gMapIoAcad

Generating AutoCAD scripts for plotting graphics in Autodesk AutoCAD

Rev. 2.0

ge0mlib.com
23.03.2024

Contents

Revision History

Revision	Date	Description of Changes
1.0	2017-10-01	First issue document// Ivan D.
1.1	2019-11-02	Added a description of the new functions // Ivan D.
2.0	2024-03-23	Added a description of the new functions, edited the text, prepared English and Russian versions // Ivan D., Margarita B.

Figures list

Tables list

Introduction

The gMapIoAcad library contains MatLab / GNU Octave functions for AutoCAD scripts generating to plot graphics in Autodesk AutoCAD. AutoCAD scripts are text files that contain commands for plotting graphical objects written using the AutoCAD syntax [*1*].

Let's say we have an object track (drone, vessel) previously processed in MatLab. You can plot the track in AutoCAD without using the gMapIoAcad library by the following steps: (1) upload the track to a text file (points with X, Y coordinates and text labels of these points); (2.1) write a script in Lisp that reads the file data and prepares graphics based on coordinates to plot directly in AutoCAD or (2.2) plot additional graphics manually (circles for each nth point, point labels, etc.).

The gMapIoAcad library functions (***Table 0.1***, ***Figure 0.1***) allow you to generate an AutoCAD script directly from MatLab that:

-- creates a layer in AutoCAD with the track name;

-- draws a polyline based on the coordinates of the track in the created layer;

-- outlines every tenth point of the track with a circle;

-- signs every fiftieth point of the track;

-- draws a circle with a larger radius at the beginning of the track and signs the track name.

Table 0.1 gMapIoAcad functions

Function name	Function description
gMapIoAcadColor	Set a Color
gMapIoAcadZoom	Set a Zoom in the AutoCAD window
gMapIoAcadLayerMake	Create a new Layer or switch to an existing one
gMapIoAcadOsnap	Disable a Snap (send the “osnap” command)
gMapIoAcadSendCommand	Send an arbitrary AutoCAD Command (writing arbitrary text to a script)
gMapIoAcadCircle	Draw a Circle
gMapIoAcadText	Print a Text
gMapIoAcadPline	Draw a Polyline
gMapIoAcadPolygon	Draw a Polygon with Hatch
gMapIoAcadImage	Insert a bitmap Image into the specified XY coordinates with the scale defined
gMapIoAcadGeoReffImage	Insert a georeferenced bitmap Image using world-file data
gMapIoAcadTrackMask	Draw a Track-polyline using a mask for points that should not be drawn
gMapIoAcadGraph	Draw a Graph along the track-polyline
gMapIoAcadPLDraw	Plot a Track with additional graphics (marks and signatures for track points, track name)
gMapIoAcadWiggle	Draw a Wiggle along the track-polyline
gMapIoAcadWiggleMask	Draw a Wiggle along the track-polyline using a mask for sections that should not be drawn
gMapIoAcadIgesRead	Import the coordinates of elementary graphical objects from the *.iges file (*. iges is a format exported from AutoCAD). The file includes coordinates of the circle center and its radius, coordinates of the ends of the straight line, coordinates of the polyline nodes

The gMapIoAcad library functions write a set of commands for AutoCAD to a text file with the *.scr extension (AutoCAD script). To execute the script, the English version of AutoCAD must be installed. The script is sent for execution by dragging and dropping a script-file into the AutoCAD window. The gMapIoAcad library functions allow you to write commands to the AutoCAD script that plot elementary graphics (polylines, circles, polygons, text in certain coordinates), as well as commands that plot more complex graphical objects (georeferenced tiff, wiggle-type graphs).

If snap is enabled in AutoCAD before executing the script, this leads to the incorrect input and displacement of graphical objects due to the snapping of new drawn objects nodes to nodes of objects drawn earlier. To avoid this, it is recommended to use the function gMapIoAcadZoom(fId,[0 0 0.0001],4) at the beginning of the script (snap does not work for objects located outside the AutoCAD graphics window) or the gMapIoAcadOsnap(fId,'off') function (disabling a snap). It is also possible to disable a snap manually.

