gMagy

GeoMLib (β)

Marine engineering geophysical data processing toolbox

In progress

Ivan V. Dmitriev
04.02.2020

Contents

Figures list

Tables list

1 gMagy functions set description

MatLab functions set for reading G882 and SeaSpy marine magnetometer/gradiometer files, simple modeling and charting. The functions are shown in ***Table 1***.

Table 1 gMagy and the same functions

Function name	Function description
Read MagLog files (Geometrics software; G882 cesium magnetometer)	
gMagyReadG88Int	Read MagLog's *.int file for single magnetometer
gMagyReadG88Mag	Read MagLog's *.mag file for single magnetometer
gMagyReadG88tvgMag	Read MagLog's *.mag file for TVG-frame
gMagyG88GpsRead	Read MagLog's *.gps file ($GPGGA message)
Read SeaSpy/SeaSpy2 logged data (Marine Magnetics; logged with gLog)	
gLogMagyReadSeaSpy	Read SeaSpy Magnetometer data from gLog (see gLog manual)
gLogMagyReadSeaSpyGrad	Read SeaSpy Gradiometer data from gLog (see gLog manual)
Simple magnetic field modeling	
gMagyModNormal	Normal field's components calculation
gMagyModDipol	Dipole field modeling
gMagyModCylinder	Horizontal Cylinder field modeling
Graphics and pick	
gMagyDrawWiggle	Draw wiggle
gMagyDrawWiggleMask	Draw wiggle with holes
gAcadWiggle	Draw Wiggle in XY coordinates (see gAcad manual)
gAcadWiggleMask	Draw Wiggle in XY coordinates with Mask (see gAcad manual)
gMagyPickHandle	Targets pick

1.1 Read data

Today the gMagy set includes functions for read files from two type of equipment:

-- G882 cesium magnetometer and gradiometer data registered by MagLog (Geometrics software);

-- SeaSpy/SeaSpy2 proton magnetometer and gradiometer (Marine Magnetics) data registered by gLog software.

When gLog-software is used for data registration with SeaSpy/SeaSpy2, it is recommend to use follow configuration:

-- parallelize serial data flow from Magnetometer/Gradiometer Transceiver in two serial lines;

-- use BOB software for data visualization and commands to terminal window (first serial line);

-- not connect GPS to BOB software to except "time steps" when time synchronized;

-- use PC-time for Magnetometer/Gradiometer time set;

-- use gComLogOvOut logger for Magnetometer/Gradiometer data registration and data flow relay to navigation system (second serial line);

-- use gComLogOv for navigation messages registration (time, coordinates, etc).

When processing: 1) find time-shift between Computer Clock and Magnetometer/Gradiometer Clock (Magnetometer/Gradiometer data flow); 2) move Magnetometer/Gradiometer Clock time-stamps to

Computer Clock; 3) calculate UTC time-stamps, for Magnetometer/Gradiometer Clock (equal Computer Clock) using navigation messages data flow (must include time-messages).

1.2 Simple modeling

Calculate magnetic field from elementary objects. There are:

-- Dipole with parameters: Magnetic Moment (Module, Declination and Inclination), dipole's coordinate and field calculation points coordinate;

-- Horizontal Cylinder (line of dipoles) with parameters: Magnetic Moment (Module, Declination and Inclination), two cylinder's points coordinate and field calculation points coordinate.

The three coordinate systems are used for magnetic modeling. The space coordinate system: x (forward/north), y (right/east), z (up). The Earth magnetic field coordinate system: x (forward/north), y (right/east), z (down); Declination – rotation from x-to-y +; Inclination – rotation from xy-to-z +. Magnetization coordinate system: y (forward/north), x (right/east), z (down); Declination – rotation from x-to-y +; Inclination – rotation from xy-to-z + (*Figure 1*).

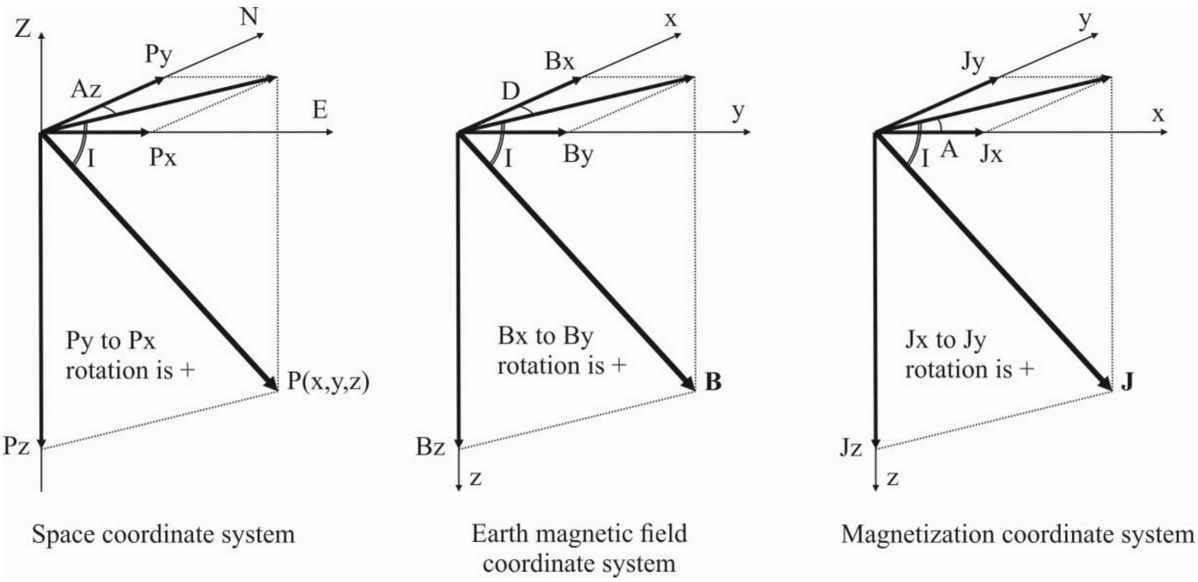

Figure 1 Magnetic parameters coordinate system

Dipole

Dipole's (sphere) magnetic field calculation (isotropic, no demagnetization adjustment, SI values; X, Y, Z, I, D are in magnetization coordinate system; x, y, z are in space coordinate system):

