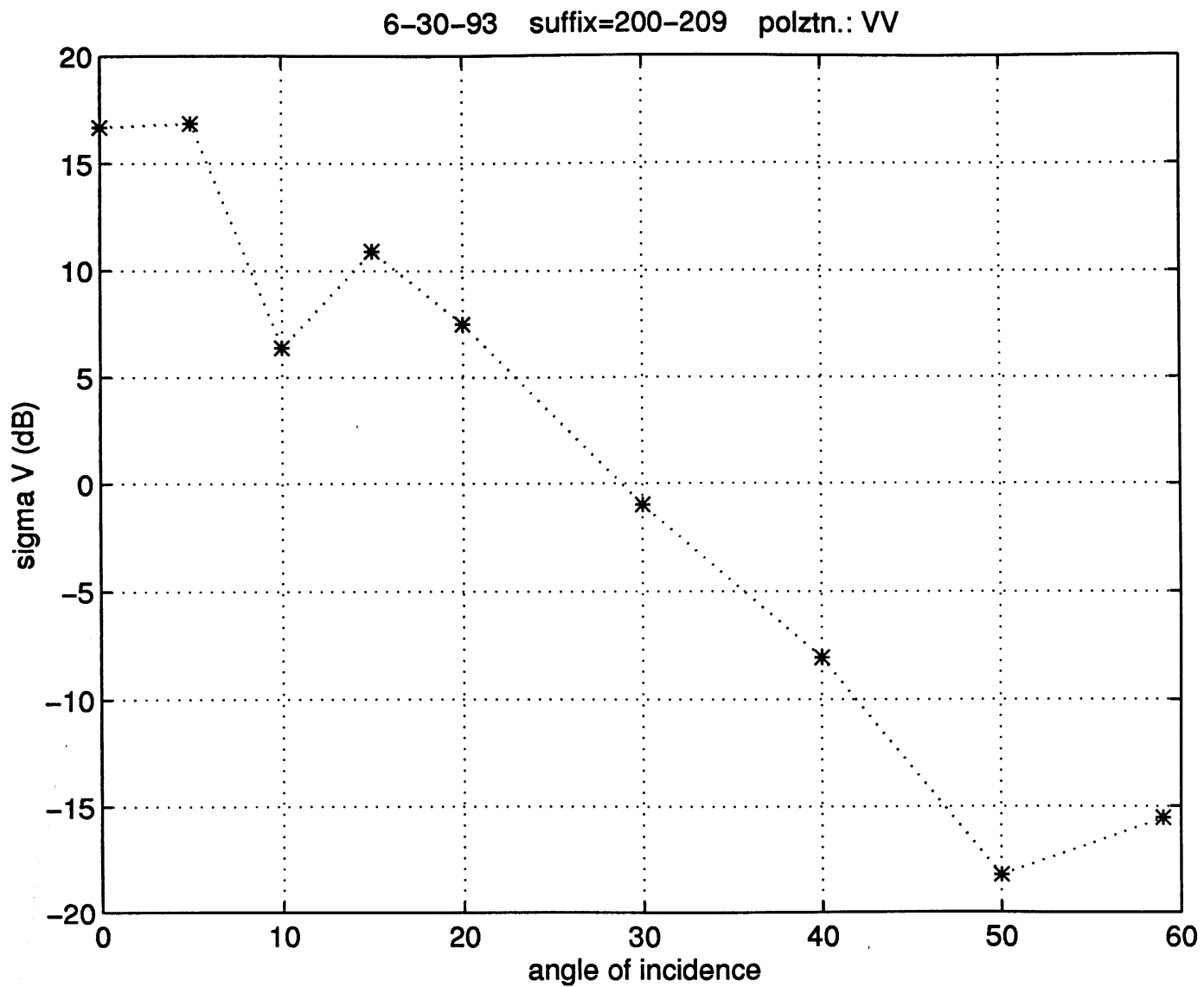


Figure 6: Output plot from test.m with VV polarization.

