

**GGENERAL**  
**GEOMLIB (ALPHA)**

**MARINE ENGINEERING GEOPHYSICAL  
DATA PROCESSING TOOLBOX**

Ivan V. Dmitriev  
12.07.2018

***Contents***

1	Ge0MLib Summary.....	4
2	Geophysical field’s “components” .....	5
3	Geophysical data acquisition and processing stages .....	6
4	Data Content types .....	11
5	Pre-defined structures field names .....	14
5.1	General time reference fields .....	14
5.2	General position/coordinates reference fields .....	14
5.3	PolyLine structure .....	14
5.4	Navigation’s Sensors fields structure .....	14
5.5	Pre-Processed Set fields structure .....	14
6	Functions Glossary .....	15
6.1	gFields – Row content's functions.....	15
6.2	gData – Matrix content's functions.....	15
6.3	gNav – Navigation and coordinates manipulations.....	15
6.4	gMap – Geometrics tasks decision, trackplots/pipelines/lineplanning, picking, graphics .....	16
6.5	gLog – Serial data logging/read/write .....	17
6.6	gJsf – Jsf files read/write/manipulations .....	18
6.7	gXtf – Xtf files read/write/manipulations .....	19
6.8	gSgy – Sgy files read/write/manipulations.....	20
6.9	gMagy – Magnitometers data read/write/manipulations.....	21
6.10	gUhr – Ultra High Resolution Seismic data read/write/manipulations.....	22
6.11	gGrav – Gravimeters data read/write/manipulations.....	22
6.12	gP190 – P190 files read/write/manipulations.....	22
6.13	gWfr – Images with coordinate world-files and XYZ-grid-files manipulations .....	22
6.14	gAcad – AutoCad scripts creation.....	23
6.15	gZ – Ancillary functions .....	23
	Citation.....	24

## ***Figures list***

<b><i>Figure 1</i></b> Geophysical field's "components".....	5
<b><i>Figure 2</i></b> Data acquisition and processing chart.....	6
<b><i>Figure 3</i></b> Data acquisition and processing chart (extended).....	10
<b><i>Figure 4</i></b> Measurements' data content types .....	12
<b><i>Figure 5</i></b> Dataset content types .....	13

## ***Formatting and colors***

***Links*** – the internal (figures, tables, paragraphs) links or external links (web-sites);

***Terms*** – the definitions or concept used first time in the text;

**Text** – the text concerning particular GeOMLib's realizations in the context of general statements;

**Subheading** – the logical subtitles;

**Marks** – the text need to correction in future.

## 1 Ge0MLib Summary

Ge0MLib is library with predefined structures and basic algorithms for Marine Engineering Geophysical Survey data logging and processing; there are: Subbotom Profiling, Side Scan Sonar, Magnetometry and Navigation.

The “specific” Ge0MLib features are:

- the usable MatLab structures for Geophysical Survey’s tasks decision;
- vessel with equipment is regarded as a single object described math models (for example, some corrections in navigation sensors data can easy applied for all towing/mounting equipment’s position);
- easy access to each byte of data;
- access to all source codes and algorithms; possibility for its modification;
- special tools for serial sensors’ data logging and synchronization.

Data processing procedures are developed in MatLab2015b (<https://mathworks.com/>). Data loggers are developed in Free Pascal (<http://www.freepascal.org/>).

Ge0MLib sources can be downloaded from <http://ge0mlib.ru/> (compiled loggers are included). The Ge0MLib function sets are shown in the *Functions Glossary*; the set names are linked to sets' description. I try to create independent sets without cross-references, but sometimes (usually in Examples) a functions from another sets can be used. The functions from different sets can be copied to one folder.

