



GACAD

EXPORT DATA TO AUTOCAD SCRIPTS

VOL. 03-01

Ivan V. Dmitriev
06.12.2021

Contents

1. gAcad general description4

2. Functions.....5

 2.1 Set color5

 2.2 Set zoom.....5

 2.3 Create layer5

 2.4 Send “osnap” command6

 2.5 Send command6

 2.6 Draw Circle6

 2.7 Draw Text.....7

 2.8 Draw Polyline.....7

 2.9 Draw Polygon.....8

 2.10 Insert Image.....8

 2.11 Insert Geo-referenced Image.....8

 2.12 Draw Track-plot with Mask9

 2.13 Draw Track-plot with Graph10

 2.14 Draw Track-plot with Wiggle11

 2.15 Draw Track-plot with Wiggle and Mask.....12

 2.16 Iges import14

Citation.....15

Tables list

<i>Table 1.1</i> gAcad functions	4
---	---

Figures list

<i>Figure 2.1</i> gAcadGraph example result (two scripts in AutoCad window)	11
<i>Figure 2.2</i> gAcadWiggle example result (script in AutoCad window)	12
<i>Figure 2.3</i> gAcadWiggleMask example result (script in AutoCad window)	13
<i>Figure 2.4</i> gAcadWiggleMask function result (script in AutoCad window); draw magnetic survey data along survey lines.....	13
<i>Figure 2.5</i> AutoCAD window with circles, polylines and lines.....	14

1. gAcad general description

Create AutoCad scripts for English Language with elementary figures. There are several functions (*Table 1.1*); each of them sends strings to script-file.

In accordance AutoCad “snap”, it is recommend to use function `gAcadZoom(fId,[0 0 0.0001],4)`; first in the script body or `gAcadOsnap(fId,'off')`; or manual control the snap is switch off. Otherwise incorrect script drawing is possible.

Table 1.1 gAcad functions

Function name	Function description
<code>gAcadColor</code>	Set color
<code>gAcadZoom</code>	Set zoom
<code>gAcadLayerMake</code>	Create layer
<code>gAcadOsnap</code>	Send “osnap” command
<code>gAcadSendCommand</code>	Send “osnap” command
<code>gAcadCircle</code>	Draw Circle
<code>gAcadText</code>	Draw Text
<code>gAcadPline</code>	Draw Polyline
<code>gAcadPolygon</code>	Draw one poly-line with Hatch
<code>gAcadImage</code>	Insert image in XY coordinates with scale m
<code>gAcadGeoReffImage</code>	Insert image using World-file data
<code>gAcadTrackMask</code>	Draw Track-plot with Mask
<code>gAcadGraph</code>	Draw Track-plot with Graph
<code>gAcadWiggle</code>	Draw Wiggle in XY coordinates
<code>gAcadWiggleMask</code>	Draw Wiggle in XY coordinates with Mask
<code>gAcadIgesRead</code>	Read circles' center coordinates and polylines coordinates (not completed) from iges-file (AutoCad's export).

2. Functions

2.1 Set color

function gAcadColor(fId,ColorRGB)

Write "default rgb-color" to AutoCad script file.

Parameters:

fId – file identifier;

ColorRGB – ColorRGB- [R G B] color vector (0..255) or symbol y,m,c,r,g,b,w,k.

Function Example:

```
>> X=[1 2 3 5]';Y=[4 5 7 10]'; fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);  
>> gAcadColor(fId,[255 0 0]);gAcadCircle(fId,X,Y,1,[2 2 0]);  
>> gAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);fclose(fId);
```

2.2 Set zoom

function gAcadZoom(fId,XYM,dgt)

Write "zoom" to AutoCad script file. Warning!!! There is some problem with "zoom" command for AutoCadCivil.

Parameters:

fId – file identifier;

XYM – [x-coordinate (right/E), y-coordinate (up/N), scale] for zoom;

dgt – printing digits after decimal points for XYM column (if isempty - dgt=5).