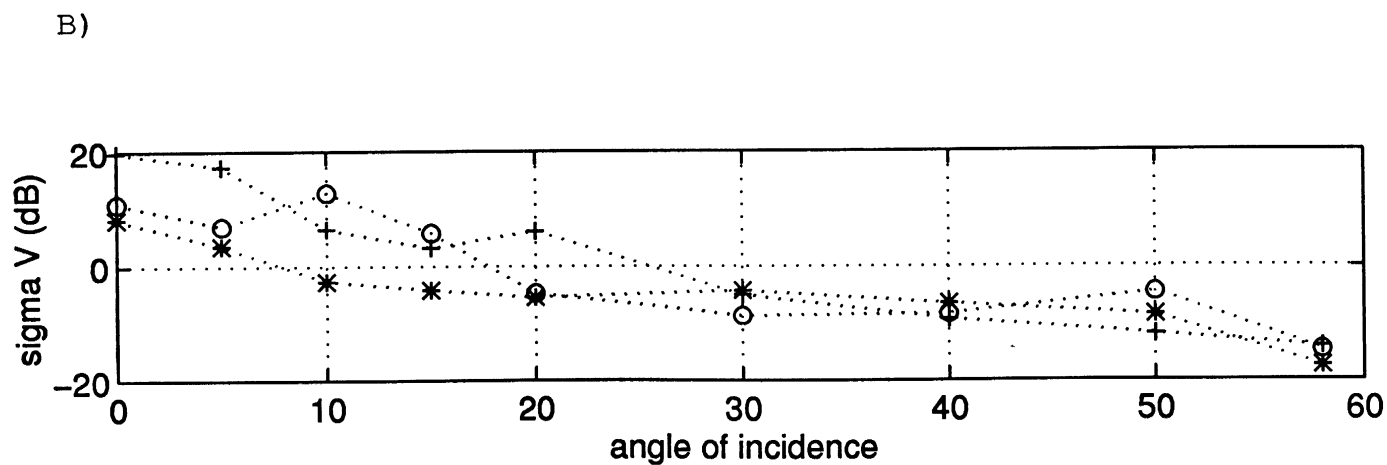
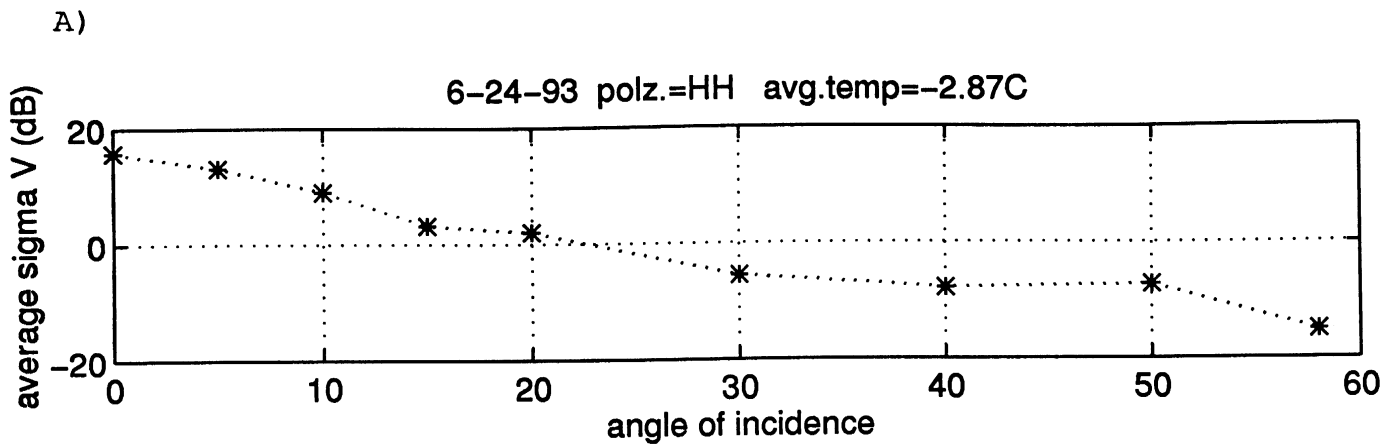


Figure 7: A) Output plot from the MATLAB program avplot.m.

B) Plot of the 3 measurement sets before averaging.