$$X = M\frac{(2x^2 - y^2 - z^2)\cdot\cos(I)\cdot\cos(D) + 3x\cdot(y\cdot\cos(I)\cdot\sin(D) - z\cdot\sin(I))}{(x^2 + y^2 + z^2)^{5/2}};$$

$$Y = M\frac{(2y^2 - x^2 - z^2)\cdot\cos(I)\cdot\sin(D) + 3y\cdot(x\cdot\cos(I)\cdot\cos(D) - z\cdot\sin(I))}{(x^2 + y^2 + z^2)^{5/2}};$$

$$Z = M\frac{(2z^2 - y^2 - x^2)\cdot\sin(I) - 3z\cdot\cos(I)\cdot(y\cdot\sin(D) + x\cdot\cos(D))}{(x^2 + y^2 + z^2)^{5/2}};$$

$$M_I = \frac{4\pi\chi B_{norm}\cdot r^3}{3\cdot 100};$$

where,

x,y,z, – points coordinate relative sphere's position (0,0,0);

X,Y,Z – magnetic induction field components;

M – sphere's magnetic moment intensity;

D – magnetic moment azimuth (declination);

I – magnetic moment inclination;

$$\vec{M} = \vec{M}_I + \vec{M}_R;$$

M_R – sphere's remnant magnetization;

M_I – sphere's induced magnetization;

χ – magnetic susceptibility;

B_{norm} – normal magnetic induction field intensity;

r – sphere's radius.

The dipole's (sphere) magnetic field model's parameters and modeling results are shown in ***Figure 2***; the Potent v4.12.04 software was used for modeling. The Potent v4.12.04 and gMagyModDipol function calculations results comparison is shown in ***Figure 3***.