Function Example:

```
>> fId=fopen('c:\temp\113.scr','w');gAcadZoom(fId,[0 0 0.0001],4);fclose(fId);
```

2.3 Create layer

function gAcadLayerMake(fId,layerName)

Write to AutoCad script file: make layer (create and set active).

Parameters:

fId – file identifier;

layerName – layer name; layer 1)will be create and set active; 2)if layer exist, it is set active and turned on.

Function Example:

```
>> fId=fopen('c:\temp\112.scr','w');gAcadLayerMake(fId,'layer1');fclose(fId);
```

2.4 Send “osnap” command

function gAcadOsnap(fId,key)

Write "-osnap" command with Key to AutoCad script file. Usually used for switch off snap for true script execution.

Parameters:

key – key for "-osnap" command: 'off','end', etc.

Function Example:

```
>> X=[1 2 3 5];Y=[4 5 7 10];fId=fopen('c:\temp\112.scr','w');gAcadOsnap(fId,'off');gAcadColor(fId,[255 0 0]);gAcadCircle(fId,X,Y,1,[2 2 0]);gAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);fclose(fId);
```

2.5 Send command

function gAcadSendCommand(fId,keyZ)

Write defined command+CR+LF to AutoCad script file.

Parameters:

keyZ – command send to AutoCad.

Function Example:

```
>> fId=fopen('c:\temp\112.scr','w');gAcadSendCommand(fId,'-osnap off');gAcadSendCommand(fId,'-color truecolor 10,10,0');fclose(fId);
```

2.6 Draw Circle

function gAcadCircle(fId,X,Y,R,dgt)

Write to AutoCad script file: draw circles in X(:),Y(:) coordinates, with radius R(:).

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

R – radius (scalar or vector);

dgt – printing digits after detimal points for X, Y and R columns (if isempty - dgt=[5 5 1]).

AutoCad script line example: `Circle 582160.91000,5306794.61000 1`

Function Example:

```
>> X=[1 2 3 5]';Y=[4 5 7 10]';  
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);  
>> gAcadColor(fId,[255 0 0]);gAcadCircle(fId,X,Y,1,[2 2 0]);  
>> gAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);fclose(fId);
```

2.7 Draw Text

function gAcadText(fId,X,Y,FontSize,TextRotAngle,AText,dgt)

Write to AutoCad script file: draw texts in X(:),Y(:) coordinates, with size FontSize(:) and angle TextRotAngle(:).

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

FontSize – text's font size (scalar or vector);

TextRotAngle – text's rotation angle (scalar or vector);

AText – writing text: 1) vector with a numbers; 2) cell-vector with a strings; 3) single char string for single point.

dgt – printing digits number after decimal points for X,Y and AText (if isempty - dgt=[5 5 0]).

AutoCad script line example: `Text 1.00000,1.00000 5 0 1`

Function Example:

```
>> X=[1 2 3 5];Y=[4 5 7 10];
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);
>> gAcadColor(fId,[255 0 0]);gAcadCircle(fId,X,Y,1,[2 2 0]);
>> gAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);fclose(fId);
```

2.8 Draw Polyline

function gAcadPline(fId,X,Y,dgt)

Write to AutoCad script file: draw one poly-line in X(:),Y(:) coordinates.

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

dgt – printing digits after decimals points for X and Y (if isempty - dgt=[5 5]).

AutoCad script line example:

`pline 1.00000,1.00000 2.00000,1.00000 3.00000,1.00000 4.00000,1.00000 5.00000,1.00000`

Function Example:

```
>> X=[1 2 3 5 6 7 8];Y=[1 2 4 7 11 16 22];
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);
>> gAcadColor(fId,[255 0 0]);gAcadPline(fId,X,Y,[2 2]);fclose(fId);
```

2.9 Draw Polygon

function gAcadPolygon(fId,X,Y,ColorLine,ColorHatch,dgt)

Write to AutoCad script file: draw one poly-line with Hatch in X(:),Y(:) coordinates. The poly-line is auto closed (first point will be added to end).