timeseries of backscatter & temperature polarization = HH

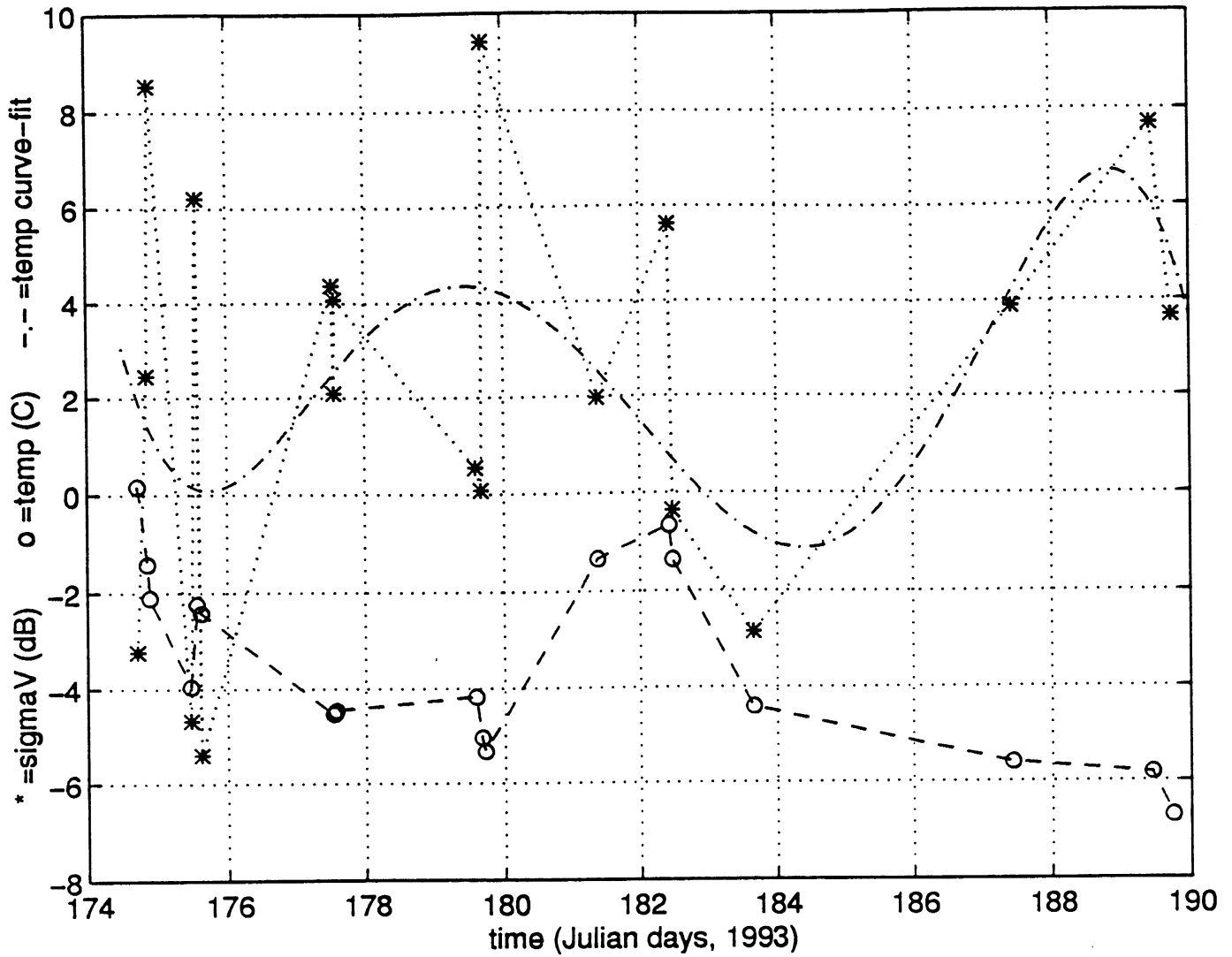


Figure 8: Julian day 174 = June 23<sup>rd</sup>. Backscatter data points for 20° incidence are marked with asterisks (\*) and are connected by a dotted line. Air temperatures taken concurrently with the radar data are marked with circles (o) and connected with a dashed line. A fourth degree polynomial curve has been fitted to the backscatter data to smooth it somewhat, shown as a dot-dashed line (-.-.).

not been averaged over several spots. The data in figure 8 are rather erratic, but this is because each point represents a single radar observation. We choose to plot individual observations in order to improve our time resolution.

There seems to be a weak relationship between  $\sigma_{vol}^{\circ}$  of the snow surface and T during the early and middle part of the record. The first three data points show the 20° backscatter from the snow surface increasing while the temperature decreases. One would expect the opposite to be true if the backscatter were determined solely by dielectric effects: when the temperature decreases below 0° C and liquid water freezes, the dielectric contrast between the snow surface and the air decreases and so backscattering should decrease (Ulaby et al., 1982). That this is not observed indicates that changes in surface roughness also come into play. The initial increasing backscatter with decreasing temperature most likely indicates an increasing surface roughness during the time sequence of the first three data points. This changing surface roughness could be due to wind action, since the winds varied in direction and strength at Dye-2.

From days 176 to 180 the temperature remains low relative to the initial warm period. The snow surface becomes moderate to high and the fluctuations decrease relative to the first six data points. This would seem to indicate a relatively steady and large surface roughness. Another physical mechanism may also lead to this high backscatter, however. Ice layers and lenses within 15 cm of the snow surface were observed both in pits and with the radar



following the initial warm period. These ice inclusions presumably formed from percolation and refreezing of meltwater. They are among the first of the season's ice layers. Since the resolution of the radar in snow was 11 cm, the radar measurement from the range cell of the snow surface probably includes these ice inclusions, which are known from previous work (Jezek, Gogineni, and Shanableh, 1994; Swift et. al, 1985) to be strong scatterers.

Around day 182 another warming to near 0° C occurs, and the backscatter drops. Presumably some melting occurred and penetration was limited to within a few centimeters of the snow surface, masking the ice inclusions only slightly deeper down. If the surface was fairly smooth at this time, the backscatter at 20° incidence would be relatively low, as observed. Then after day 184 the temperatures are low and the backscatter is high again as the radar beam once again penetrates and scatters strongly from ice inclusions within about 10 cm of the snow surface.

## 7.0 Summary

A series of microwave-wavelength radar measurements were taken on the Greenland ice sheet, in conjunction with precise, accompanying surface physical properties measurements. These measurements show the snow surface backscatter being determined initially mainly by the snow surface roughness. Later in the season after an ice layer has begun to form in the near-surface due to refreezing of meltwater, the surface return is dominated by scattering from this layer. The strong contribution of this layer

(and also from the previous summer's ice layer) has direct implications on the accuracy of satellite radar altimetry data. These results are consistent with earlier work by Jezek et al. (1994).

## 8.0 References

Jezek, K. C., P. Gogineni, and M. Shanableh, Radar Measurements of the Melt Zones on the Greenland Ice Sheet, *Geophys.Res.Lett.*, 21, 1, 33-36, 1994.

Fahnestock, M., R. Bindshadler, R. Kwok, and K. Jezek, Greenland Ice Sheet Surface Properties and Ice Dynamics from ERS-1 SAR Imagery, *Science*, 262, 1530-1534, Dec., 1993.