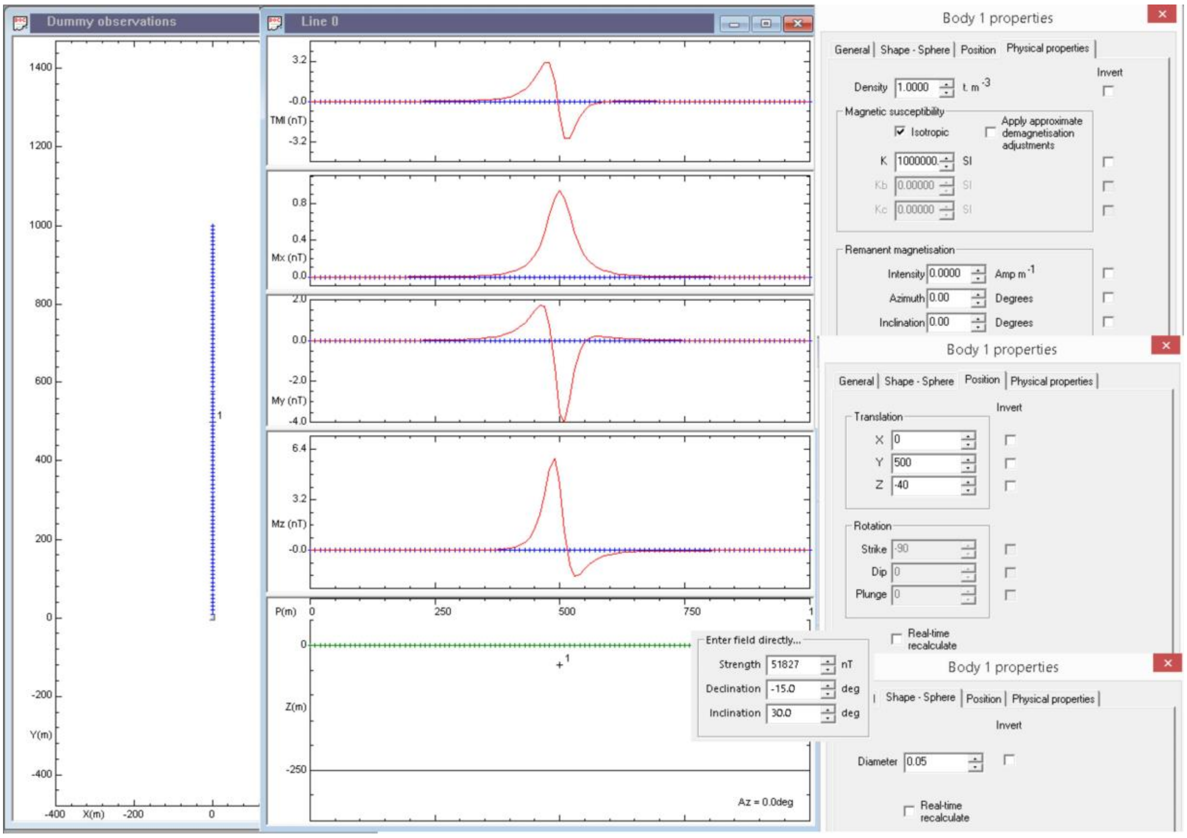

Figure 2 Potent v4.12.04 model's parameters and modeling results

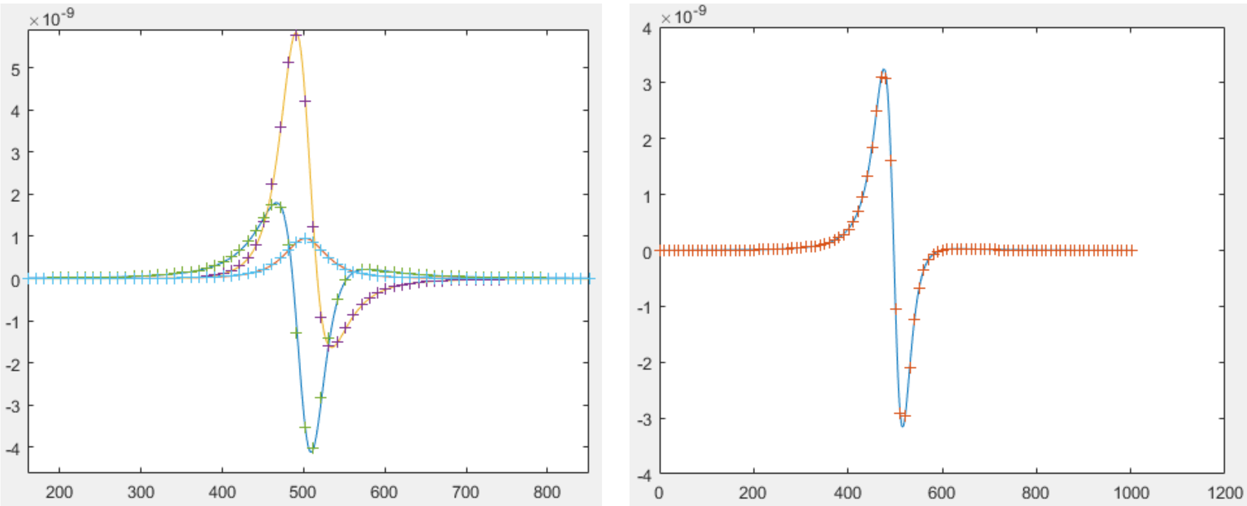

Figure 3 Potent v4.12.04 (cross-marks) and gMagyModDipol (lines) calculation comparison

Left side – X, Y, Z components; right side – Ta field. Horizontal axis – meters; vertical axis – Tesla.

Cylinder

Need to check. In progress.

2 Magnetometers files read/write

2.1 Read MagLog's *.int file for single magnetometer

function HInt=gMagyReadG88Int(fName,key)

Read data from text file with was created MagLog Geometrics program for G88x single magnetometer (*.INT)

Parameters:

fName – reading file name;

key – the key for Int-file format;

HInt – reading data structure with fields: CompDay,CompTime,GpsLon,GpsLat,GpsFixQuality, GpsHgtGeoid,'MagAbsT,MagPrecSignal,Depth,Altitude,INT_ShiftLon,INT_ShiftLat,INT_Atargets, INT_Nmags,INT_LonMag,INT_LatMag,INT_Line,INT_Layback

if key==1, then *.INT file format example:

MAG1 SIGNAL1 DEPTH1(m) ALTITUDE1(m) DATE TIME GPS_LON GPS_LAT SHIFT_LON SHIFT_LAT ATARGETS NMAGS LON_MAG1 LAT_MAG1 GPS_QC GPS_HEIGHT LINE LAYBACK(m)

56636.708 1211.000 1.318 4.958 06/07/14 08:43:12.562 142.0003269 56.0009905 142.0003269 56.0009905 0 1 142.0003269 56.0009905 11 0.693 0 0.00

56636.784 1182.000 1.250 4.958 06/07/14 08:43:12.671 142.0003243 56.0009903 142.0003243 56.0009903 0 1 142.0003243 56.0009903 11 0.695 0 0.00

INT file fields: Mag Signal Depth Altitude DateM DateD DateY TimeH TimeM TimeS GpsLon GpsLat ShiftLon ShiftLat Atargets Nmags LonMag LatMag GpsFixQuality GpsHgtGeoid Line Layback.

if key==2, then *.INT file format example (Route added):

MAG1 SIGNAL1 DEPTH1(m) ALTITUDE1(m) DATE TIME GPS_LON GPS_LAT SHIFT_LON SHIFT_LAT ATARGETS NMAGS LON_MAG1 LAT_MAG1 GPS_QC GPS_HEIGHT LINE ROUTE LAYBACK(m)

56636.708 1211.000 1.318 4.958 06/07/14 08:43:12.562 142.0003269 56.0009905 142.0003269 56.0009905 0 1 142.0003269 56.0009905 11 0.693 0 NO_PLANNED_ROUTE 0.00

56636.784 1182.000 1.250 4.958 06/07/14 08:43:12.671 142.0003243 56.0009903 142.0003243 56.0009903 0 1 142.0003243 56.0009903 11 0.695 0 NO_PLANNED_ROUTE 0.00

INT file fields:

Mag Signal Depth Altitude DateM DateD DateY TimeH TimeM TimeS GpsLon GpsLat ShiftLon ShiftLat Atargets Nmags LonMag LatMag GpsFixQuality GpsHgtGeoid Line Route Layback.

Function Example:

```
>> HInt=gMagyReadG88Int('c:\temp\123.INT',1);
```

2.2 Read MagLog's *.mag file for single magnetometer

function HMag=gMagyReadG88Mag(fName,K)

Read data from text file with was created MagLog Geometrics program for G882 single magnetometer (*.MAG)

Parameters:

fName – reading file name;

K – Altimeter and Depth sensor coefficients: [AltScale AltBias DepthScale DepthBias];

HMag – data structure with reading fields: G88Koeff, CompDay, CompTime, MagAbsT, MagPrecSignal, Depth, Altitude.

