BASIC MAGNETIC PROCESSING AND DISPLAY IN MATLAB

Charles T. Young, Department of Geological and Mining Engineering and Sciences, Michigan Technological University, Houghton, MI 49931, telephone 906 487-2072 fax 906 487-3371
ctyoung@mtu.edu

ABSTRACT

A short Matlab program is used to create colorized and contoured maps of data from XYZ files (data with random locations) and do other processing. The programs control gridding and smoothing the data and allow the user to set contour values and utilize vivid color scales. Magnetic data usually display plus-minus anomalies which are due to the dipolar nature of the magnetic sources and the interaction with the Earth's field. These anomalies may be converted into positive values as if the observations were made at the Earth's magnetic pole and if the magnetization of the buried body was purely induced. The data are converted to the magnitude of the analytic signal. The imaginary part of the analytic signal is computed with the Matlab Hilbert transform which converts sine components into cosine components and vice versa. If magnetic anomalies from an infinitesimal dipole are converted to positive anomalies by this method, then the depth to the dipole source may be found directly by inspection using half-width rules. This process of finding the depth is further simplified by displaying the contours as as 10 times the logarithm to the base 10.

PRINCIPLES

Magnetic data usually display plus-minus anomalies which are due to the dipolar nature of the magnetic sources and the interaction with the Earth's field. These anomalies may be converted into positive anomalies by using the Hilbert transform to create the "analytic signal". The result is that the profile or map is converted to the field that would be observed if the observations were made at the Earth's magnetic pole and if the magnetization of the buried body was purely induced. In magnetic processing literature the operation is known as "reduction to the pole". Tabbaugh, 1997, presents an excellent discussion of principles and examples of reduction to the pole to archaeological data. In principle, reduction to the pole requires a knowledge of the direction of the Earth's field and the direction of magnetization of the body causing the anomaly e.g. Blakely (1995). I argue that since the reduction to the pole is simply a matter of shifting phases of signals to force all the anomalies to be positive, that all processes that accomplishes this are equivalent. Readers who disagree may wish to call my processing "pseudo reduction to the pole".

The quantity that is desired from the magnetic data is also known as the "analytic signal", which is the square root of the sum of the squares of the original signal and its Hilbert transform. The operation is identical to finding the amplitude of the (complex) seismic trace, Taner et al (1979). The Hilbert transform is a 90 degree phase shift operator; it does not change the amplitude of the original signal. Its operation can be described in either as a convolution in the space (time) domain or as complex multiplication in the frequency domain, as described in Claerbout (1985). An example of the HT applied to a elemental tapered sine-like anomaly or wavelet is shown in Figure 1, top panel. The cosine-like Hilbert transformed anomaly is shown in the middle panel and the square root of the sum of the squares of the original signal and the Hilbert transform is shown in the bottom panel and appears as a positive Gaussian-like bump.

The Matlab Hilbert transform operates on one-dimensional data. For the work described here, it is necessary to adapt the HT to two-dimensional data. I do this by analogy to the two-dimensional Fourier Transform (strictly, the "discrete Fourier Transform").

Basic equations (e.g. Lim, 1990) show that the two-dimensional discrete Fourier Transformation consists of computing the one-dimensional discrete Fourier transformation of each row of data to form another matrix, then computing the one dimensional Fourier transformation on each column of that second matrix . Thus, the two dimensional HT data could be accomplished by computing the forward two dimensional Fourier Transform (2DFT) of the data, carrying out the phase shifting, then inverse 2DFTing back to the data domain. Fortunately, it is simpler to utilize the one-dimensional HT in an analogous manner to accomplish the two dimensional HT. This can be done with several steps of Matlab programming which are grouped into three lines.