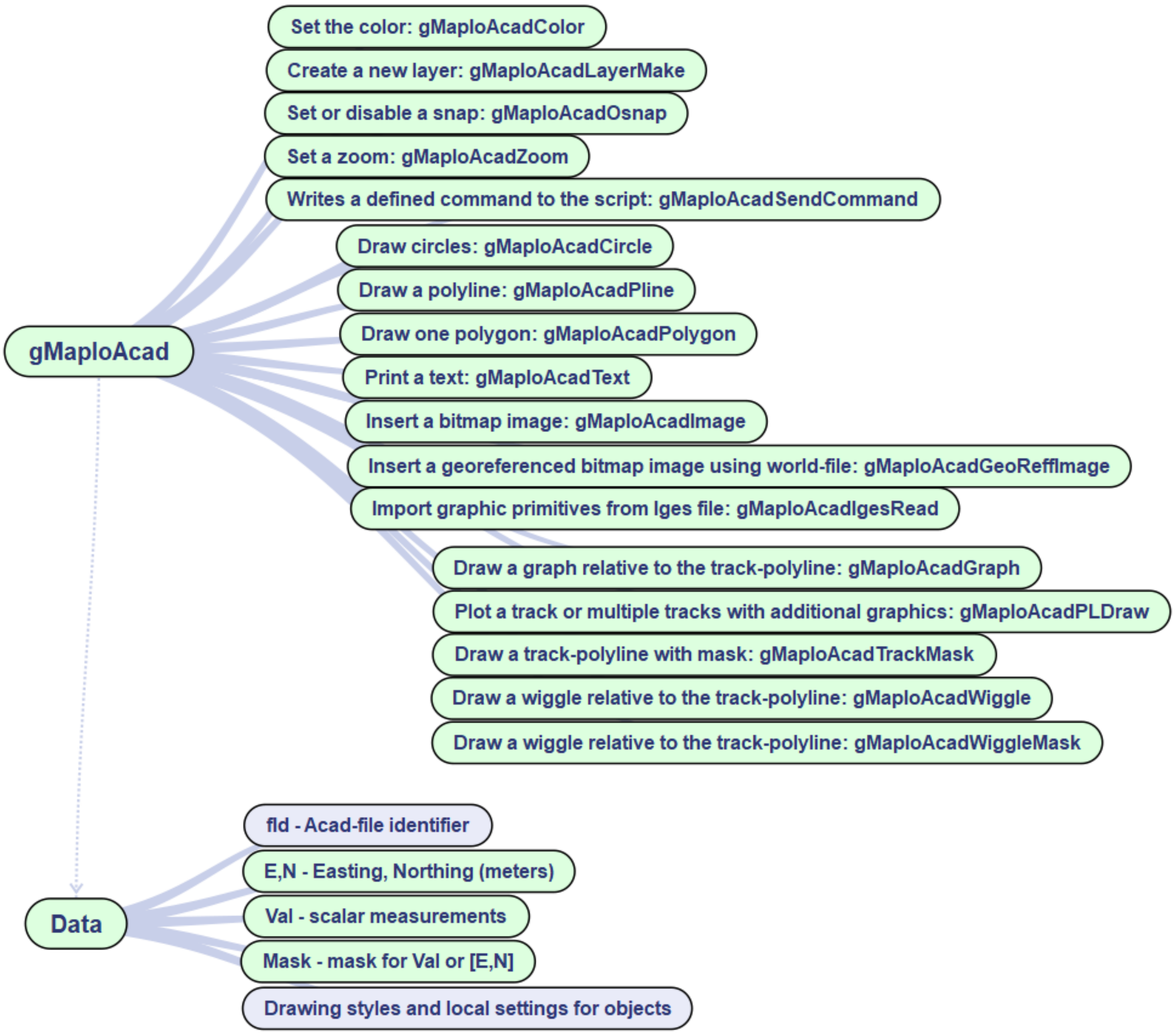

Figure 0.1 The gMapIoAcad library structure

1. Set a Color

function gMapIoAcadColor(fId,ColorRGB)

Writes a command to the AutoCAD script file – to set the Color for graphical objects.

Parameters:

fId – script file identifier;

ColorRGB – the color code specified by [R G B] vector (with values from 0 to 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');
>> gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadColor(fId,[255 0 0]);
>> gMapIoAcadCircle(fId,X,Y,1,[2 2 0]);
>> gMapIoAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);
>> fclose(fId);
```

AutoCAD script line example:

```
-color truecolor 255,0,0
```

2. Set a Zoom in the AutoCAD window

function gMapIoAcadZoom(fId,XYM,dgt)

Writes a command to the AutoCAD script file – to set a Zoom in the AutoCAD window. When using this command for AutoCAD Civil, an error occurs.

Parameters:

fId – script file identifier;

XYM – a vector of three values: x-coordinate (right/E) for the center of the window, y-coordinate (up/N) for the center of the window, scale for the AutoCAD window;

dgt – the number of digits after the decimal point for numeric values XYM (if the value is set as an empty array [ ] , dgt=5).

Function Example:

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

AutoCAD script line example:

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

3. Create a new Layer or switch to an existing layer

function gMapIoAcadLayerMake(fId,layerName)

Writes a command to the AutoCAD script file – to create a new Layer and set it as an active one. If the layer does not exist, it will be created and set as active; if the layer exists, it will be enabled and set as active.

Parameters:

fId – script file identifier;

layerName – layer name;

Function Example:

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

AutoCAD script line example:

```
-layer m "layer1"
```

4. Customize a Snap

function gMapIoAcadOsnap(fId,key)

Writes a command to the AutoCAD script file – to customize an object snap in AutoCad (an object snap “attracts” the drawing tool to the other objects’ node points, such as the end of a line or the center of a circle). To customize a snap, use the "-osnap" command with the Key. To disable a snap, use the 'off' Key.

Parameters:

fId – script file identifier;

key – key word for "-osnap" command. Example: 'off','end', 'mid', 'cen, 'node', 'quad', 'int', 'ins', 'perp', etc.

Function Example:

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

AutoCAD script line example:

```
-osnap off
```

5. Send an arbitrary Command

function gMapIoAcadSendCommand(fId,keyZ)

Writes a defined arbitrary Command (an arbitrary text) to the AutoCAD script file, adding the line termination characters CR+LF (Carriage return + Line feed).

Parameters:

keyZ – a command/text to be written in the AutoCAD script file.

Function Example:

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

AutoCAD script lines example:

```
-osnap off
-color truecolor 10,10,0
```

6. Draw Circles

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

Writes a command sequence to the AutoCAD script file – to draw Circles with centers in X(:), Y(:) coordinates and radii R(:).

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) for circles centers;

Y – y-coordinate row vector (up/N) for circles centers.

R – radius (one value or row vector for each pair of XY);

dgt – the number of digits after the decimal point for numeric values X, Y and R (if the value is set as an empty array [ ] - dgt=[5 5 1]).

Function Example:

```
>> X=[1 2 3 5]’;Y=[4 5 7 10]’;
>> fId=fopen('c:\temp\112.scr','w');
>> gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadColor(fId,[255 0 0]);
>> gMapIoAcadCircle(fId,X,Y,1,[2 2 0]);
>> gMapIoAcadText(fId,X-0.5,Y-0.5,2,0,X,[2 2 1]);
>> fclose(fId);
```

AutoCAD script lines example:

```
circle 1.00,4.00 1
circle 2.00,5.00 1
circle 3.00,7.00 1
circle 5.00,10.00 1
```

7. Print a Text

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

Writes a command sequence to the AutoCAD script file – to print a Text in the X(:),Y(:) coordinates with the size of FontSize(:) and the angle of TextRotAngle(:).

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) for the bottom left corner of the text;

Y – y-coordinate row vector (up/N) for the bottom left corner of the text;

FontSize – text font size (scalar or row vector for each pair of XY);

TextRotAngle – text rotation angle (scalar or row vector for each pair of XY);

AText – the text to be written in X(:), Y(:) coordinates. The text can be defined using the following types of variables:

1) row vector with numeric values;

2) cell row vector with a string in each cell;

3) character matrix (char) where each pair of XY corresponds to one row of the matrix.

dgt – the number of digits after the decimal point for numeric values X, Y and AText (if the value is set as an empty array [ ] - dgt=[5 5 0]).

Function Example:

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

AutoCAD script lines example:

```
text 0.50,3.50 2 0 1.0
text 1.50,4.50 2 0 2.0
text 2.50,6.50 2 0 3.0
text 4.50,9.50 2 0 5.0
```

8. Draw a Polyline

function gMapIoAcadPline(fId,X,Y,dgt)

Writes a command to the AutoCAD script file – to draw a Polyline with the coordinates of the nodes X(:), Y(:).

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of polyline nodes;

Y – y-coordinate row vector (up/N) of polyline nodes;

dgt – the number of digits after the decimal point for numeric values X and Y (if the value is set as an empty array [ ] - dgt=[5 5]).

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');
```

```
>> gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadColor(fId,[255 0 0]);
>> gMapIoAcadPline(fId,X,Y,[2 2]);
>> fclose(fId);
```

AutoCAD script line example:

```
pline 1.00,1.00 2.00,2.00 3.00,4.00 5.00,7.00 6.00,11.00 7.00,16.00 8.00,22.00
```

9. Draw a Polygon

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

Writes a command sequence to the AutoCAD script file – to draw one Polygon as a closed Polyline with Hatch. The polyline will be closed automatically (the first node will be added to the end by the function).