*.MAG file format example:

```
$ 56637.438,1204,0139,0659  06/07/14 08:43:13.359
$ 56637.529,1208,0139,0640  06/07/14 08:43:13.468
```

Mag file columns:

1) total magnetic field; 2) signal; 3) depth; 4) altitude; 5) computer date; 6) computer time.

Function Example:

```
>> HMag=gMagyReadG88Mag('c:\temp\123.MAG',[0.010 -1.55 0.064255 -2.55]);
```

2.3 Read MagLog's *.mag file for TVG-frame

function HMag=gMagyReadG88tvgMag(fName,K)

Read data from text file with was created MagLog Geometrics program for G882-TVG double magnetometer (*.MAG)

Parameters:

fName – reading file name;

K – Altimeter and Depth sensor coefficients: [AltScale1 AltBias1 DepthScale1 DepthBias1; AltScale2 AltBias2 DepthScale2 DepthBias2];

HMag – data structure with reading fields: G88Koeff, CompDay, CompTime, MagAbsT, MagPrecSignal, Depth, Altitude

*.MAG file format example:

```
$ 35179.070,1047,0603,1550, 35177.225,1121,0574,0112  09/12/07 10:41:21.791
$ 35178.779,1047,0610,1567, 35176.918,1123,0574,0112  09/12/07 10:41:21.901
```

Mag file columns:

1) total magnetic field1; 2) signal1; 3) depth1; 4) altitude1; 5) total magnetic field2; 6) signal2; 7) depth2; 8) altitude2; 9) computer date; 10) computer time.

Function Example:

```
>> HMag=gFMagyReadG88tvgMag('c:\temp\123.MAG',[0.010 -1.55 0.064255 -2.55; 0.010 -1.55 0.064255 -2.55]);
```

2.4 Read MagLog's *.gps file

function HGps=gMagyG88GpsRead(fName,CompTimeLocShift)

Read GPGGA-messages from text file (*.GPS) with was created MagLog Geometrics program.

Parameters:

fName – reading file name;

CompTimeLocShift – Computer time minus Utc_Gps time (in seconds).

HGps – output data structure with fields:

CompDay, CompTime, GpsDay, GpsTime, CompTimeLocShift, CompTimeDelta, CompTimeShift, GpsLat, GpsLon, GpsFixQuality, GpsSatNum, GpsHorizDilution, GpsAltSea, GpsHgtGeoid, GpsDgpsUpdate, GpsDgpsId

*.GPS file format example:

$GPGGA,234303.85,5600.000366,N,14200.008168,E,11,12,1.0,00000.709,M,00000.000,M,0.00,*71 06/07/14 08:43:13.562

$GPGGA,234304.87,5600.000551,N,14200.006776,E,11,12,1.0,00000.675,M,00000.000,M,0.00,*7B 06/07/14 08:43:14.578

*.GPS fields:

S Utc Lat LatC Lon LonC FixQuality SatNum HorizDilution AltSeaGps AltSeaC HgtGeoid HgtGeoidC DgpsUpdate DgpsId Cheksum DateM DateD DateY TimeH TimeM TimeS.

Where $GPGGA include:

$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,08,1004*47

123519 – Fix taken at 12:35:19 UTC.

4807.038,N – Latitude 48 deg 07.038' N.

01131.000,E – Longitude 11 deg 31.000' E.

Fix quality: 0=invalid; 1=GPS fix (SPS); 2=DGPS fix; 3=PPS fix; 4=Real Time Kinematic; 5=Float RTK; 6=estimated (dead reckoning) (2.3 feature); 7=Manual input mode; 8=Simulation mode.

08 – Number of satellites being tracked.

0.9 – Horizontal dilution of position.

545.4,M – Altitude, Meters, above mean sea level.

46.9,M – Height of geoid (mean sea level) above WGS84 ellipsoid.

08 – Time in seconds since last DGPS update.

1004 – DGPS station ID number.

*47 – Checksum data, always begins with *.

Function Example:

```
>> HMag=gMagyG88GpsRead('c:\temp\123.GPS',10*3600);
```

3 Magnetic field modeling

3.1 Normal field's components calculation

function XYZm=gMagyModNormal(siz,parM,keyC)

Create Normal magnetic field's X,Y,Z components matrix for [Module,Declination,Inclination] parameters.

Parameters:

siz – elements number for output matrix (3xN vector) creation;

XYZm – 3-rows matrix with magnetic field components (X,Y and Z);

parM – normal field [T,D,I] values; for example [51872,-10.29,61.18] for 46.61N, 141.90E, 2016 year.

keyC – coordinate system key for XYZm: 'E' is Earth magnetic field coordinate system; 'M' is Magnetization coordinate system (***Figure 1***).

Function Example (***Figure 4***):

```
>> [X,Y]=meshgrid(-20:.1:20,-30:.1:30);
>> XYZn= gMagyModNormal (numel(X),[51872,-10.29,61.18],'E');
>> Zn=reshape(XYZn(3,:),size(X));mesh(X,Y,Zn);axis ij;
```