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

ColorLine – line [R G B] color vector; if isempty, than not changed;

ColorHatch – hatch [R G B] color vector; if isempty, than not changed;

TransparencyHatch – hatch transparency in percent from 0 to 100 (if isempty, than not set);

dgt – printing digits after detimal points for X and Y (if isempty - dgt=[5 5]).

Using functions: gAcadColor.

Function Example:

```
>> X=[1 2 3 5 6 7 8]';Y=[1 2 4 7 11 16 22]';  
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);  
>> gAcadPolygon(fId,X,Y,[255 0 0],[0 255 0],50,[2 2]);fclose(fId);
```

2.10 Insert Image

function gAcadImage(fId,fR,XY,m,Ang)

Write to AutoCad script file: insert image in XY coordinates with scale m (mean that horizontal images' length is 1). Tiff-image convert to png format.

Parameters:

fId – file identifier;

fR – path to image file (string);

XY – left down images' corner coordinates [x-coordinate(right/E) y-coordinate(up/N)];

m – scale (mean that horizontal images' length is 1);

Ang – rotation angle (around XY, left rotation in degree).

AutoCad script line example:

Function Example:

```
>> fId=fopen('c:\temp\112.scr','w');  
>> gAcadImage(fId,'c:\temp\2017_12_04_doc20171206203403_004.jpg',[10 10],1,0); fclose(fId);
```

2.11 Insert Geo-referenced Image

function gAcadGeoReffImage(fId,fR)

Write to AutoCad script file: insert image to XY coordinates using World-file (tfw, jfw and same). Tiff image convert to png format.

Parameters:

fId – file identifier;

fR – path to image file (string), includes image extension.

Function Example:

```
>> fId=fopen('c:\temp\112.scr','w');gAcadGeoReffImage(fId,'c:\temp\Prod02.tif');fclose(fId);  
>> fId=fopen('c:\temp\112.scr','w');for n=1:23,gAcadGeoReffImage(fId,['c:\temp\1\' num2str(n,'%02d')  
'tif']);end;fclose(fId);
```

===== World-file lines =====

Line1_A: x-component of the pixel width (x-scale);

Line2_D: y-component of the pixel width (y-skew);

Line3_B: x-component of the pixel height (x-skew);

Line4_E: y-component of the pixel height (y-scale), typically negative.

Line5_C: x-coordinate of center of upper left pixel;

Line6_F: y-coordinate of center of upper left pixel.

=====

2.12 Draw Track-plot with Mask

function gAcadTrackMask(fId,X,Y,mask,FontSize,TextRotAngle,AText,dgt)

Write to AutoCad script file: draw one TrackPlot in X(:),Y(:) coordinates with out-of-data by mask.

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

mask – mask for each point; 0/false is not drawing point; the poly-line will cut for a number of segments;

FontSize – text's font size (scalar or vector);

TextRotAngle – text's rotation angle (scalar or vector);

AText – writing text string “in start point” for each segment;

dgt – printing digits number after detimal points for X,Y and AText (if isempty - dgt=[5 5 0]).

Using functions: gAcadPline, gAcadText.

Function Example:

```
>> x=[1 2 3 4 5 6 7 8 9 10];y=[1 3 4 1 5 6 1 1 2 3];m=[1 1 1 1 0 0 1 1 1 1];  
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);gAcadColor(fId,[255 0 0]);  
>> gAcadTrackMask(fId1,x,y,m,1,90,'E95',[2 2 1]);fclose(fId);
```

AutoCad script line example:

```
zoom c 0.0000,0.0000 0.0001
```

```
-layer m "Track_GpsRaw"
```

```
pline 1.00,1.00 2.00,3.00 3.00,4.00 4.00,1.00
```

```
text 1.00,1.00 1 90 E95
```

```
pline 7.00,1.00 8.00,1.00 9.00,2.00 10.00,3.00
```

```
text 7.00,1.00 1 90 E95
```

2.13 Draw Track-plot with Graph

function gAcadGraph(fId,X,Y,T,Ang,ColorGraph,ColorTrack,dgt)

Write to AutoCad script file: draw plot T relativeiy to trackplot in X,Y coordinates.