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of polygon nodes (without closing node);

Y – y-coordinate row vector (up/N) of polygon nodes (without closing node);

ColorLine – line [R G B] color vector; if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorHatch – hatch [R G B] color vector; if the value is set as an empty array [ ], the color setting command is not written into the script file;

TransparencyHatch – the hatch transparency as a percentage from 0 to 100 (if the value is set as an empty array [ ], then the Hatch will be opaque);

dgt – the number of digits after the decimal point for numeric values X and Y (if the value is set as an empty array [ ] - dgt=[5 5]).

Functions used: gMapIoAcadColor

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');
>> gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadPolygon(fId,X,Y,[255 0 0],[0 255 0],50,[2 2]);
>> fclose(fId);
```

AutoCAD script lines example:

```
-color truecolor 255,0,0
pline 1.00,1.00 2.00,2.00 3.00,4.00 5.00,7.00 6.00,11.00 7.00,16.00 8.00,22.00 1.00,1.00
-color truecolor 0,255,0
-hatch properties solid transparency 50 select last
```

10. Insert a bitmap Image from the file

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

Writes a command to the AutoCAD script file – to insert a bitmap Image to the XY coordinates with the scale factor defined (the initial horizontal side length of the image is taken as 1 meter). The tiff images are converted to png format.

Parameters:

fId – script file identifier;

fR – path to the image file (string);

XY – coordinates of the bottom left corner of the image [x-coordinate(right/E) y-coordinate(up/N)];

m – scale factor (the initial horizontal side length of the image is taken as 1 meter);

Ang – image rotation angle around the bottom left corner in degrees (a positive angle value corresponds to a counterclockwise rotation)

Function Example:

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

AutoCAD script lines example:

```
-image a c:\temp\image.jpg
10.000000000000000,10.000000000000000 1.000000000000000 0.000000000000000
```

11. Insert a bitmap Image using the world-file

function gMapIoAcadGeoReffImage(fId,fR)

Writes a command to the AutoCAD script file – to insert a georeferenced bitmap Image using world-file data [*2*, *3*, *4*]. The tiff images are converted to png format.

The Geo-referenced Image includes two files: (1) a bitmap file such as gif, jpg, png and (2) a text file with a coordinate reference for a bitmap file called “world-file”. The file extension with a coordinate reference is formed from the extension of the graphic file, with the replacement of two characters with fw; for example: tfw, jfw, pfw. The description of the lines of the file with the coordinate reference is given below:

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.

Parameters:

fId – script file identifier;

fR – path to the image file (string), includes image extension (the world-file must be in the same folder as the image file).

Function Example:

```
>> fId=fopen('c:\temp\112.scr','w');
>> gMapIoAcadGeoReffImage(fId,'c:\ge0mlib\FAA_UTM18N_NAD83.tif');
>> fclose(fId);
>> fId=fopen('c:\temp\112.scr','w');
>> for n=1:23, gMapIoAcadGeoReffImage(fId,['c:\temp\1\' num2str(n,'%02d') '.tif']);
>> end;fclose(fId);
```

AutoCAD script lines example:

```
-layer m "FAA_UTM18N_NAD83"
-image a c:\ge0mlib\FAA_UTM18N_NAD83.png
424178.114726012805477,4285432.609310633502901 30081.540914537796198 0.000000000000000
```

12. Draw a Track-polyline using a mask

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

Writes a command sequence to the AutoCAD script file – to draw a signed Track-polyline with the X(:),Y(:) coordinates; the data marked with a 0/false mask is not drawn (a gap forms in the polyline and it is divided into several segments).

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of polyline points;

Y – y-coordinate row vector (up/N) of polyline points;

mask – a mask (row vector) for drawing polyline points; the mask value 0/false marks the points that will not be drawn, in this case the polyline will be divided into several segments;

FontSize – font size of the text with the track-polyline name;

TextRotAngle – rotation angle of the text with the track-polyline name;

AText – text string with the name of the track-polyline; the name will be printed at the starting point for each segment;

dgt – the number of digits after the decimal point for numeric values X, Y and AText (if the value is set as an empty array [ ] - dgt=[5 5 0]).

Functions used: gMapIoAcadPline, gMapIoAcadText.

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');
>> gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadColor(fId,[255 0 0]);
>> gMapIoAcadTrackMask(fId,x,y,m,1,90,'E95',[2 2 1]);
>> fclose(fId);
```

AutoCAD script lines 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
```

13. Draw a Graph relative the track-polyline

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

Writes a command sequence to the AutoCAD script file – to draw a Graph relative to the track-polyline.

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of polyline points;

Y – y-coordinate row vector (up/N) of polyline points;

T – row vector of the graph polyline values that will be drawn relative to the track;

Ang – graph-line rotation angle (the value of the rotation angle is calculated in the coordinate system used to determine the azimuth; Ang = 0 - the axis is pointing up/N; when turning right/clockwise, the angle sign is positive).

ColorGraph – graph [R G B] color vector; if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorTrack – track-polyline [R G B] color vector; if the value is set as an empty array [ ], the color setting command is not written into the script file;

dgt – the number of digits after the decimal point for numeric values X and Y (if the value is set as an empty array [ ] - dgt=[5 5]).

Functions used: gMapIoAcadColor.

Function Example (*Figure 13.1*):

```
>> 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');gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadGraph(fId,X,Y,T,0,[255 0 0],[0 255 0],[2 2]);fclose(fId); %red
>> fId=fopen('c:\temp\2.scr','w');gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadGraph(fId,X,Y,T,45,[255 255 0],[0 255 0],[2 2]);fclose(fId); %yellow
```

AutoCAD script lines example:

```
zoom c 0.0000,0.0000 0.0001
-color truecolor 255,0,0
pline 1.00,1.00 3.00,2.00 8.00,4.00 5.00,7.00 3.00,11.00 2.00,16.00 3.00,22.00
6.00,30.00