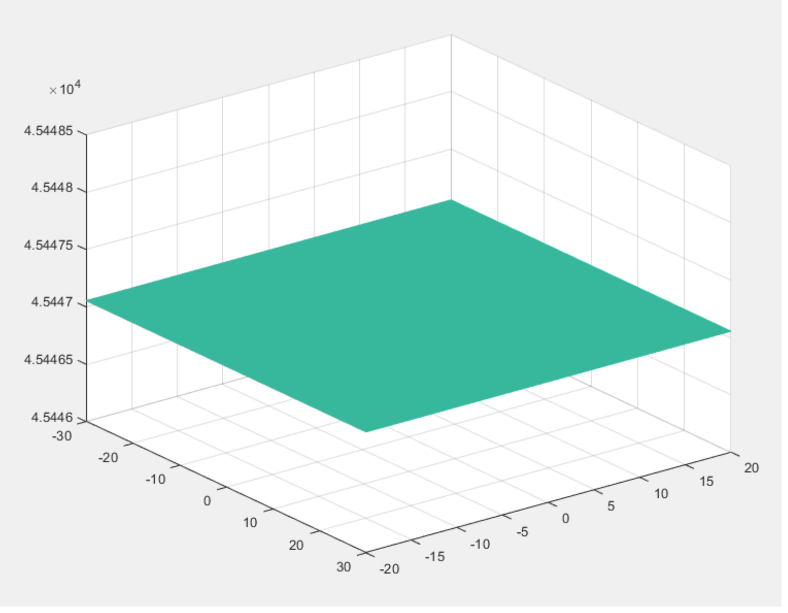

Figure 4 gMagyModNormal using result (example)

3.2 Dipole field modeling

function XYZm=gMagyModDipol(X,Y,Z,parM,keyC)

Calculate magnetic field's X,Y,Z components for dipole with parameters [Moment,Declination,Inclination,X_dipole_coord,Y_dipole_coord,Z_dipole_coord].

Parameters:

X,Y,Z – rows with coordinates for field calculation, the "space coordinate system" used;

XYZm – 3-rows matrix with magnetic field components (X,Y and Z);

parM – dipole's parameters [Moment (M*mu0/4/pi), Declination, Inclination, X_dipole_coord, Y_dipole_coord, Z_dipole_coord];

For example [53.75,-10.29,61.18,0,0,0], for 46.61N, 141.90E, 2016 year;

keyC – coordinate system key for parM and XYZm: 'E' is Earth magnetic field coordinate system; 'M' is Magnetization coordinate system.

Induced magnetization Moment_intensity: Mi=4pi*ksi*Tnorm*r^3/300 (the induced magnetization is set along Earth magnetic field).

Function Example (***Figure 5***):

```
>> [X,Y]=meshgrid(-20:.1:20,-30:.1:30);Z=repmat(4,size(X));
>> XYZm=gMagyModDipol(X(:)',Y(:)',Z(:)',[mmm,-10.29,61.18,0,0,0],'E');
>> Zm=reshape(XYZm(3,:),size(X));figure(1);mesh(X,Y,Zm);axis ij;
>> XYZn=gMagyModNormal(length(X(:)),[51827,-10.29,61.18],'E');
>> Zn=reshape(XYZn(3,:),size(X));figure(2);mesh(X,Y,Zn);axis ij;
>> XYZmn=XYZm+XYZn;Tmn=sqrt(XYZmn(1,:).^2+XYZmn(2,:).^2+XYZmn(3,:).^2)-51827;
>> T=reshape(Tmn,size(X));figure(3);mesh(X,Y,T);axis ij;
>> figure(4);contour(X,Y,T,20,'ShowText','on');axis equal;axis ij;
```

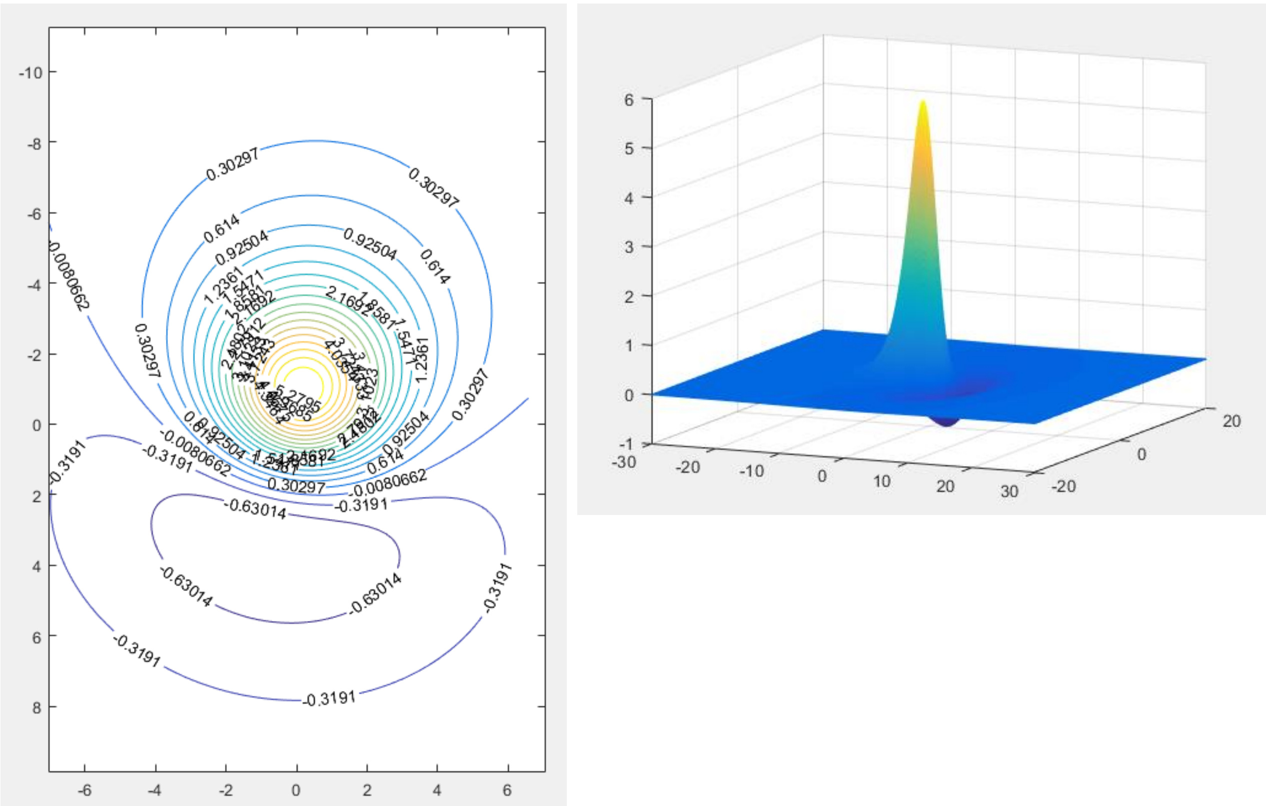

Figure 5 gMagyModDipol using result (example)