Zabel, I. H. H., P. A. Baggeroer, and K. C. Jezek, Radar and Snow Structure Studies in the Percolation Zone of the Greenland Ice Sheet: A Data Report on the 1993 Field Season at Dye-2, BPRC Technical Note 94-01, 1994.

Ulaby, F. T., R. K. Moore, and A. K. Fung, Microwave Remote Sensing, Volume II, Addison-Wesley, Reading, Mass, 1982, pp. 821-828.

Swift, C. T., P. S. Hayes, J. S. Herd, W. L. Jones, and V. E. Delnore, Airborne Microwave Measurements of the Southern Greenland Ice Sheet, *J. Geophys. Res.* 90, 1983-1994 (1985).

Appendix I: Printouts of programs used:

- 1) sig.m
- 2) test.m
- 3) avplot.m

```
%% SIG.M written by M. Shanableh and I. H. H. Zabel, 1993
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%This program takes the raw data and calculates power, calls the volume
%illuminated program, calculates the range and SigmaV. It also calculates
%travel times from and to the antenna.
%Average over the "N" number of files collected in one sample (i.e.,
average
%over the files HH,HV,VV, and VH, or however many were saved).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Note: correction for bistatic angle of luneberg lens: correction factor
% is  $\exp(-\ln(1/2) * (\phi/\theta_{3dB})^2)$  where  $\phi$  is the bistatic angle at
% range R [ $\phi = 2 * \text{atan}(11.25\text{cm}/R)$ ] and  $\theta_{3dB}$  is the 3dB angle of the
% lens ( $\theta_{3dB} = 5.7$  degrees).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear

imax = 512;
fs = 50000; %sampling frequency
f = fs*(0:(imax-1))/imax;
freqoffset = 2390; %Frequency corresponding to
% offset due to cable between
% radar and horns (Hz). From
% calibration measurements.%
c = 3e10; %Speed of light in air, in cm/s%

timeoffset = (2*freqoffset)/(15.4*c); %Time offset%

Res = 0.1476; %free-space resolution of radar in meters%
numpoints = 220; %Number of data points%
off = freqoffset/(fs/imax);
offsetvv = floor(off); %Offsets determined from calibration target%
offsethh = floor(off);
offsetvh = floor(off);
offsethv = offsetvh;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Rcal = 3.35; %From range calibrations (Luneberg lens)%
Prcalvv = 25.39;
```

```

Prcalhh = 25.17;
phi = (180/pi)*2*atan(.1125/Rcal);
lenscorrec = 10*log10(exp(-0.693*(phi/5.7)^2));
%%%%%%%%%%
Prcalvh = mean([Prcalvv Prcalhh]);
Prcalhv = Prcalvh;

stmo = input('Enter month data were collected: ','s');
stdy = input('Enter day data were collected: ','s');
dt = [stmo,'-',stdy,'-93'];

%%-----"Prcaldl(h,v)" are determined from calibration -----
%%-----measurements on 6/23/93 and 7/2/93. For 7/9/93 -----
%%-----only the VV value is needed; it was 20.5 (the radar ----
%%-----was set at higher gain, and so the Prcalvv, Prcalhh-----
%%-----should be adjusted accordingly as well).-----
if stdy == '09'
    Prcaldlv = 20.5
elseif stdy == '9'
    Prcaldlv = 20.5
else
    Prcaldlh = 12.1;
    Prcaldlv = 15.5;
end
sigmacal = 12;                %Cross-section of Luneberg lens at 13.5 GHz
(dB)%

angle = input('Enter incidence angle: ');
Rinch = input('Enter range at zero degrees, in inches: ');
R0 = Rinch*0.0254;

lett = input('Enter the first letter of the data file (c or x): ','s');
suffix = input('Enter the three-digit suffix: ','s');
N = input('Enter the number of data files: ');

if lett == 'c',
    if suffix(1) == '1',
        offset = offsethh;
        PrcaldB = Prcalhh;
        Prcaldl = Prcaldlh;
        polz = 'polztn.: HH';
    elseif suffix(1) == '2',
        offset = offsetvh;
        PrcaldB = Prcalvh;
        Prcaldl = Prcaldlh;
        polz = 'polztn.: HV';
    end
elseif lett == 'x',
    if suffix(1) == '1',
        offset = offsethv;
        PrcaldB = Prcalhv;