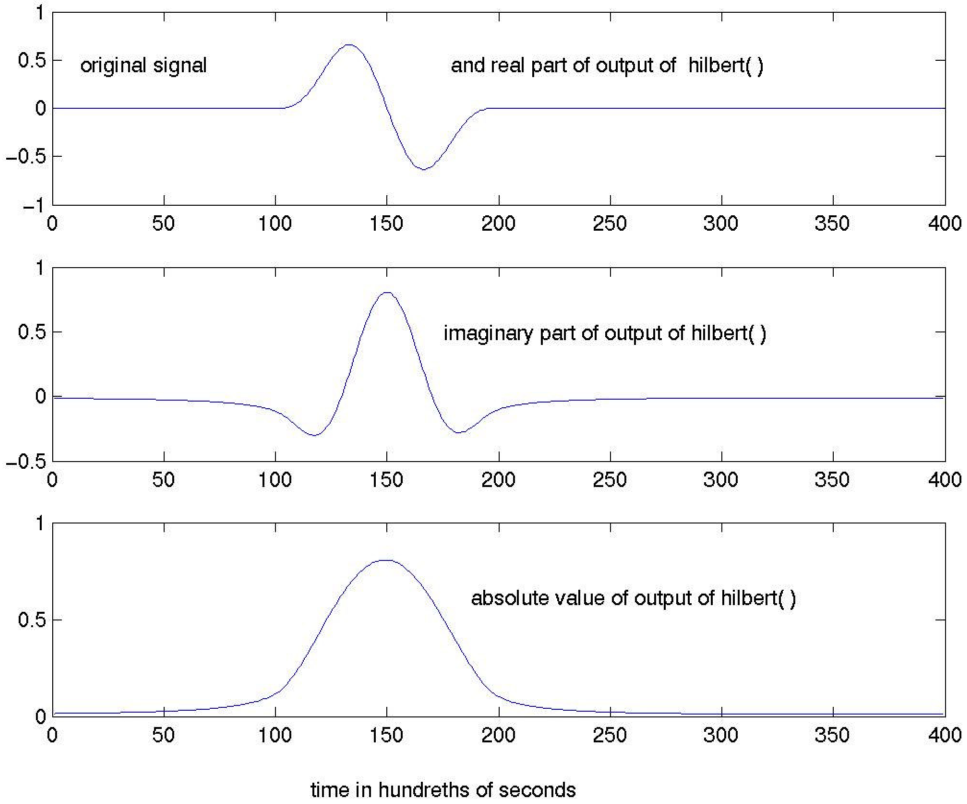

Figure 1. Computed example of Hilbert transform and magnitude of the analytic signal of a simple anomaly or wavelet. Time zero is at point 100 for these examples.

MATLAB IMPLEMENTATION

The data are from a Geometrics 858 gradiometer, and are created with the Geometrics MagMapper program in XYZ format. I have manually removed a header line consisting of column labels and columns of the upper and lower magnetometer readings, leaving x location, y location and magnetic gradient.

The data consist of columns:

Column 1	Column 2	Column 3
x location	y location	magnetic gradient

The survey dimensions were 15 meters in the x direction and 20 meters in the y direction.

The complete Matlab program is provided in Appendix I. Important steps are discussed here.

The data are interpolated onto regular grid. The x and y values at 0.2 meter spacing (matrices XI and YI) are defining the grid are created by

```
XI=0.4:0.2:14.6;                    % create new x values for gridding data
YI=0.4:0.2:19.6;                    % create new y values for gridding data
```

I clipped 0.4 m off the start and end of these matrices, because the interpolation routine in Matlab will not interpolate to the edge of the grid.

The regridding is accomplished by

```
ZI=griddata(x,y,z,XI,YI');
```

The ' mark in YI is the matrix transpose operator. In this case, it transforms the row matrix YI into a column matrix.

A single call of the one-dimensional Matlab Hilbert transform function will cause it to operate on all the columns of the two dimensional matrix. We are interested in the magnitude of the original data and its hilbert transform treated as complex pairs, and thus use the Matlab abs (absolute value) function.

```
h1=abs(hilbert(ZI));
```

The output matrix obtained above needs to be transposed because the Hilbert transform function operates on columns.

```
h2=abs(hilbert(h1'));
```

Finally, the output matrix needs to be transposed back to its original orientation.

```
ZI=h2';
```

Actually, the operations on these three lines could be accomplished in one line, which would save memory by eliminating matrices h1 and h2, but they are kept separate here for clarity.

For display, I often scale the data by finding 10 times the logarithm to the base 10 of data. This is the "db" (decibel) scale for power which is well known in electrical engineering. I display the data as a colorized map overlaid with contours. The scaling and contouring provides two advantages: the visibility of weak anomalies is improved, and it is simple to identify depth by the half-width method (e.g. Bevan, 1998); the half width of a peak is found simply by counting down six contour lines from the peak.

APPLICATION TO FIELD DATA

Magnetic gradiometer data used here were obtained within 100 meters of an abandoned lighthouse Lac La Belle, in the Keweenaw Peninsula portion of the Upper Peninsula (Northern Michigan). The lighthouse was operated during the first half of the 20th century. Written records of buildings and wharfs is incomplete. The soil is loose sand of unknown depth, thus it could have been redistributed by severe storms to cover former buildings or foundations. The survey grid was located by scanning the lighthouse grounds with the gradiometer without recording data. The data grid enclosed several anomalies found by the scan. The dimensions of the grid are 15 meters long by 20 meters wide. Data were obtained along lines spaced 0.5 meters apart.

The resulting data are shown before and after Hilbert transformation. The positive-negative anomaly pairs have clearly been converted into positive-only anomalies. Some anomalies are approximately equally spaced along orthogonal lines. Thus the spatial distribution and the fact that the survey is near the lighthouse supports the argument that they are manmade structures, perhaps some part of a building or wharf.