3.3 Horizontal Cylinder field modeling Warning! Need to check

function [XYZm]= gMagyModCylinder(X,Y,Z,parM,keyC)

Calculate magnetic field's X,Y,Z components for Horizontal Cylinder with parameters [Moment, Declination, Inclination, X1_cylind_coord, Y1_cylind_coord, Z1_cylind_coord, X2_cylind_coord, Y2_cylind_coord].

Parameters:

X,Y,Z – rows with field calculation coordinates, the "space coordinate system" used;

XYZm – 3-rows matrix with magnetic field components (X,Y and Z);

parM – cylinder parameters [Moment_intensity, Moment_azimuth_(declination), Moment_inclination, X1_cylinder, Y1_cylinder, Z1_cylinder, X2_cylinder, Y2_cylinder]; for example [306,-10.29,61.18,0,0,1,1,1] for 46.61N, 141.90E, 2016 year;

keyC – coordinate system key for XYZm: 'E' is Earth magnetic field coordinate system; 'M' is Magnetization coordinate system.

M*mu0/4/pi calculation: (R1(0.6m)^2-R2(0.58m)^2)*ksi(100)*B(51872nT)/4=30604; %the calculated field in nT; inductive magnetization along Earth magnetic field.

Function Example 1 (***Figure 6***):

```
>> [X,Y]=meshgrid(-20:.1:20,-30:.1:30);Z=repmat(2,size(X));
>> XYZm= gMagyModCylinder(X(:)',Y(:)',Z(:)',[53.75,-10.29,61.18,0,0,0,0,30]);
>> Zm=reshape(XYZm(3,:),size(X));figure(1);mesh(X,Y,Zm);axis ij;
>> XYZn=gMagyModNormal(numel(X),[51872,-10.29,61.18]);
>> XYZmn=XYZm+XYZn;Tmn=sqrt(XYZmn(1,:).^2+XYZmn(2,:).^2+XYZmn(3,:).^2)-51872;
>> T=reshape(Tmn,size(X));figure(3);mesh(X,Y,T);axis ij;
```

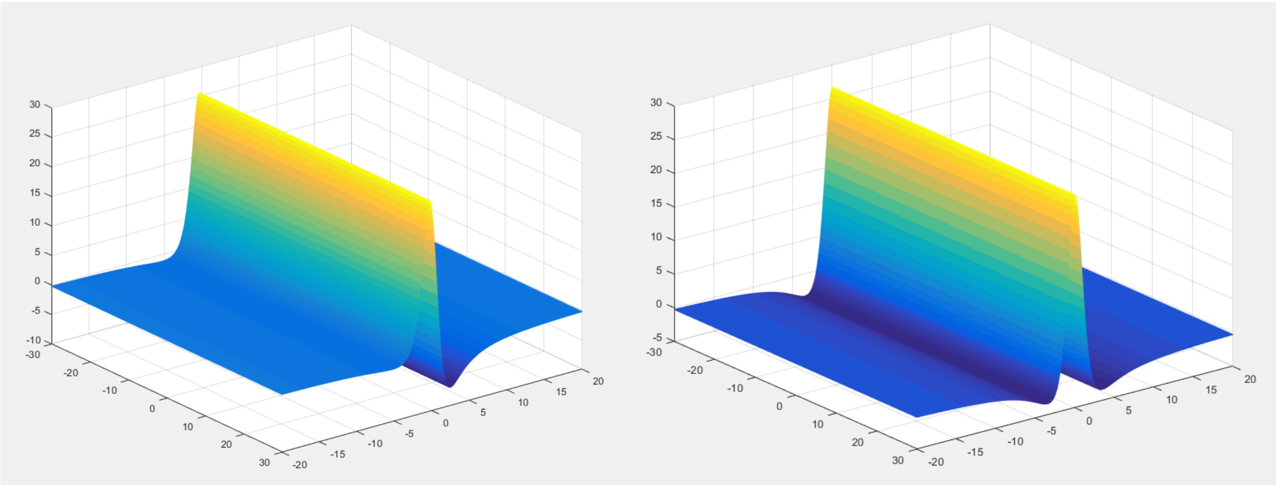

Figure 6 gMagCylinder using result (example 1)

Function Example 2 (***Figure 7***):

```
>> [X,Y]=meshgrid(-20:.1:20,-30:.1:30);Z=repmat(2,size(X));
```

```
>> XYZn=gMagyModNormal(numel(X),[51872,-10.29,61.18]);
>> figure(100);hold on;
>> for n=0:60:350;
        XYZm=gMagyModCylinder(X(:)',Y(:)',Z(:)',[53.75,-10.29,n,0,0,0,0,30]);
        XYZmn=XYZm+XYZn;Tmn=sqrt(XYZmn(1,:).^2+XYZmn(2,:).^2+XYZmn(3,:).^2)-51872;
        T=reshape(Tmn,size(X));plot(X(5,:),T(5,:));
end;
```