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

T – graph-line value vector;

Ang – plot-line rotation angle (one number; 0- up/N, right/clockwise rotation sign is +);

ColorGraph – graph [R G B] color vector; if isempty, than not changed;

ColorTrack – track-line [R G B] color vector; if isempty, than not changed;

dgt – printing digits after decimal points for X and Y (if isempty - dgt=[5 5]). Using functions: gAcadColor.

Function Example (*Figure 2.1*):

```
>> X=[1 2 3 4 5 6 7 8];Y=[1 2 4 7 11 16 22 30];T=[0 1 5 1 -2 -4 -4 -2];
```

```
>> fId=fopen('c:\temp\1.scr','w');gAcadZoom(fId,[0 0 0.0001],4);
```

```
>> gAcadGraph(fId,X,Y,T,0,[255 0 0],[0 255 0],[2 2]);fclose(fId); %red
```

```
>> fId=fopen('c:\temp\2.scr','w');gAcadZoom(fId,[0 0 0.0001],4);
```

```
>> gAcadGraph(fId,X,Y,T,45,[255 255 0],[0 255 0],[2 2]);fclose(fId); %yellow
```

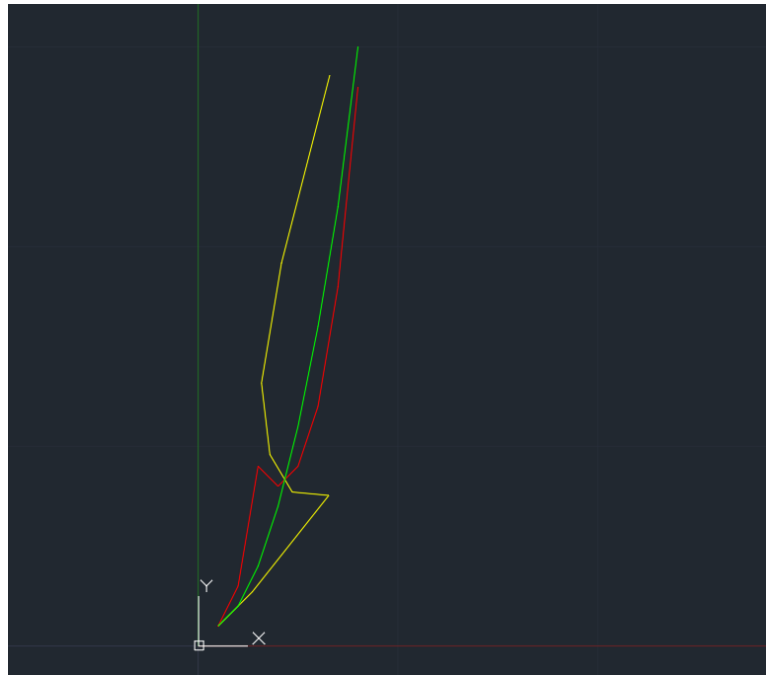


Figure 2.1 gAcadGraph example result (two scripts in AutoCad window)

2.14 Draw Track-plot with Wiggle

function gAcadWiggle(fId,X,Y,T,Ang,ColorLine,ColorHatch1,ColorHatch2, TransparencyHatch,FontSize,TextRotAngle,AText,dgt)

Write to AutoCad script file: draw Wiggle in X(:),Y(:) coordinates.

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

T – wiggle-line value vector;

Ang – wiggle-line rotation angle (scalar; 0- up/N, right/clockwise rotation sign is +);

ColorLine – wiggle-line color (scalar); if isempty, than not changed;

ColorHatch1 – wiggle-hatch1 (up zero) [R G B] color; if isempty, than not changed;

ColorHatch2 – wiggle-hatch2 (down zero) [R G B] color; if isempty, than not changed;

TransparencyHatch – hatch transparency in percent from 0 to 100 (if isempty, than not set);

FontSize – text's font size (scalar or vector);

TextRotAngle – text's rotation angle (scalar or vector);

AText – writing text string “in start point” for each segment;

dgt – printing digits number after detimal points for X,Y and AText (if isempty - dgt=[5 5 0]). Using functions: gAcadColor.