```

```

    Prcaldl = Prcaldlv;
    polz = 'polztn.: VH';
elseif suffix(1) == '2',
    offset = offsetvv;
    PrcaldB = Prcalvv;
    Prcaldl = Prcaldlv;
    polz = 'polztn.: VV';
end
end

%%%%%%%% Load in the data %%%%%%%%%%
ang = num2str(angle);
if N == 4,
    eval(['load ',lett,'hha',ang,'.',suffix]);
    eval(['load ',lett,'hva',ang,'.',suffix]);
    eval(['load ',lett,'vva',ang,'.',suffix]);
    eval(['load ',lett,'vha',ang,'.',suffix]);
    file1 = eval([lett,'hha',ang]);
    file2 = eval([lett,'hva',ang]);
    file3 = eval([lett,'vva',ang]);
    file4 = eval([lett,'vha',ang]);
    tstr = ['date: ',dt,'; suffix: ',suffix,'; angle: ',ang,'; ',polz];
    data(:,1) = file1(1:numpoints,1);
    data(:,2) = file2(1:numpoints,1);
    data(:,3) = file3(1:numpoints,1);
    data(:,4) = file4(1:numpoints,1);

elseif N == 3,
    stri1 = input('Enter polarization label of first file: ','s');
    stri2 = input('Enter polarization label of second file: ','s');
    stri3 = input('Enter polarization label of third file: ','s');
    eval(['load ',lett,stri1,'a',ang,'.',suffix]);
    eval(['load ',lett,stri2,'a',ang,'.',suffix]);
    eval(['load ',lett,stri3,'a',ang,'.',suffix]);
    file1 = eval([lett,stri1,'a',ang]);
    file2 = eval([lett,stri2,'a',ang]);
    file3 = eval([lett,stri3,'a',ang]);
    tstr = ['date: ',dt,'; suffix: ',suffix,'; angle: ',ang,'; ',polz];
    data(:,1) = file1(1:numpoints,1);
    data(:,2) = file2(1:numpoints,1);
    data(:,3) = file3(1:numpoints,1);

elseif N == 2,
    stri1 = input('Enter polarization label of first file: ','s');
    stri2 = input('Enter polarization label of second file: ','s');
    eval(['load ',lett,stri1,'a',ang,'.',suffix]);
    eval(['load ',lett,stri2,'a',ang,'.',suffix]);
    file1 = eval([lett,stri1,'a',ang]);
    file2 = eval([lett,stri2,'a',ang]);
    tstr = ['date: ',dt,'; suffix: ',suffix,'; angle: ',ang,'; ',polz];
    data(:,1) = file1(1:numpoints,1);
    data(:,2) = file2(1:numpoints,1);

```

```

elseif N == 1,
    stril = input('Enter polarization label of first file: ','s');
    eval(['load ',lett,stril,'a',ang,'.',suffix]);
    file1 = eval([lett,stril,'a',ang]);
    tstr = ['date: ',dt,'; suffix: ',suffix,'; angle: ',ang,'; ',polz];
    data(:,1) = file1(1:numpoints,1);
end

[rw cl] = size(data);

%%% Calculate the power spectrum %%%%%%%%%%%
w = hanning(numpoints);
for ii = 1:cl
    z(:,ii) = data(:,ii).*w;
end
ft = fft(z,imax);
mag = 20*log10(abs(ft));
powlin = (abs(fft(z,512))).^2;
time = 2*f/(15.4*c) - timeoffset;

%%%%%% Compute internal calibration (from feedthrough)
% For 512 pt. Fourier transform
feedpos = 8;
% For 220 pt. Fourier transform
%feedpos = 4;
newpwrcl(1:cl) = mag(feedpos,1:cl);

%%%%%% Calculate ranges, based on an average snow dielectric constant %%%
R0 = R0/cos(angle*pi/180);
epsilonavg = 1.75; %Average dielectric constant in snow%
Ressnow=Res/sqrt(epsilonavg); % Resolution in snow%
fftfrac = imax/numpoints;
m = round(fftfrac*R0/Res)+offset;
for iii = 1:m
    Range(iii) = Res*(iii-offset)/fftfrac;
end
for jj = m+1:imax
    Range(jj) = Res*(m-offset)/fftfrac + Ressnow*(jj-m)/fftfrac;
end

%%%%%% Calculate the illuminated volume at each range (call "volillg.m") %%%
V = volillg(angle,Range,Ressnow/fftfrac);

%%%%%% Adjust quantities to take into account the offset between the
%%%%%% oscillator and the horns.

Range=Range(offset+1:imax);
mag=mag(offset+1:imax,:);
V=V(offset+1:imax);

%%%%%% Convert everything that is not already in dB to dB for the sigma0

```

```
%%%%%%%%calculation.
```

```
RcaldB = 10*log10(Rcal);  
RangedB = 10*log10(Range);  
VdB = 10*log10(V);  
sigmacaldB = sigmacal;
```

```
%%%%%%%% calculate volume backscattering coefficient, "sigmaV".
```

```
scalarterm = Prcaldl - PrcaldB - 4*RcaldB + sigmacaldB + lenscorrec;  
for jjj = 1:cl  
    sigma(:,jjj) = mag(:,jjj) - newpwrdl(jjj) + 4*RangedB' +scalarterm;  
    sigmaV(:,jjj) = mag(:,jjj) - newpwrdl(jjj) + 4*RangedB' - VdB'  
+scalarterm;  
end  
sigmaVlin=(10).^(sigmaV/10);  
Rangelin=(10).^(RangedB'/10);  
Vlin=(10).^(VdB'/10);
```

```
%%%%%%%% Plot individual sigmaV vs. range on top of each other; then plot  
%%%%%%%% the average of the N sigmaV.
```

```
subplot(311)  
    h = 0.5*imax;  
    xx = Rangelin(1:h);  
    yy = sigmaV;  
if N == 4,  
plot(xx,yy(1:h,1),'y',xx,yy(1:h,2),'m',xx,yy(1:h,3),'g',xx,yy(1:h,4),'c')  
elseif N == 3,  
    plot(xx,yy(1:h,1),'y',xx,yy(1:h,2),'m',xx,yy(1:h,3),'g')  
elseif N == 2,  
    plot(xx,yy(1:h,1),'y',xx,yy(1:h,2),'m')  
elseif N == 1,  
    plot(xx,yy(1:h,1),'y')  
end  
xlabel('range from radar (m)')  
ylabel('sigmaV (dB)')  
title(tstr)  
set(gca,'xlim',[0 9])  
grid
```

```
subplot(312)  
temp = 10.^(sigmaV'/10);  
if N == 1  
    av = temp;  
else  
    av = (sum(temp))/N;  
end  
avgsigma = 10*log10(av');  
rrr = Rangelin(1:0.5*imax);  
avsg = avgsigma(1:0.5*imax);
```



```

plot(Rangelin(1:0.5*imax),avgsigma(1:0.5*imax))
xlabel('range from radar (m)')
ylabel('sigmaV (dB)')
title('average sigmaV')
set(gca,'xlim',[0 9])
grid
ang = num2str(angle);
eval(['save ',lett,ang,'_',suffix, ' rrr avsg '])