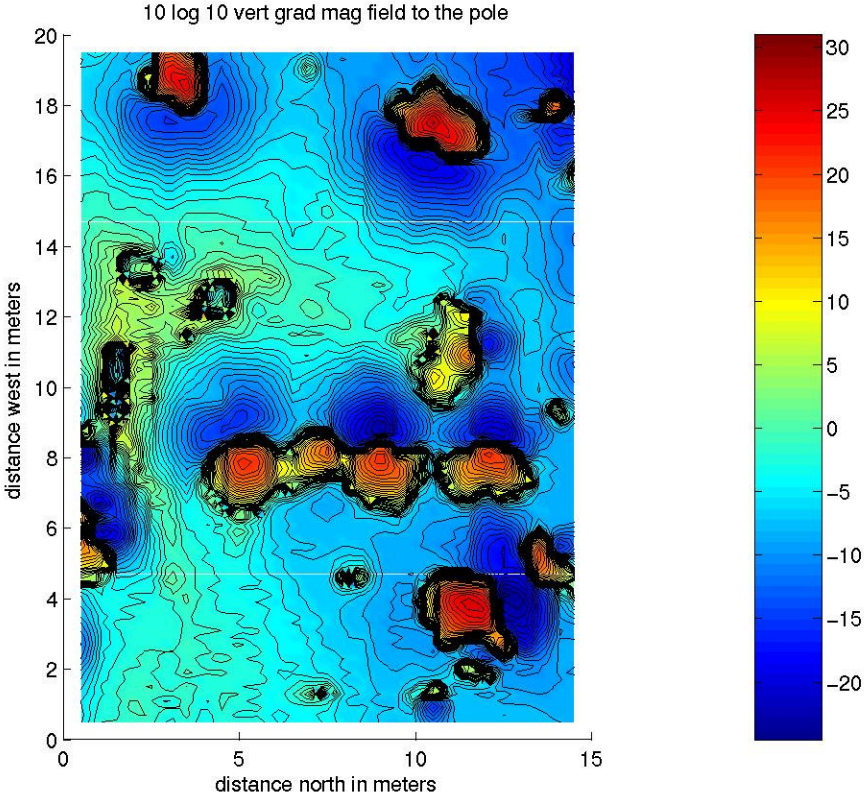

Figure 2. Original magnetic gradiometer data, scaled by sign(data)*10* logarithm to the base 10 (absolute value (data)). The contour interval is unity.

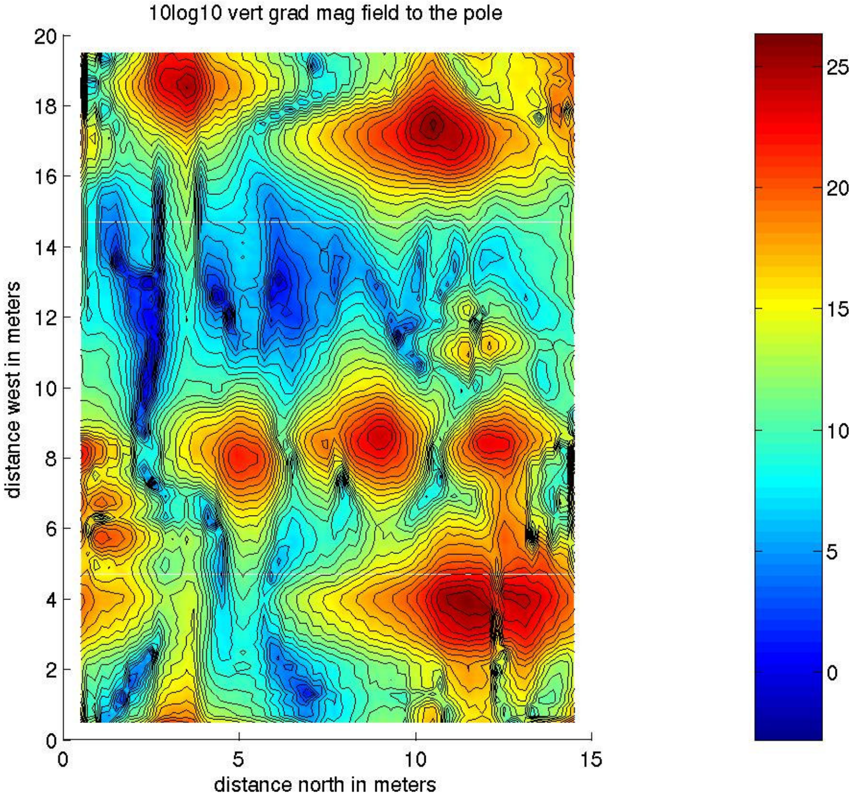

Figure 3 . Magnetic gradiometer data after conversion to analytic signal, scaled by 10*logarithm to the base 10 (data). The contour interval is unity.