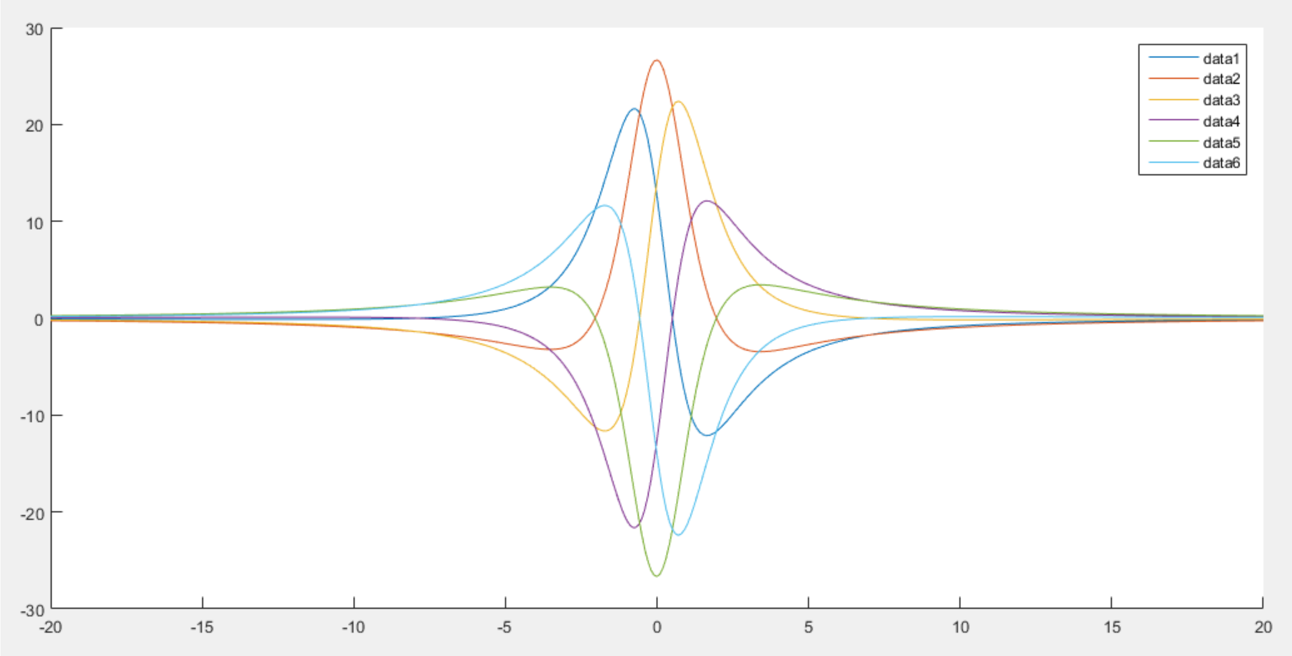

Figure 7 gMagyModCylinder using result for different magnetization direction (example 2)

4 Graphics

4.1 Draw wiggle

function [p1,p2]=gMagyDrawWiggle(figN,X,Y,T,ang)

Draw wiggle. Magnetic colors: blue is minus, red is plus.

Parameters:

figN – figure number;

X – wiggle track x-coordinates;

Y – wiggle track y-coordinates;

T – wiggle amplitudes;

ang – amplitudes rotation angle; if zero, then T along y-axis; plus is counterclockwise rotation;

[p1,p2] – pointers to red patch and blue patch.

The coordinate system:

^ x(forward/roll)

|

o---> y(right/pitch)

Function Example:

```
>> gMagyDrawWiggle(1,[1 2 3 4 5 6 7 8 9],[5 5 5 5 5 5 5 5 5],[1 1 2 2 3 2 2 1 1],-45);
```

4.2 Draw wiggle with holes

function gMagyDrawWiggleMask(figN,X,Y,T,mask,AText,ang)

Draw wiggle with holes for Mask-values. Magnetic colors: blue is minus, red is plus.

Parameters:

figN – figure number;

X – wiggle track x-coordinates;

Y – wiggle track y-coordinates;

T – wiggle amplitudes;

mask – holes mask;

AText – writing text with a fist wiggle point;

ang – amplitudes rotation angle; if zero, then T along y-axis; plus is counterclockwise rotation.

Function Example:

```
>> gMagyDrawWiggleNan(1,[1 2 3 4 5 6 7 8 9],[5 5 5 5 5 5 5 5 5],[1 1 2 2 3 2 2 1 1],true(1,9),'E95',-45);
```

4.3 Targets pick

function fM=gMagyPickHandle(Prof,WF,WP,ProfNum,f)

Interactive Survey Lines parts (Targets) selection. Each Target's data includes start and end makers time and ID.

Parameters:

PR – Survey Lines structure;

WF – field names for Wiggle drawing
{PrName,DayName,TimeName,XName,YName,{GraphsFieldsName},QMaskName,QMaskValue}, for example: {'PrName','Mag.CompDay','Mag.CompTime','Mag.GpsEL','Mag.GpsNL', {'Mag.DepthRaw','Mag.AltitudeRaw'},'Mag.QMask',1024+2048}

WP – WiggleParam=[Row number of Value; divider A=Value/divScl; limit for wiggle clipping from -lim to lim; wiggle direction for plane], for example: [1,0.2,70,0];

ProfNum – profiles numbers for wiggle;

f – handle for figure or figure number.

Keys for "pick mode" (target window must be active):

 – press button to start piking;

q – switch the MapWindow and GraphsWindow; b – stop piking (exit from MapWindow/GraphsWindow);

z – zoom mode on/off; x – pan mode on/off; c – datacursormode on/off;
need to realized >> 0..9 – save user-markers (datacursor) form 0 to 9 for selected target; user-markers showing only when target is selected.

s – select Line when MapWindow picking; s – select Graph for piking/axis_scale when GraphsWindow picking;

LMB – set marker_1; RMB – set marker_2; MMB – create target from marker_1 to marker_2 and input target's number;

a – Select Target for current Line or Deselect Target; new target (MMB created) is auto-selected;

e – erase selected target; t – input comment for selected target.