```

```

%%%%%%%%%% Plot linear power vs. travel time.

```

```

subplot(313)
tm = time(1:h);
pl = powlin;
if N == 4,
plot(tm,pl(1:h,1),'y',tm,pl(1:h,2),'m',tm,pl(1:h,3),'g',tm,pl(1:h,4),'c')
elseif N == 3,
    plot(tm,pl(1:h,1),'y',tm,pl(1:h,2),'m',tm,pl(1:h,3),'g')
elseif N == 2,
    plot(tm,pl(1:h,1),'y',tm,pl(1:h,2),'m')
elseif N == 1,
    plot(tm,pl(1:h,1),'y')
end
xlabel('time (s)')
ylabel('linear power')
title('travel times')
set(gca,'xlim',[0 6e-8])
grid

```

```

% Compute illuminated area
% bw is beamwidth of antenna
bw=10*pi/180;
rad=angle*pi/180;
hh=R0*cos(rad);
a=R0*tan(bw/2);
X2=hh*tan(rad+bw/2);
X1=hh*tan(rad-bw/2);
b=0.5*(X2-X1);
Aill=pi*a*b;

```

```

% Compute sigma nought. %-----

```

```

count = floor(R0/(Res/fftfrac))-7;

```

```

pow = 10.^(mag(1:180,:)/10);
precratio = pow(count:180,1)./(10^(PrcaIdB/10));
pww = sum(precratio);

```

```
ptrnratio = (10^(Prcaldl/10))/(10^(newpwrdl(1)/10));
rangeratio = (R0/Rcal)^4;
sigmaCalLin = 10^(sigmacaldB/10);

sig0 = 10*log10(pww*ptrnratio*rangeratio*sigmaCalLin/Aill) + lenscorrec

%avlin = 10.^(avgsigma/10);
%newsigno = 10*log10((Ressnow/fftfrac)*sum(avlin(count:180)))
```

%% TEST.M written by C. M. Gero, 1994

%%%%%%%% input %%%%%%%%%

```
%angle = input('enter incidence angle: ');  
lett = input('enter the first letter of the data file (c or x): ','s');  
suffix = input('enter file suffix: ');  
stsuffix = num2str(suffix);
```

```
stmo = input('enter month data were collected: ','s');  
stdy = input('enter day data were collected: ','s');  
dt = [stmo,'-',stdy,'-93'];  
ht = input('enter range to snow surface(inches): ');  
dp = input('enter depth (below surf.) of layer of interest(inches): ');
```

%%%%%%%% polarization %%%%%%%%%

```
if lett == 'c',  
    if stsuffix(1) == '1';  
        polz = 'polztn.: HH';  
    else stsuffix(1) == '2';  
        polz = 'polztn.: HV';  
    end  
elseif lett == 'x';  
    if stsuffix(1) == '1';  
        polz = 'polztn.: VH';  
    else stsuffix(1) == '2';  
        polz = 'polztn.: VV';  
    end  
end
```

%%%%%%%% load & organize data into matrix %%%%%%%%%

```
i = 1;  
for angle = 0:5:20  
    e = 0;  
    count = 0;  
    while e <= 1 & count <= 4;  
        ang = num2str(angle);  
        suff = num2str(suffix);  
        filename = [lett,ang,'_',suff,'.mat'];  
        e = exist(filename);  
        if e <= 1;  
            suffix = suffix + 1;  
            count = count + 1;  
        elseif e >= 2;  
            eval(['load ' filename]);  
            Mrange(1:256,i) = rrr;  
            Msigma(1:256,i) = avsg;  
            a(i) = angle;  
            suffix = suffix + 1;  
        end  
    end  
end
```

```

        i = i + 1;
        count = 0;
    end
end
if count >= 1 & e == 0;
suffix = suffix - count;
end
end

for angle = 25
    for suffix = suffix:suffix + 2
        ang = num2str(angle);
        suff = num2str(suffix);
        filename = [lett,ang,'_',suff,'.mat'];
        e = exist(filename);
        if e <= 1;
        elseif e >= 2;
            eval(['load ' filename]);
            Mrange(1:256,i) = rrr;
            Msigma(1:256,i) = avsg;
            a(i) = angle;
            i = i + 1;
            suffix = suffix + 1;
            break
        end
    end
end

for angle = 30:10:40
    for suffix = suffix - 3:suffix + 2
        ang = num2str(angle);
        suff = num2str(suffix);
        filename = [lett,ang,'_',suff,'.mat'];
        e = exist(filename);
        if e <= 1;
        elseif e >= 2;
            eval(['load ' filename]);
            Mrange(1:256,i) = rrr;
            Msigma(1:256,i) = avsg;
            a(i) = angle;
            i = i + 1;
            endsuff = suff;
            suffix = suffix + 1;
            break
        end
    end
end

for angle = 45
    for suffix = suffix:suffix + 2
        ang = num2str(angle);