Function Example (*Figure 2.2*):

```
>> X=[1 2 3 4 5 6 7 8];Y=[1 2 4 7 11 16 22 30];T=[0 1 5 1 -2 -4 -4 -2];
```

```
>> fId=fopen('c:\temp\112.scr','w');gAcadZoom(fId,[0 0 0.0001],4);
>> gAcadWiggle(fId,X,Y,T,90,[0 0 255],[255 0 0],[0 255 0],50,1,90,'E95',[2 2 1]);fclose(fId);
```

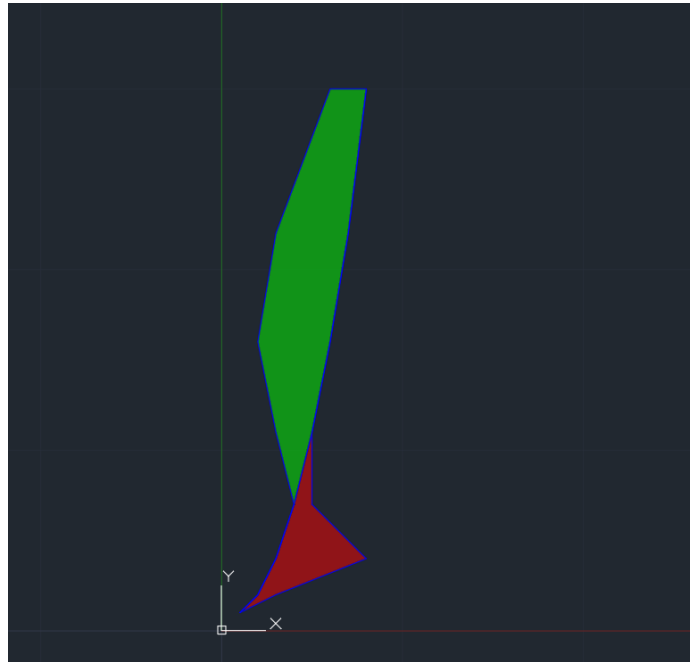


Figure 2.2 gAcadWiggle example result (script in AutoCad window)

2.15 Draw Track-plot with Wiggle and Mask

function gAcadWiggleMask(fId,X,Y,T,mask,Ang,ColorLine,ColorHatch1,ColorHatch2, TransparencyHatch,FontSize,TextRotAngle,AText,dgt)

Write to AutoCad script file: draw one WiggleToMap in X(:),Y(:) coordinates with out-of-data by mask.

Parameters:

fId – file identifier;

X – x-coordinate vector (right/E);

Y – y-coordinate vector (up/N);

T – wiggle-line value vector;

mask – mask for each point; 0/false is not drawing point; the Wiggle will cut for a number of segments;

Ang – wiggle-line rotation angle (scalar; 0- up/N, right/clockwise rotation sign is +);

ColorLine – wiggle-line color (scalar); if isempty, than not changed;

ColorHatch1 – wiggle-hatch1 (up zero) [R G B] color; if isempty, than not changed;

ColorHatch2 – wiggle-hatch2 (down zero) [R G B] color; if isempty, than not changed;

TransparencyHatch – hatch transparency in percent from 0 to 100 (if isempty, than not set);

FontSize – text's font size (scalar or vector);

TextRotAngle – text's rotation angle (scalar or vector);

AText – writing text string “in start point” for each segment;

dgt – printing digits number after detimal points for X,Y and AText (if isempty - dgt=[5 5 0]).

Using functions: gAcadWiggle.