Any bugs, comments or questions can be sent to [mail@ge0mlib.ru](mailto:mail@ge0mlib.ru)

## 2 Geophysical field's "components"

Surveyed Geophysical Field is characterized the follow *components* (Figure 1):

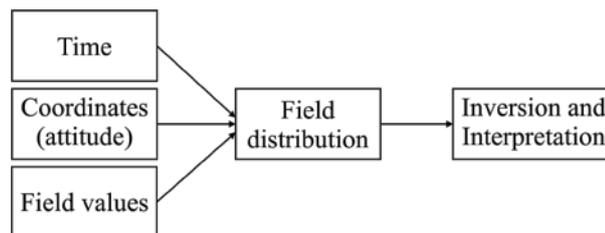
- Geophysical Field measurement's *time* (synchronization);
- Measured Field *coordinates* (calculated using measurement's equipment attitude);
- *Feld* measurements values.

The components are characterized the *Field distribution* in time and space. Each component is calculated/formed using a number of sensors and algorithms. There are:

- 1) The data from physical field sensors – the physical field measurements value. This is geophysical equipment data.
- 2) The data from "time sensors". This is navigation equipment data (GPS-fixes and/or computer's clock).
- 3) The data from "attitude/coordinates sensors". This is navigation equipment data.

The navigation equipment's measurements are very important part of geophysical data – the navigation equipment presents physical field distribution in time and space. The "field distribution" is the basis of field's inversion and interpretation.

The data acquisition and processing stages are discussed in *Geophysical data acquisition and processing stages* more widely.



*Figure 1* Geophysical field's "components"

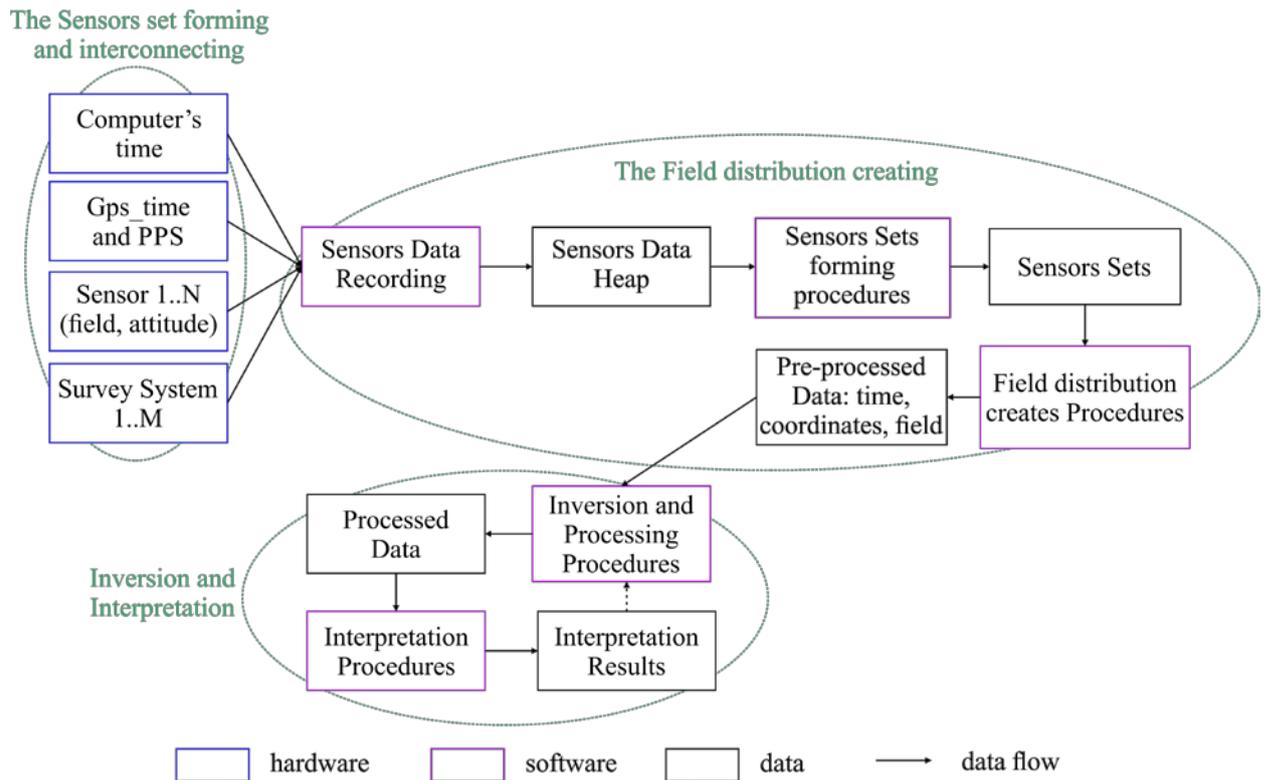
The first/main task, solved by Ge0MLib is the correct "field distribution creating", using geophysical and navigational equipment data. Why that? The geophysical field processing and inversion are the good-formalized tacks; there can be defined input/output parameters and a number math of decision. Usually, the engineer-man cannot easy and quick apply changes to math (boundary conditions for example) because this action needs specific knowledge/education/specialization; but, usually there are no needs for same changes – we paid for using good-formalized and slow-changeable software developed by specialists.