```

```

    suff = num2str(suffix);
    filename = [lett,ang,'_',suff,'.mat'];
    e = exist(filename);
    if e <= 1;
    elseif e >= 2;
        eval(['load ' filename]);
        Mrange(1:256,i) = rrr;
        Msigma(1:256,i) = avsg;
        a(i) = angle;
        i = i+1;
        suffix = suffix + 1;
        break
    end
end
end
end

for angle = 50
    for suffix = suffix - 3:suffix + 2
        ang = num2str(angle);
        suff = num2str(suffix);
        filename = [lett,ang,'_',suff,'.mat'];
        e = exist(filename);
        if e <= 1;
        elseif e >= 2;
            eval(['load ' filename]);
            Mrange(1:256,i) = rrr;
            Msigma(1:256,i) = avsg;
            a(i) = angle;
            i = i + 1;
            suffix = suffix + 1;
            break
        end
    end
end
end

angle = 55;           %%% check for angles 55 through 60
e = 0;
count = 0;
while e <= 1 & count <= 8;
    for suffix = suffix - 5:suffix + 1;
        ang = num2str(angle);
        suff = num2str(suffix);
        filename = [lett,ang,'_',suff,'.mat'];
        e = exist(filename);
        if e <= 1;
            count = count + 1;
        elseif e >= 2;
            eval(['load ' filename]);
            Mrange(1:256,i) = rrr;
            Msigma(1:256,i) = avsg;
            a(i) = angle;

```

```

        endsuff = suff;
        break
    end
end
if angle >= 60;
    break, end
if e >= 2;
    break, end
angle = angle + 1;
count = 0;
end

%%%%%% calculate actual dist. to snow surf. along beam path %%%%%%%%%%

[horiz vert] = size(Mrange);

for i = 1:vert
    height = ht in meters
    height = 0.0254*ht;
    newheight = height-.25;
    adjustment(.25m)
    theta(i) = a(i)*pi/180;
    dist(i) = newheight/cos(theta(i));
end
    %%% theta = angle of incidence in radians
    %%% newheight = calc. curve
    %%% height = vertical distance from
    %%% radar to snow surface
    %%% dist = distance along beam path
    %%% from radar to snow surface (in air)

%%%%%% Snell's law correction (refraction at snow surf.) %%%%%%%%%%

depth = 0.0254*dp;
dpth = num2str(depth);
for i = 1:vert
    ns = 1.32;
    na = 1.00;
    angrefr(i) = asin(na*sin(theta(i))/ns);
    distrefr(i) = depth/cos(angrefr(i));
    traveldist(i) = dist(i) + distrefr(i);
end
    %%% depth (from surf.) to layer in meters
    %%% index of refraction for snow
    %%% index of refraction for air
    %%% => angle of refraction
    %%% => distance along refracted
    %%% => total (1 way) distance of
    %%% beam
    %%% travel from radar to layer at depth

%%%%%% designate stpt-endpt interval for each column(i) in Msigma matrix
%%%%%%

for i = 1:vert
    interval(i) = .50/cos(angrefr(i));
    vertical
    start(i) = traveldist(i) - interval(i);
    depth
    finish(i) = traveldist(i) + interval(i);
    for horiz = 1:256;
        if start(i) < rrr(horiz);
            stpt(i) = horiz;
            break,end
        end
    end
end

```

```

    end
end

for i = 1:vert
    start(i) = traveldist(i) - interval(i);
    finish(i) = traveldist(i) + interval(i);
    for horiz = 1:256;
        if finish(i) < rrr(horiz);
            endpt(i) = horiz;
            break,end
        end
    end
end

startpoint = rrr(stpt);
endpoint = rrr(endpt);

[row column] = size(Msigma);
stend = endpt - stpt;

%%% for each column (i): calculate linpwr from (logarithmic) Msigma, find
%%% peaks in linpwr columns (=linpeak), find highest peak in linpeak
column %%
%%% (=maxlinpk), and create range vector (1 by i) corresponding to
maxlinpk %

for i = 1:column
    numpeak = 0;
    linpwr = 10.^(Msigma(stpt(i):endpt(i),i)/10);
    linrange = Mrange(stpt(i):endpt(i),i);
    for k = 2:stend(i);
        if linpwr(k) > linpwr(k-1);
            if linpwr(k) > linpwr(k+1);
                linpwr(k);
                linpwr(k-1);
                linpwr(k+1);
                numpeak = numpeak + 1;
                linpeak(numpeak,i) = linpwr(k);
                rangepeak(numpeak,i) = linrange(k);
            end
        end
    end
    for k = 1:numpeak;
        if linpeak(k,i) ~= 0 & k == 1;
            maxlinpk(i) = linpeak(k,i);
            rmax(i) = rangepeak(k,i);
        elseif linpeak(k,i) ~= 0 & k >= 2;
            if linpeak(k-1,i) ~= 0 & linpeak(k,i) > linpeak(k-1,i) |
linpeak (k-1,i) == 0
                maxlinpk(i) = linpeak(k,i);
                rmax(i) = rangepeak(k,i);
            end
        end
    end
end