-color truecolor 0,255,0
pline 1.00,1.00 2.00,2.00 3.00,4.00 4.00,7.00 5.00,11.00 6.00,16.00 7.00,22.00
8.00,30.00
```

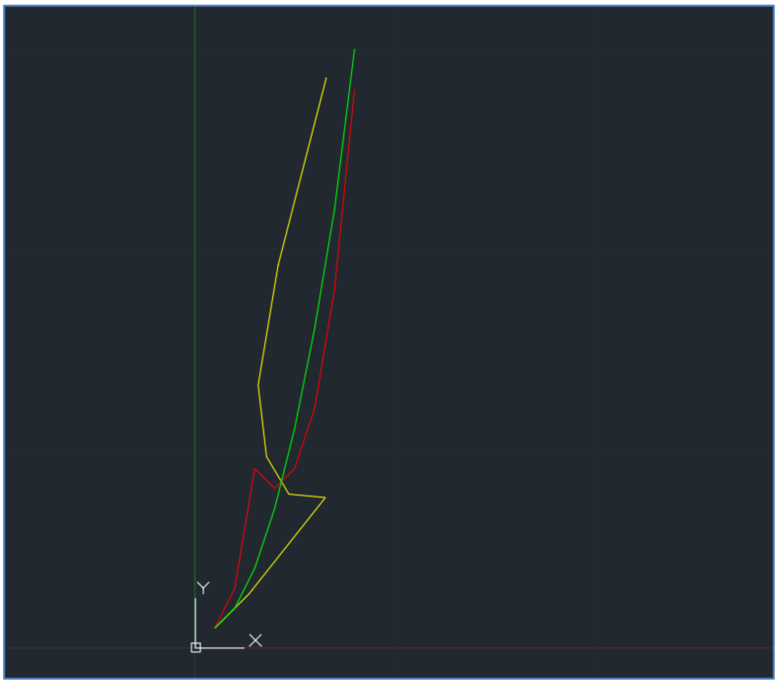

Figure 13.1 gMapIoAcadGraph example result – two graphs (red and yellow) drawn along the track (green)

14. Plot a Track with additional graphics

function gMapIoAcadPLDraw(fName, PL, PLNameAttr, CircleAttr, KPNameAttr, DigitNum, PointsStep,flLayers)

Writes a command sequence to the AutoCAD script file – to plot a Track or multiple tracks (if the PL(n) variable contains multiple values) with additional graphics. When plotting each track, the following steps are performed:

-- a new layer with the name of the track is created in AutoCAD; it is a layer in which the track will be plotted;

-- a polyline is drawn based on the coordinates of the track with a given step (for example, you can draw a polyline only for every tenth value, but not for all coordinates of the track);

-- a circle of the specified size is drawn for the first track point and the track name is printed;

-- circles are drawn for polyline nodes (track points after decimation) with a given step (if, for example, it is necessary to mark every hundredth node of the polyline);

-- signatures are printed with a given step for polyline nodes (track points after decimation) (a more detailed description of the signature text is given down below).

Parameters:

fName – the name or identifier of the script file for plotting a track;

If the file name ends with “\”, the script will be written to a file with the track name, which is contained in the PL(1) structure field. PLName;

PL(n) – a variable of structure type containing the geometry of the track. The structure includes the following fields:

1) PLName – track name;

2) GpsE – x-coordinate row vector (right/E) of track points;

3) GpsN – y-coordinate row vector (up/N) of track points;

4) GpsKP – field which contains signatures for each track point.

Generally, it is assumed that the GpsKP field contains data on the distance from first track point to each of its points (KP stands for kilometer point). However, when used with this function, the GpsKP field contains the data that will be used as signatures to the polyline nodes. The data format in the GpsKP field must match the text format for the gMapIoAcadText function (row vector with numeric values or cell row vector with a text in each cell).

PLNameAttr – vector specifying the text format for the polyline name:

1) Size – text font size;

2) Angle – text rotation angle;

3) dE – distance (in meters) between the bottom left corner of the text and the first point of the track along the x axis;

4) dN – distance (in meters) between the bottom left corner of the text and the first point of the track along the y axis;

5) CircleRadius – radius of the circle that is used to outline the first point of the track.

If only the Size and Angle parameters are set for the PLNameAttr vector, then dE=0, dN=0, and the first point of the track is not outlined by a circle.

CircleAttr – vector with the step and radius of circles, which are used to outline the specific polyline nodes:

1) Radius – circles radius;

2) DrawModulio – step of drawing circles for polyline nodes.

If the CircleAttr vector includes more than two values, the polyline for the track is not drawn.

KPNameAttr – vector specifying the text format for polyline points signatures (data for signatures are in the GpsKP field):

1) Size – text font size;

2) Angle – text rotation angle;

3) DrawModulio – step of signing polyline nodes;

4) KPDigitNum – the number of digits after the decimal point for numeric values, which the track points are signed with (if PL(n).GpsKP is defined by a vector of numeric values).

Signatures can be either a numeric vector or a cell vector.

DigitNum – the number of digits after the decimal point for all coordinates and radii of circles.

PointsStep – step of drawing polyline nodes (relative to the track points).

flLayers – flag for creating a new layer for each track from the PL(n) variable; the layer name is the same as the track name.

Functions used: gMapIoAcadZoom, gMapIoAcadLayerMake, gMapIoAcadText, gMapIoAcadCircle, gMapIoAcadPline.

Function Example (*Figure 14.1*):

```
>> PL = struct('PLName','PL','GpsE',10:23,'GpsN',5:18,'GpsKP',1:14);
>> gMapIoAcadPLDraw('c:\temp\117.scr',PL,[1.5 0 0 -2 1],[0.4 2],[1 0 4 2],2,1,1);
```

AutoCAD script lines example:

```
-layer m "PL"
text 10.00,3.00 1.5 0 PL
circle 10.00,5.00 1.00
pline 10.00,5.00 11.00,6.00 12.00,7.00 13.00,8.00 14.00,9.00 15.00,10.00 16.00,11.00
17.00,12.00 18.00,13.00 19.00,14.00 20.00,15.00 21.00,16.00 22.00,17.00 23.00,18.00
circle 11.00,6.00 0.40
circle 13.00,8.00 0.40
circle 15.00,10.00 0.40
circle 17.00,12.00 0.40
circle 19.00,14.00 0.40
circle 21.00,16.00 0.40
circle 23.00,18.00 0.40
text 13.00,8.00 1 0 4.00
text 17.00,12.00 1 0 8.00
text 21.00,16.00 1 0 12.00
```

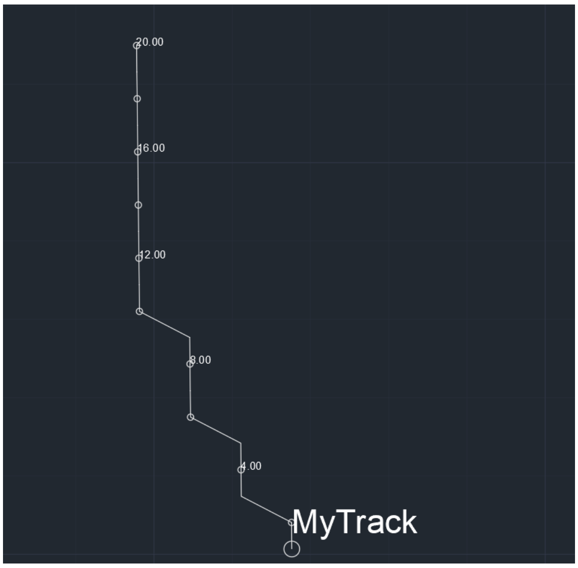

Figure 14.1 Examples of a track plotted by gMapIoAcadPLDraw (an AutoCAD script execution result)

15. Draw a Wiggle relative to the track-polyline

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

Writes a command to the AutoCAD script file – to draw a Wiggle relative to the track-polyline.

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of track-polyline points;

Y – y-coordinate row vector (up/N) of track-polyline points;

T – row vector of wiggle values;

Ang – wiggle-line rotation angle (the value of the rotation angle is calculated in the coordinate system used to determine the azimuth; Ang = 0 - the axis is pointing up/N; when turning right/clockwise, the angle sign is positive).

ColorLine – wiggle-line color (scalar value); if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorHatch1 – hatch [R G B] color for positive wiggle values (above the track-polyline (zero line)); if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorHatch2 – hatch [R G B] color for negative wiggle values (below the track-polyline (zero line)); if the value is set as an empty array [ ], the color setting command is not written into the script file;

TransparencyHatch – the hatch transparency as a percentage from 0 to 100 (if the value is set as an empty array [ ], then the Hatch will be opaque);

FontSize – font size of the text with the track-polyline name;

TextRotAngle – rotation angle of the text with the track-polyline name;

AText – text string with the name of the track-polyline;

dgt – the number of digits after the decimal point for numeric values X, Y and AText (if the value is set as an empty array [ ] – dgt = [5 5 0]).

Functions used: gMapIoAcadColor.

A wiggle drawn for a few points is shown in ***Figure 15.1***. A wiggle drawn for a large set of points (the results of measuring the magnetic field along the tracks) is shown in ***Figure 16.2***.

Function example (*Figure 15.1*):