The "field distribution creating" is not so good formalized task in practice. There are different equipment and on-line software producing companies, different protocols features (latency for example), and (the main!) different survey methodic (it is means equipment location, towing methods, control, etc.). The "five-buttons-software" or vectors-calculators are poor tools for this case; the better tool is math-programming-language with special libraries. The MatLab was chosen as a language with a code high-density (you need not spend your time/attention to proگرامing cycles or operations like matrixes multiple) and easy for learning.

### 3 Geophysical data acquisition and processing stages

The data acquisition and processing chart (was drawn the context of philosophy, which realized in Ge0MLib) are shown in *Figure 2*. The navigation and geophysical sensors data are used. There are follow stages:

**0) The Sensors/Hardware set forming and interconnecting** (there are time sensors, navigation/attitude sensors/physical fields' sensors and survey systems) for main task decision (to get a representative Interpretation Results).



*Figure 2* Data acquisition and processing chart

#### 1) Data acquisition/recording

The data flows from sensors come to data recording software. The main source of data flows linkage is the time-stamps; the process named as synchronization – each sensor data link to time-of-measurements presented by “time sensor”. This stage result is *Sensors Data Heap*, which contained linked records for all sensors we have in digital format; there is the on-line survey result.

There are two synchronization “sources” used by Ge0MLib (see gLog tool):

- the “local” source is the computer clock data flow with 0.001 second discrecity;
- the “global” fix-source (for a number computers synchronization), usually it is GPS PPS (pulse per second) and/or GPS-message with time and coordinates.

In the practice – the computer clock time is used for interpolation inside GPS’s PPS; the GPS-time applies for: 1) long-time-drift correction to computer’s clock data; 2) computer’s time-stamp errors search (really, the Windows is not real-time system; the OS can be “busy” and be delayed).

The gLog tool is linked any messages from serial port to computer's clock and save linked message to HDD; the PPS linked to computer's clock too, as a "PPS-sensor".

## 2) The Sensors Sets forming

A number of Sensors messages/data, selected for "field distribution creating" are formed to *Sensors Set* (the *adjusted sensors data* are re-link/binding to *common set's time*). There are follow procedures for each sensor:

- a) Data search (into Sensors Data Heap) using defined time and import.
- b) The "bad" measurements delete. The "bad" measurement natures is not defined by measured value physics, it is define by extraneous (third-party) forces. There are two "bad" measurements types:
  - spikes and other "features", which cannot defined by measured value physics;
  - values with bad quality flag, included to data flow from sensor (for example GpsFixQuality = 0).
- c) Measurements smooth/filtering/de-noise procedures. The procedures applied for normal-distribution noise decrease, sensors' output resolution influence decrease, sensors characteristics applying (for example: noise's frequency decrease using amplitude-frequency characteristic for seismic source and seismic receiver). Procedures are referring to "specific" sensor type and characteristics.
- d) Adjusted sensors data re-link/binding to common set's time (usually, GPS-time). Usually the Sensors Set is prepared for *geophysical survey line*. The common set's time start and end is defined by survey line record time.