```

```

        else linpeak(k,i) == 0
        end
    end
    maxsigpk(i) = 10*log10(maxlinpk(i));
end

%%% linpeak = linpwr values for all peaks within the stpt-endpt interval.
%%% maxlinpk = the highest linpeak in a set of linpeak values (i.e. for
one %%% angle)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot the data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

subplot(3,1,1)
plot(a,maxsigpk,'r*',a,maxsigpk,'r:')
    xlabel('angle of incidence')
    ylabel('sigma V (dB)')
    grid on
    title([dt,'      suffix=',stssuff,'-',endsuff,'      ',polz,'
depth=',dpth,'m'])

subplot(3,1,2)
plot(a,rmax,'r*',a,rmax,'r:')
    xlabel('angle of incidence')
    ylabel('range (m)')
hold on
plot(a,dist,'co',a,dist,'c:')
    %text(2,2.7,' ...*... dist. to high peak in range interval')
    %text(2,2.5,' ...o... actual dist. along beam path from radar to
surf.')
```

```

    grid on

subplot(3,1,3)
plot(2*distrefr,maxsigpk,'r*',2*distrefr,maxsigpk,'r:')
    xlabel('2-way dist. traveled in snow (m)')
    ylabel('sigmaV (dB)')
    grid on

%subplot(4,1,4)
%plot(rmax,maxlinpk,'r*',rmax,maxlinpk,'r:')
%    xlabel('range (m)')
%    ylabel('linear power')
%    grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% save the data
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

date = [stmo,stdy,'93'];

eval(['save ',lett,date,stssuff,' a maxsigpk rmax maxlinpk distrefr dist
depth '])

```



```
%%%                AVPLOT.M                written by C. M. Gero, 1995
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% input %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
stmo = input('enter month data were collected: ','s');  
stdy = input('enter day data were collected: ','s');  
dt = [stmo,stdy,'93'];  
lett = input('enter the first letter of the data file (c or x): ','s');  
spot = input('enter number of spots to be averaged: ');  
polz = input('enter polarization: ','s');
```

```
for i = 1:spot  
    stsuff(i) = input('enter file suffix: ');  
    suff(i,1:3) = num2str(stsuff(i));  
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% load the data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
for i = 1:spot  
    file = [lett,dt,suff(i,1:3)];  
    eval(['load ',file,'.dat']);  
    data = eval(file);  
    a = data(:,1);  
    rmax = data(:,2);  
    maxsigpk = data(:,3);  
    maxlinpk = data(:,4);  
    temp = data(:,5);  
  
    l = length(maxsigpk);  
    sig(i,1:l) = maxsigpk';  
    ang(i,1:l) = a';  
    rangemax(i,1:l) = rmax';  
    lin(i,1:l) = maxlinpk';  
    tmp(i,1:l) = temp';  
    %dist(i,1:l) = dist';  
    %distrefr(i,1:l) = distrefr';  
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% average data from diff. spots %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
avlin = sum(lin)/spot;  
avsig = 10*log10(avlin);  
avang = sum(ang)/spot;  
avrmax = sum(rangemax)/spot;  
avtmp = sum(tmp)/spot;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% plot the data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
date = [stmo,'-',stdy,'-93'];  
avtmp = avtmp(1);  
ttltmp = num2str(avtmp);
```

```

subplot(3,1,1)
plot(avang,avsig,'r*',avang,avsig,'r:')
    xlabel('angle of incidence')
    ylabel('average sigma V (dB)')
    grid on
    title([date,' polz.=' ,polz,' avg.temp=' ,ttltmp,'C' ])

%subplot(3,1,2)
%plot(avang,avrmax,'r*',avang,avrmax,'r:')
%    xlabel('angle of incidence')
%    ylabel('range (m)')
%hold on
%plot(avang,dist,'co',avang,dist,'c:')
%    %text(2,2.7,' ...*... dist. to high peak in range interval')
%    %text(2,2.5,' ...o... calc. dist. along beam path from radar to
surf.')
%    grid on

%subplot(3,1,3)
%plot(2*distrefr,avsig,'r*',2*distrefr,avsig,'r:')
%    xlabel('2-way dist. traveled in snow (m)')
%    ylabel('sigmaV (dB)')
%    grid on

subplot(3,1,3)
plot(ang(1,:),sig(1:),'r:',ang(1,:),sig(1:),'ro',ang(2,:),sig(2:),'c:',a
ng(2,:),sig(2:),'c+',ang(3,:),sig(3:),'y:',ang(3,:),sig(3:),'y*')
    xlabel('angle of incidence')
    ylabel('sigma V (dB)')
    grid on

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% save data %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

eval(['save ',polz,dt ' avang avsig avrmax suff avtemp'])

```