Function Example (*Figure 2.3*):

```
>> X=[1 2 3 4 5 6 7 8];Y=[1 2 4 7 11 16 22 30];T=[0 1 5 1 nan nan -4 -2];mask=logical([1 1 1 1 0 0 1 1]);  
>> fld=fopen('c:\temp\112.scr','w');gAcadZoom(fld,[0 0 0.0001],4);  
>> gAcadWiggleMask(fld,X,Y,T,mask,90,[0 0 255],[255 0 0],[0 255 0],50,1,90,'E95',[2 2 1]);  
>> fclose(fld);
```

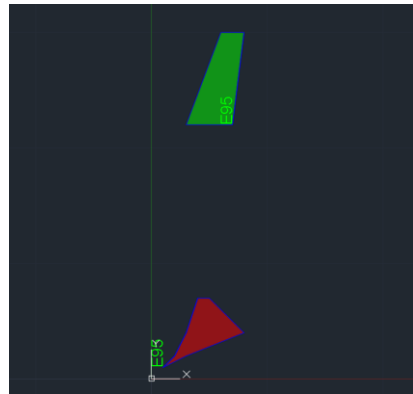


Figure 2.3 gAcadWiggleMask example result (script in AutoCad window)

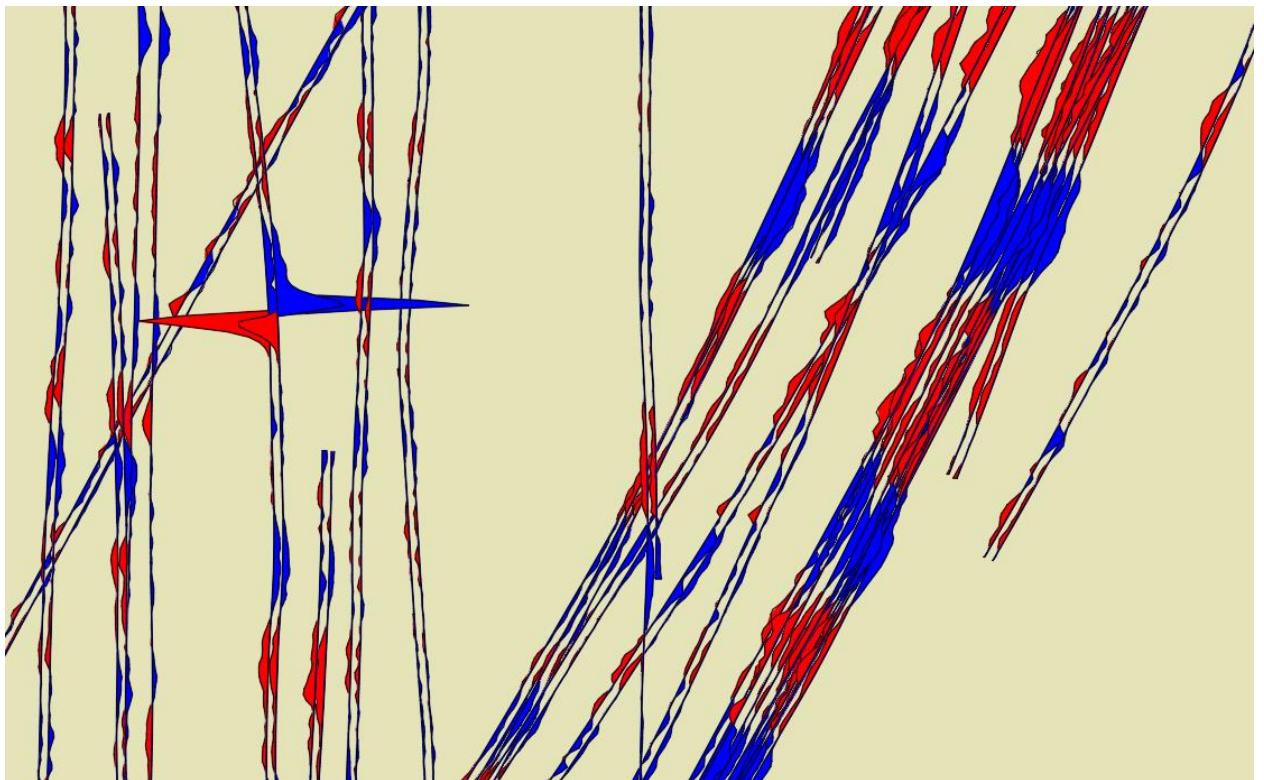


Figure 2.4 gAcadWiggleMask function result (script in AutoCad window); draw magnetic survey data along survey lines