The Ge0MLib functions are not used database for Data Heap realization. The Data Heap realized on the file-system basis, using special folders-names and files-names. The folder contains Survey Systems data files (SSS, SBP, UHR, etc.) were created special on-line survey software and data files from serial ports, were created gLog tools. The gLog's saved data used for synchronization and navigation calculations.

The geophysical survey lines time takes from *Survey Logs* was filled on-line geophysicists in survey time. Using survey lines time the data search (in the file system by file names) is realized.

## 3) The Field-distribution creates Procedures

The Field-distribution creates Procedures are "combined" sensors data to calculate "more universal" *pre-processed data* (which described Field-distribution) in accordance with some mathematical model. For example: vessel's GPS/MRU/Draft-sensors/Nod's-position data, tow-cable length, tow fish's altitude/depth sensors data are combines for calculation of "towed fish position".

The mathematical model is not deal with "specific" sensors characteristics (it was applied in Sensors Sets forming stage); only with abstract "sensor's measurements results" with measurement's error. The mathematical model is not deal with with measured physical field sources/equations (it will be applied in Inversion and Processing stage).

There are follow steps for Pre-processed Data set creation:

- a) Data Set sensor's interpolation to common time-series (usually, the time "points" are defined by geophysical sensors measurement time);
- b) Position (coordinates, attitude) calculations on the basis of navigation sensors;
- c) Geophysical field calculations (adjustment) in accordance with own mathematical models (for example: Earth's magnetic variations or vessel's deviations applying);
- d) Disambiguation for Sets data (for example: the bathy data, created by vessel's MBES and bathy data, created by towed fish's altimeter are different; so, we need correct towed fish's position calculation model).
- e) Data Set transforming to *Pre-processed Data Set*, which describe geophysical field distribution (the original sensor's fields/data are removed). The Pre-processed Data Set includes: adjusted physical field values, time of measurements and coordinates/attitude of measurements.

The follow types of position calculations (b) are realized in Ge0MLib (see gNav tool):

- Nod's position calculation (GPS, MRU/IMU, Gyrocompass, Draft-sensor, etc.). It will be used for nods-position calculation (for mounting equipment and tow-points).
- Towing equipment position calculation using Layback. It deal with towing-nod position, cable length (cable counter's measurements), towed body's attitude sensors (depth sensor, altimeter, MRU/IMU, Gyro, etc.).
- Beacon's position calculation using USBL (head positioning, USBL data, etc).
- The Optimizing models, which used combining and optimization for Layback and USBL data.

#### **4) The Inversion and Processing Procedures**

The *Processing Procedures* deal with physical field distribution in time and space. The procedures' mathematical model deals with physical field nature (equations). The procedures' tasks are inverse problem decision or the same. The *Processed Data* are the physical field's sources distribution/characteristics or another inverse geophysical task' results.

The example of Processing Procedures is local object's (UXO) magnetic field modeling and identification in "measured" magnetic field (more carefully – in the adjusted field with adjusted coordinates, which contained in Pre-processed Data Set). The Processed Data are position (includes depth), magnetic mass and probability of true object's identification.

#### **5) The Interpretation Procedures**

The *Interpretation Procedures* is the Processed Data "interpretation" in the terms of physical characteristics and objects of geological/engineering sections. The Interpretation results are the some information about geological/engineering/objects "features" for survey area.

The steps from 3 to 5 can be recursive for results best conformity to selected mathematical models.