CONCLUSION

This paper has described the use of a short Matlab program for interpretation of magnetic field data. The main steps in the program are filtering by Hilbert transform to convert positive-negative anomalies to positive only, displaying 10log10 of the data, colorization of the data and overlaying with line contours.

The author finds the default rainbow spectrum colorization provided by Matlab's pcolor command to be more eye-catching than that of a competing contouring program, and once the basic color presentation commands are learned they are easy and quick to use. The computation presented here can be carried out on both the student and professional versions of Matlab 6.5.

REFERENCES

Bevan, B. W., 1998, Geophysical Exploration for Archaeology; An Introduction to Geophysical Exploration, Midwest Archaeological Center, Special Report No. 1., United States Department of the Interior, Midwest Archaeological Center, Lincoln, Nebraska. ca. 300 pp.

Blakely, Richard, 1995, Potential Theory in Gravity and Magnetic Applications, Cambridge University

Press, New York , 441 pp.

Claerbout, Jon, F., 1985, Fundamentals of Geophysical Data Processing With Applications to Petroleum Prospecting, Blackwell Scientific Publications, Boston.

Lim, J. S., 1990 , Two-dimensional Signal and Image Processing, Prentice Hall, Englewood Cliffs, New Jersey, 694 pp.

MathWorks, 2003, http://www.mathworks.com/

Tabbaugh, A., Desvignes, G., Dabas, Michel, 1997, Processing of z gradiometer magnetic data using linear transformations and analytical signal, Archaeological Prospection, v. 4, pp1-13.

Taner, M. T., Koehler, F., and Sherrif, R. E., 1979, Complex seismic trace analysis, Geophysics, v. 44, pp 1041-1063.

Appendix I Matlab program listing.

```
% This matlab program makes a colored map of magnetics data.
% and adds black line contours
% The data file must be named d.dat.
%
% Copyright Charles T. Young, Houghton, MI 49931  2003
% Copying permitted for instructional purposes.
% For commercial use please contact the author at ctyoung@mtu.edu.
%

load d.dat                         % load file .. numbers go into
                                   %       matrix d
x=d(:,1);                          % assign x to first column of d
y=d(:,2)/4;                        % assign y to second column of d
                                   % original y values were 4 times too large
z=d(:,3);                          % assign new z to fifth column of d
XI=0.5:0.2:14.5;                           % create new x values for gridding data
YI=0.5:0.2:19.5;                           % create new y values for gridding data
ZI=griddata(x,y,z,XI,YI');         % grid data
% hilbert transform operates on columns of argument
h1=abs(hilbert(ZI));
h2=abs(hilbert(h1'));
ZI=h2';
ZI=10*log10(ZI);                   % work in db
hold on                            % make multiple plots on same axis
pcolor(XI,YI,ZI)                   % display data as colored tiles
shading interp                     % smooth the tiles
axis equal                         % fix up the axes
xlabel('distance north in meters')
```

```
ylabel('distance west in meters')
title('10log10 vert grad mag field to the pole')
colorbar                              % the color scale on the rhs of the figure
conts=0:1:60;                         % creates contour values
contour(XI,YI,ZI,conts,'k')           % plot contours with black (k) lines
axis equal
axis([0 15  0 20])
hold off
```

APPENDIX II: Matlab help file for function Hilbert

```
HILBERT  Discrete-time analytic signal via Hilbert transform.
   X = HILBERT(Xr) computes the so-called discrete-time analytic signal
   X = Xr + i*Xi such that Xi is the Hilbert transform of real vector Xr.
   If the input Xr is complex, then only the real part is used: Xr=real(Xr).
   If Xr is a matrix, then HILBERT operates along the columns of Xr.

   HILBERT(Xr,N) computes the N-point Hilbert transform.  Xr is padded with
   zeros if it has less than N points, and truncated if it has more.

   For a discrete-time analytic signal X, the last half of fft(X) is zero,
   and the first (DC) and center (Nyquist) elements of fft(X) are purely real.

   Example:
     Xr = [1 2 3 4];
     X = hilbert(Xr)
   produces X=[1+1i 2-1i 3-1i 4+1i] such that Xi=imag(X)=[1 -1 -1 1] is the
   Hilbert transform of Xr, and Xr=real(X)=[1 2 3 4].  Note that the last half
   of fft(X)=[10 -4+4i -2 0] is zero (in this example, the last half is just
   the last element).  Also note that the DC and Nyquist elements of fft(X)
   (10 and -2) are purely real.

   See also FFT, IFFT.
```