2.16 Iges import

function [CircXY,LL,PL]=gAcadIgesRead(fName)

Read circles' center coordinates, Lines and Polylines coordinates from iges-file (AutoCAD's export).

Parameters:

fName – name of iges file;

CircXY – output xy-coordinates and R in three rows (mess 124 and 100);

LL – output xy-coordinates of start and end Line in PL-structure (mess 110);

PL – output xy-coordinates of polyline nods in PL-structure (mess 126).

Example:

```
>> [CircXY,LL,PL]=gAcadIgesRead('e:\021\S2_KP0-KP357_Seabedfeature_20210812i_Mag.iges');
```

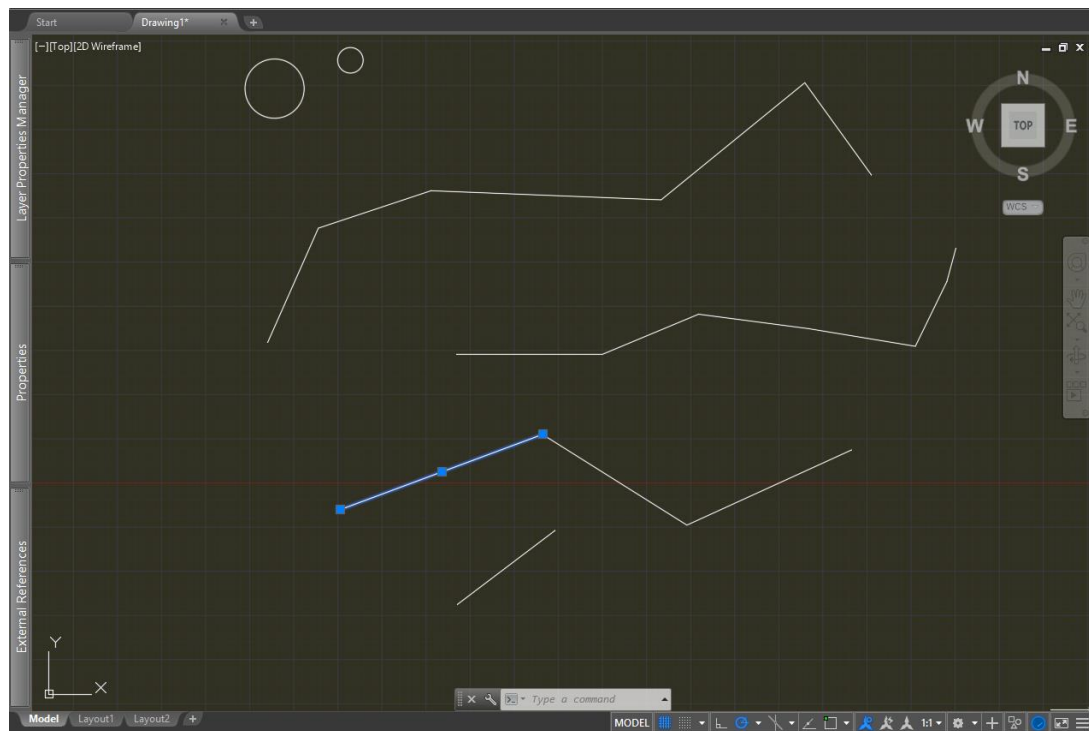


Figure 2.5 AutoCAD window with circles, polylines and lines

Citation

- 1) Autodesk AutoCAD 2017 help.