```
>> 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\112.scr','w');gMapIoAcadZoom(fId,[0 0 0.0001],4);
>> gMapIoAcadWiggle(fId,X,Y,T,90,[0 0 255],[255 0 0],[0 255 0],50,1,90,'E95',[2 2 1]);fclose(fId);
```

AutoCAD script lines example:

```
zoom c 0.0000,0.0000 0.0001
-color truecolor 0,0,255
pline 1.00,1.00 2.00,2.00 3.00,4.00 4.00,7.00 5.00,11.00 6.00,16.00 7.00,22.00 8.00,30.00 8.00,30.00 7.00,22.00 6.00,16.00 5.00,11.00 3.00,7.00 -2.00,4.00 1.00,2.00 1.00,1.00 1.00,1.00
-color truecolor 255,0,0
```

```
-hatch properties solid transparency 50 select last
-color truecolor 0,0,255
pline  1.00,1.00  2.00,2.00  3.00,4.00  4.00,7.00  5.00,11.00  6.00,16.00  7.00,22.00
8.00,30.00 10.00,30.00 11.00,22.00 10.00,16.00 7.00,11.00 4.00,7.00 3.00,4.00 2.00,2.00
1.00,1.00 1.00,1.00
-color truecolor 0,255,0
-hatch properties solid transparency 50 select last
text 1.00,1.00 1 90 E95
```

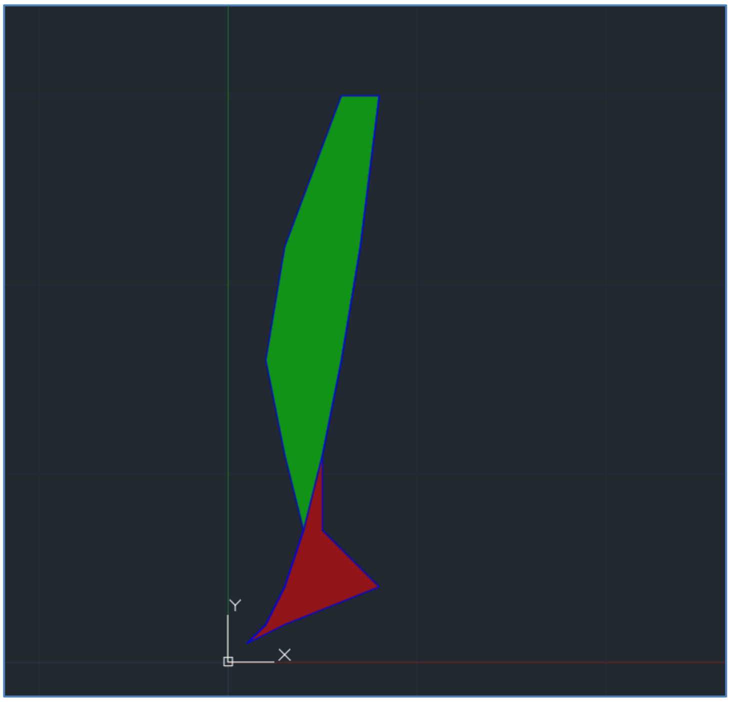

Figure 15.1 gMapIoAcadWiggle example result (an AutoCAD script execution result)

16. Draw a Wiggle relative to the track-polyline using a mask

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

Writes a command to the AutoCAD script file – to draw a Wiggle (relative to the track-polyline) by removing sections of the wiggle using a mask.

Parameters:

fId – script file identifier;

X – x-coordinate row vector (right/E) of track points;

Y – y-coordinate row vector (up/N) of track points;

T – row vector of wiggle values;

mask – a mask for drawing wiggle points; the mask value 0/false marks the points that will not be drawn; the wiggle will be divided into several segments;

Ang – wiggle-line rotation angle (the value of the rotation angle is calculated in the coordinate system used to determine the azimuth; Ang = 0 - the axis is pointing up/N; when turning right/clockwise, the angle sign is positive).

ColorLine – wiggle-line color (scalar value); if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorHatch1 – hatch [R G B] color for positive wiggle values (above the track-polyline (zero line)); if the value is set as an empty array [ ], the color setting command is not written into the script file;

ColorHatch2 – hatch [R G B] color for negative wiggle values (below the track-polyline (zero line)); if the value is set as an empty array [ ], the color setting command is not written into the script file;

TransparencyHatch – the hatch transparency as a percentage from 0 to 100 (if the value is set as an empty array [ ], then the Hatch will be opaque);

FontSize – font size of the text with the track-polyline name;

TextRotAngle – rotation angle of the text with the track-polyline name;

AText – text string with the name of the track-polyline; the name will be printed at the starting point for each segment;

dgt – the number of digits after the decimal point for numeric values X, Y and AText (if the value is set as an empty array [ ] - dgt=[5 5 0]).

Functions used: gMapIoAcadWiggle.

Function Example (*Figure 16.1*):

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

AutoCAD script lines example:

```
zoom c 0.0000,0.0000 0.0001
-color truecolor 0,0,255
pline 1.00,1.00 2.00,2.00 3.00,4.00 4.00,7.00 3.00,7.00 -2.00,4.00 1.00,2.00 1.00,1.00
1.00,1.00
-color truecolor 255,0,0
-hatch properties solid transparency 50 select last
-color truecolor 0,0,255
pline 1.00,1.00 2.00,2.00 3.00,4.00 4.00,7.00 4.00,7.00 3.00,4.00 2.00,2.00 1.00,1.00
1.00,1.00
-color truecolor 0,255,0
-hatch properties solid transparency 50 select last
text 1.00,1.00 1 90 E95
-color truecolor 0,0,255
pline 7.00,22.00 8.00,30.00 8.00,30.00 7.00,22.00 7.00,22.00
-color truecolor 255,0,0
-hatch properties solid transparency 50 select last
-color truecolor 0,0,255
```

```
pline 7.00,22.00 8.00,30.00 10.00,30.00 11.00,22.00 7.00,22.00
-color truecolor 0,255,0
-hatch properties solid transparency 50 select last
text 7.00,22.00 1 90 E95
```

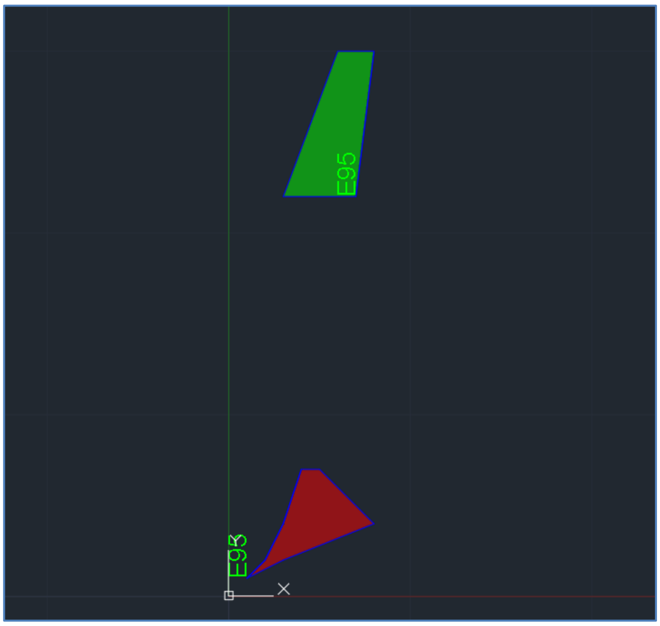

Figure 16.1 gMapIoAcadWiggleMask example result (an AutoCAD script execution result)

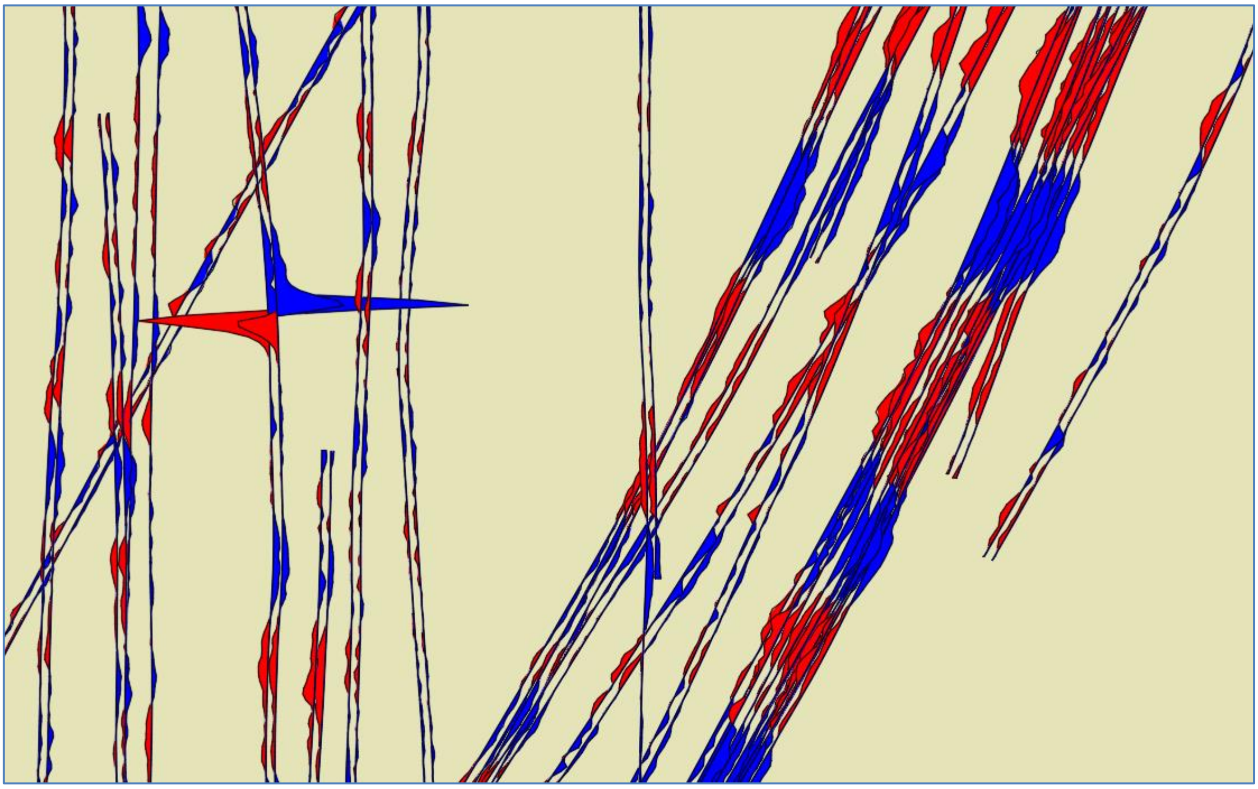

Figure 16.2 gMapIoAcadWiggleMask function result (an AutoCAD script execution result) – residual magnetic field drawn along tracks of vessel

17. Import graphics from an Iges file

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

The function is used to import graphic primitives, exported from AutoCAD to an Iges file [5], into MatLab. Iges is a text file with a set of “records”. Each “record” contains some information about the graphic object. The primitives that the gMapIoAcadIgesRead function can be used for are shown in ***Figure 17.1***. The gMapIoAcadIgesRead function is not a full-fledged Iges format converter, it can only read information about coordinates from 4 types of records: circle center coordinates and radius (124, 100), Lines (110) and Polylines (126) nodes coordinates from Iges-files. The function puts the read information into the output variables CircXY, LL, PL.

The Iges file text example is shown in ***Figure 17.2***. The record 124 contains information about the circle center coordinates, the record 100 contains information about the radius of the circle, the record 110 contains information about the coordinates of the beginning and end of the line, the record 126 contains information about the polyline nodes coordinates.

Parameters:

fName – name of the Iges file;

CircXY – output of the xy-coordinates and R (radius) in three rows (Iges records 124 and 100);

LL – output variable of structure type for lines (it is formed based on the Iges record 110); the xy-coordinates of the start and end of the line are contained in the GpsE and GpsN fields.

PL – output variable of structure type for polylines (it is formed based on the Iges record 126); the xy-coordinates of the polyline nodes are contained in the GpsE and GpsN fields.

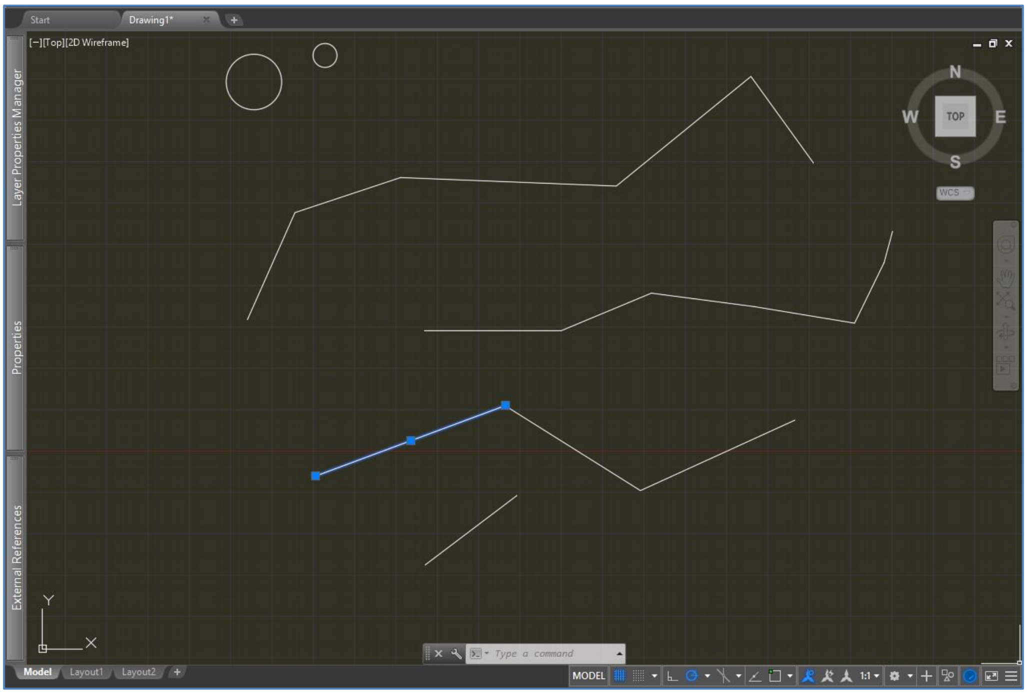

Figure 17.1 Primitives that can be imported into MatLab using the gMapIoAcadIgesRead function (circles, lines, polylines)