Function Example:

```
>> fMap=gMagyPickHandle(PR,{'PrName','Mag.CompDay','Mag.CompTime','Mag.GpsEL',
'Mag.GpsNL',{'Mag.MagAbsTSHi','Mag.MagAbsTS','Mag.AltitudeS','Mag.DepthS'},'Mag.QMask',1024
+2048},[1,0.2,70,0],1:74,1);
```

Example, take data:

```
>> PR=getappdata(fMap,'PR');
```

The PR-variable (when data taken) includes field PR{...}.Targ.Dat for survey lines where targets were marked. There are a number of rows:

[tm1 tm2 CompDay1 CompDay2 CompTime1 CompTime2 TargetId]

where,

tm1, tm2 – numbers for marker_1 and marker_2;

CompDay1, CompDay2, CompTime1, CompTime2 – computer's day and time for marker_1 and marker_2;

TargetId – target's number.

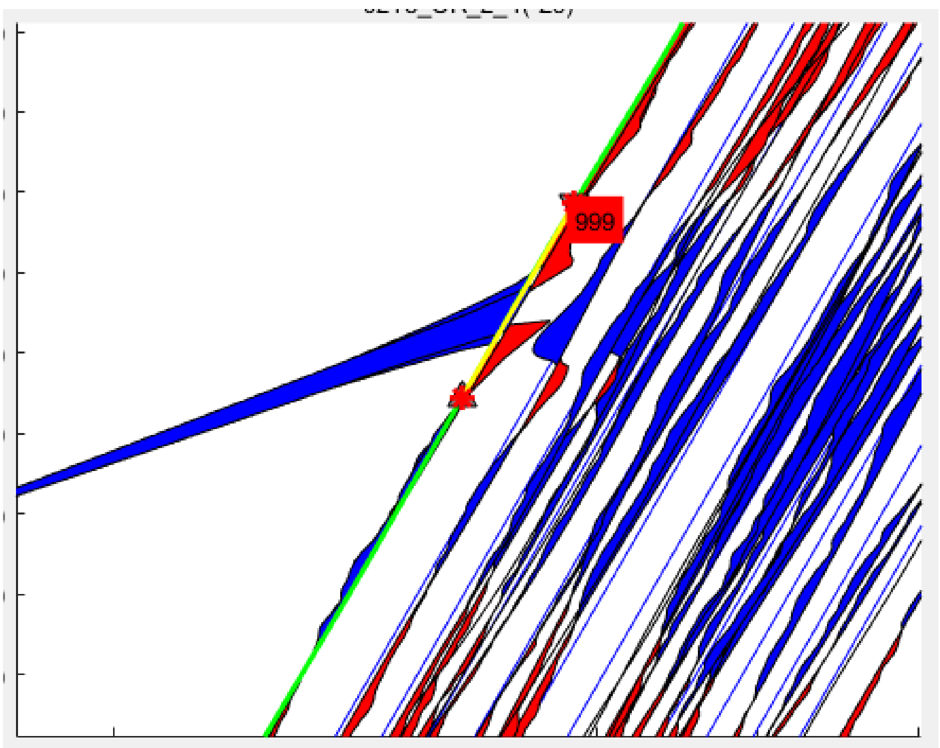

Figure 8 Zoomed MapWindow with target 999 picked

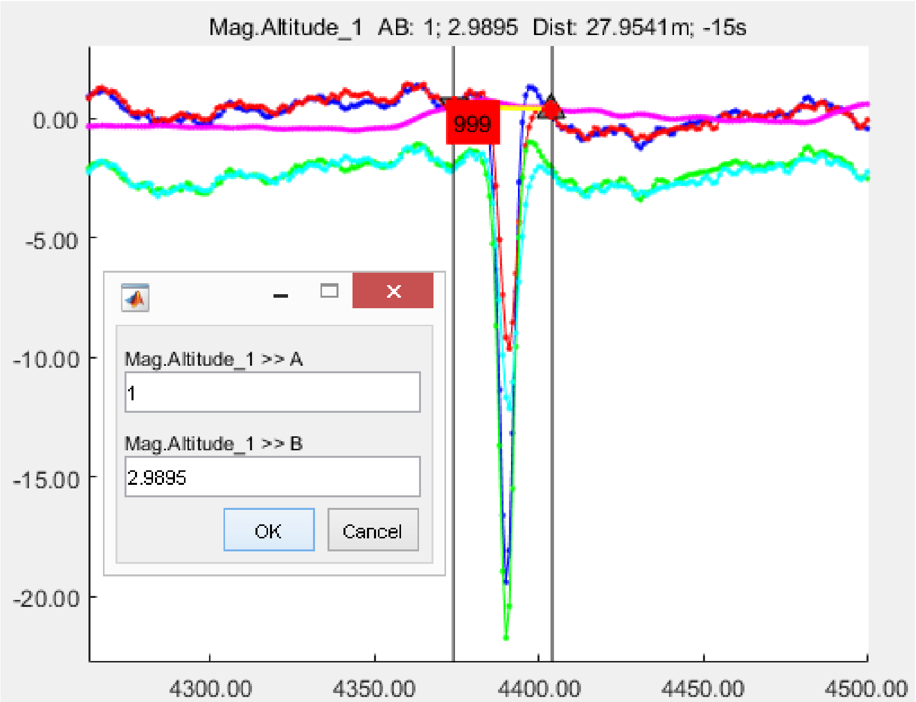

Figure 9 Zoomed GraphsWindow with target 999 picked and graph/scale selected window (key "s")

Citation

1)