### **Discussion: Data abstraction levels**

For described Geophysical data acquisition and processing stages, we can mark follow *data abstraction levels*:

- 1) The Sensors Data Sets forming procedures deal with “own/specific” sensor type and characteristics (it is means physical sensor with its own features). All “specific” information is applied to data this stage.
- 2) The Field-distribution creates Procedures deal with the “abstract sensors” measurements and creates the abstract “geophysical field distribution” (there are value, time and space coordinates/attitude).
- 3) The Processing Procedures deal with physical field equations; the result of procedures is the inverse geophysical tasks’ decision (in its simplest form – the value of field source).
- 4) The Interpretation Procedures deal with transfer from physical field’s knowledge space to geological/engineering knowledge space. The inverse geophysical tasks’ decision interpreted in geological/engineering terms.

### **Discussion: Data acquisition and processing chart extended**

There is some contradiction for described “Geophysical data acquisition and processing stages”. The Sensors set forming procedures (stage 2) sometimes needs pre-processed navigation data (which formed in stage 3).

More carefully, the “Position (coordinates, attitude) calculations” in the stage 3 is own “navigation knowledge area” with an inverse navigation task. The Extended data acquisition and processing chart showed in *Figure 3*. There is new feature – *Pipe*, which transfer information between different “knowledge areas”.