```
                                                                        S      1
,,,8HDrawing1,22HAutoCAD 2017 - English,62HAutodesk Translation FrameworG      1
k DWG producer (atf_dwg_producer),32,38,7,99,15,,1.,2,2HMM,1,0.08,15H202G      2
11206.091129,0.01,10000.,,7Hunknown,11,0,15H20211204.132652;            G      3
     126       1                       0                        00000000D      1
     126       0     -39       6       0                               0D      2
     126       7                       0                        00000000D      3
     126       0     -39       6       0                               0D      4
     124      13                                                00000000D      5
     124                       2       0                               0D      6
     124      15                                                00000000D      7
     124                       2       0                               0D      8
     100      17                       0               5        00000000D      9
     100       0     -39       1       0                               0D     10
     100      18                       0               7        00000000D     11
     100       0     -39       1       0                               0D     12
     110      19                       0                        00000000D     13
     110       0     -39       2       0                               0D     14
     110      21                       0                        00000000D     15
     110       0     -39       2       0                               0D     16
     110      23                       0                        00000000D     17
     110       0     -39       2       0                               0D     18
     110      25                       0                        00000000D     19
     110       0     -39       2       0                               0D     20
     406      27                                                00000000D     21
     406                       1      15                               0D     22
     406      28                                                00000000D     23
     406                       1      15                               0D     24
     406      29                                                00000000D     25
     406                       1      15                               0D     26
     406      30                                                00000000D     27
     406                       1      15                               0D     28
     406      31                                                00000000D     29
     406                       1      15                               0D     30
     406      32                                                00000000D     31
     406                       1      15                               0D     32
     406      33                                                00000000D     33
     406                       1      15                               0D     34
     406      34                                                00000000D     35
     406                       1      15                               0D     36
     406      35                       0                        00000000D     37
     406                       1       3                               0D     38
     314      36                                                00000200D     39
     314                       1       0                               0D     40
126,6,1,0,0,0,0,0.,0.,1.,2.,3.,4.,5.,6.,6.,1.,1.,1.,1.,1.,1.,1.,       1P      1
2670.34891496647,728.080381004324,0.,3499.9908220839,                  1P      2
728.080381004324,0.,4045.01104828645,955.024229583664,0.,              1P      3
4668.75641575856,873.324442797994,0.,5274.33443858439,                 1P      4
773.469150720192,0.,5452.97996126028,1139.60522815767,0.,              1P      5
5504.45408556042,1330.23805966721,0.,0.,6.,0.,0.,0.,0,1,33;            1P      6
126,5,1,0,0,0,0,0.,0.,1.,2.,3.,4.,5.,5.,1.,1.,1.,1.,1.,1.,             3P      7
1601.50370269813,794.650575722859,0.,1889.15326495522,                 3P      8
1442.19702410161,0.,2528.03808158316,1654.01128709936,0.,              3P      9
3833.05873180846,1602.57067796242,0.,4647.56117562181,                 3P     10
2265.24672489385,0.,5026.04744696208,1741.76290682056,0.,0.,5.,        3P     11
0.,0.,0.,0,1,35;                                                       3P     12
124,-1.,0.,0.,2070.82666897332,0.,-1.,0.,2392.33527490984,0.,0.,       5P     13
1.,0.;                                                                 5P     14
124,-1.,0.,0.,1640.86627333291,0.,-1.,0.,2231.96162104904,0.,0.,       7P     15
1.,0.;                                                                 7P     16
100,0.,0.,0.,-72.212481818995,0.,-72.212481818995,0.,0,1,29;           9P     17
100,0.,0.,0.,-167.622275214138,0.,-167.622275214138,0.,0,1,31;        11P     18
110,2675.84819864304,-689.129438514559,0.,3232.93992481239,           13P     19
-267.871747241905,0.,0,1,21;                                          13P     20
110,3978.95841053741,-238.819489764065,0.,4913.9036558181,            15P     21
187.280247664333,0.,0,1,23;                                           15P     22
110,3160.27579522837,274.437007126769,0.,3978.95841053741,            17P     23
-238.819489764065,0.,0,1,25;                                          17P     24
110,2012.18240914793,-151.662723816088,0.,3160.27579522837,           19P     25
274.437007126769,0.,0,1,27;                                           19P     26
406,1,2H58;                                                           21P     27
406,1,2H59;                                                           23P     28
406,1,2H5A;                                                           25P     29
406,1,2H5B;                                                           27P     30
406,1,2H5C;                                                           29P     31
406,1,2H5D;                                                           31P     32
406,1,2H5E;                                                           33P     33
406,1,2H5F;                                                           35P     34
406,2,0,1H0;                                                          37P     35
314,100.,100.,100.,;                                                  39P     36
S      1G      3D     40P     36                                        T      1
```

Figure 17.2 Iges (*.iges) file content

Function example:

```
>> [CircXY,LL,PL]=gMapIoAcadIgesRead('c:\temp\Drawing1.iges');
```

The export steps for Iges file are described below.

Step 1: Click to the AutoCAD icon in the top left of the screen, then select the menu item shown in *Figure 17.3*.

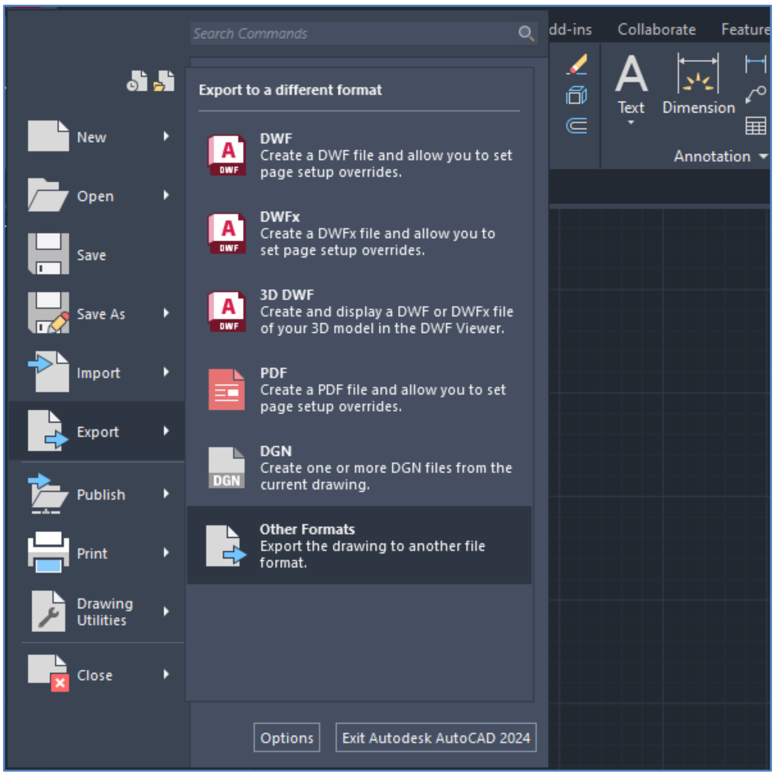

Figure 17.3 AutoCAD Export menu

Step 2: Choose the type of the file as Iges (*.iges) and click to the "Save" button (*Figure 17.4*).

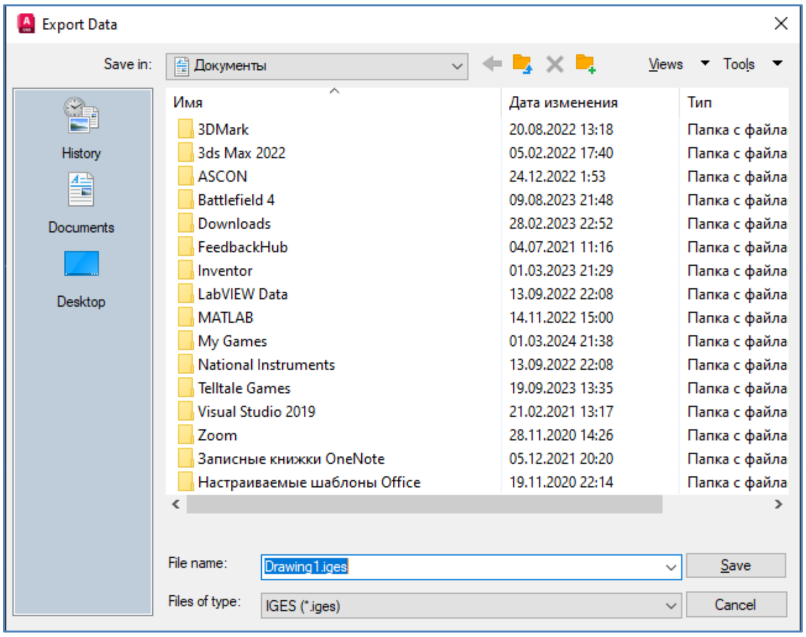

Figure 17.4 AutoCAD Save menu

Step 3: Select the objects that you want to export using the left mouse button and press the "Enter" button (*Figure 17.5*).

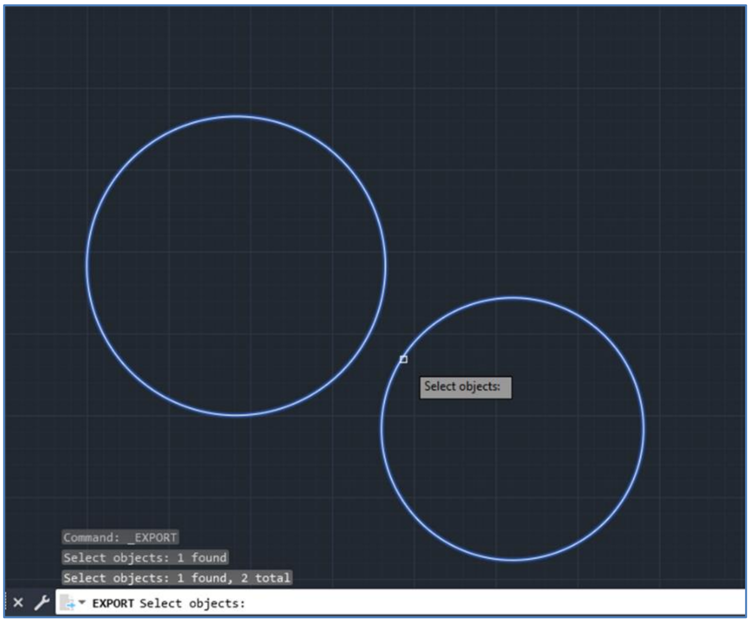

Figure 17.5 Selection of the objects for export in AutoCAD window

Citation

1. Autodesk AutoCAD 2017 help.
2. https://en.wikipedia.org/wiki/World_file
3. https://webhelp.esri.com/arcims/9.3/General/topics/author_world_files.htm
4. https://www.loc.gov/preservation/digital/formats/fdd/fdd000287.shtml
5. https://ge0mlib.com/papers/File_Formats/Iges_rev5_03.pdf