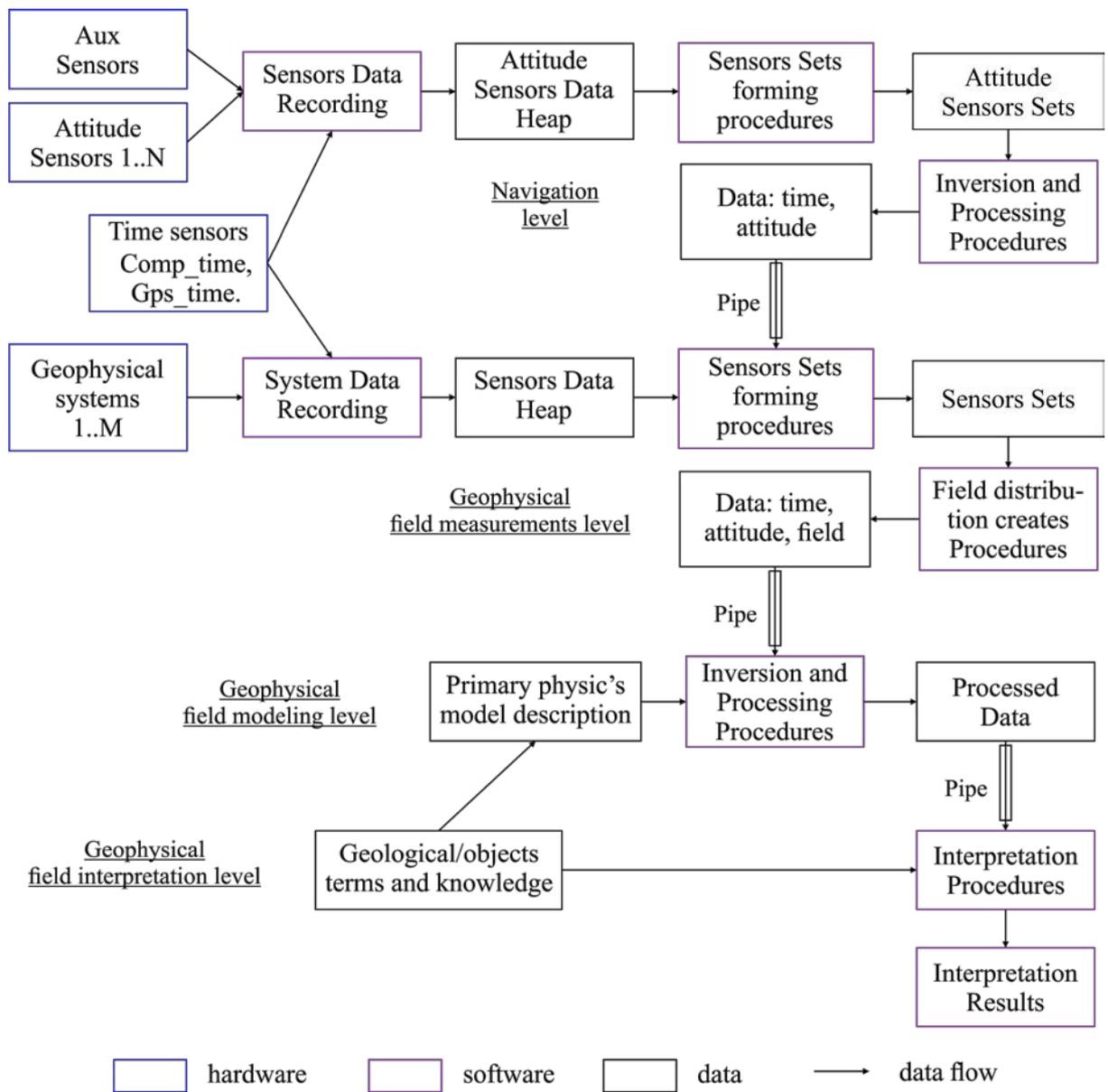


Figure 3 Data acquisition and processing chart (extended)

## 4 Data Content types

The follow text is concerned to general features for measurements' data *content types* or software variables containers. What does it means? For example, SegyMat library (read and write \*.sgy-files) organized variable with seismic-trace headers as trace-by-trace: Head(10).X – the variable contained X-coordinate scalar for trace number 10. If we need to send all X-coordinates contained in Head to some processing procedure input (the most rational input is X-vector), we need spend some forces to extract data from Head to X-vector and we need spend some forces to put procedure-output-vector to Head(1...end).X. To avoid this, it is reasonable to organized variable with seismic-trace headers Head.X(1..end). The “content types” means a variable “organizing” for different information/data with own specific features.

### Information-content

This content contained “constant” information about sensor's characteristics, measurements mode, etc. This information is applied to “data block unit”: one sensor from Data Heap or one survey lines' data. The *information-content* (Figure 4, a) examples: Ellipsoid characteristics, coefficients for depth sensor, coefficients for altimeter.

### Row-content

This content is contained measurements results along time axis (binding/linking source axis). All rows are designed as variables' fields by usability. All *row-content* (Figure 4, b) fields for data block unit (for example, one survey lines' data) have equal length. Row-content field examples, the measured along survey line: Easting coordinate, Northing coordinate, Water temperature, Pitch angle, Total magnetic field. There are several Row-content's *predefined field names* for general data types, which used by GeOMLib functions.

The row-content is the equipment measurements result with “scalar output”, when one measurement (along binding/linking source axis) is one number by one sensor. There are time-series scalar measurements: GPS messages elements, IMU/MRU messages elements, magnetic sensor data, etc.

### RowM-content

If we have a number of the same and synchronized sensors with row-content, it is reasonable to create matrix with a number of row. This matrix is alternate to several analogic Row-content fields. The *RowM-content* (Figure 4, c) example is magnetic measurements with gradiometer; there are a number (from 2 to 7) synchronized and the same magnetic sensors.

### Matrix-content

This content is contained measurements results along time axis (binding/linking source axis); each measurement is time-series inserted to matrix column. The *Matrix-content* (Figure 4, d) is the equipment measurements result with “time-series-output”, when one measurement (along binding/linking source axis) is the series numbers along time axis (by one receiver).

The matrix column examples are SBP or SSS traces. Other measurements (column/traces) characteristics (coordinates, time stamp, etc.) are contained in associated Row-content structure. The matrix columns number and Row-content field's length is equal.

### Dim3-content

If we have a number of receivers with matrix-content, it is reasonable to create additional axis with a sensor number; we will have 3-axis dimension or *Dim3-content*. A “number of receivers” example is digital-seismic streamer. There are dimension axes: 1) short time or short numbers; 2) record-time for hydrophones or recorded digits number; 3) receiver numbers.

Other measurements characteristics (coordinates, time stamp, etc.) are contained in associated Row-content structure (for example, seismic source coordinates) or RowM-content structure (for example, streamers’ hydrophones coordinates).

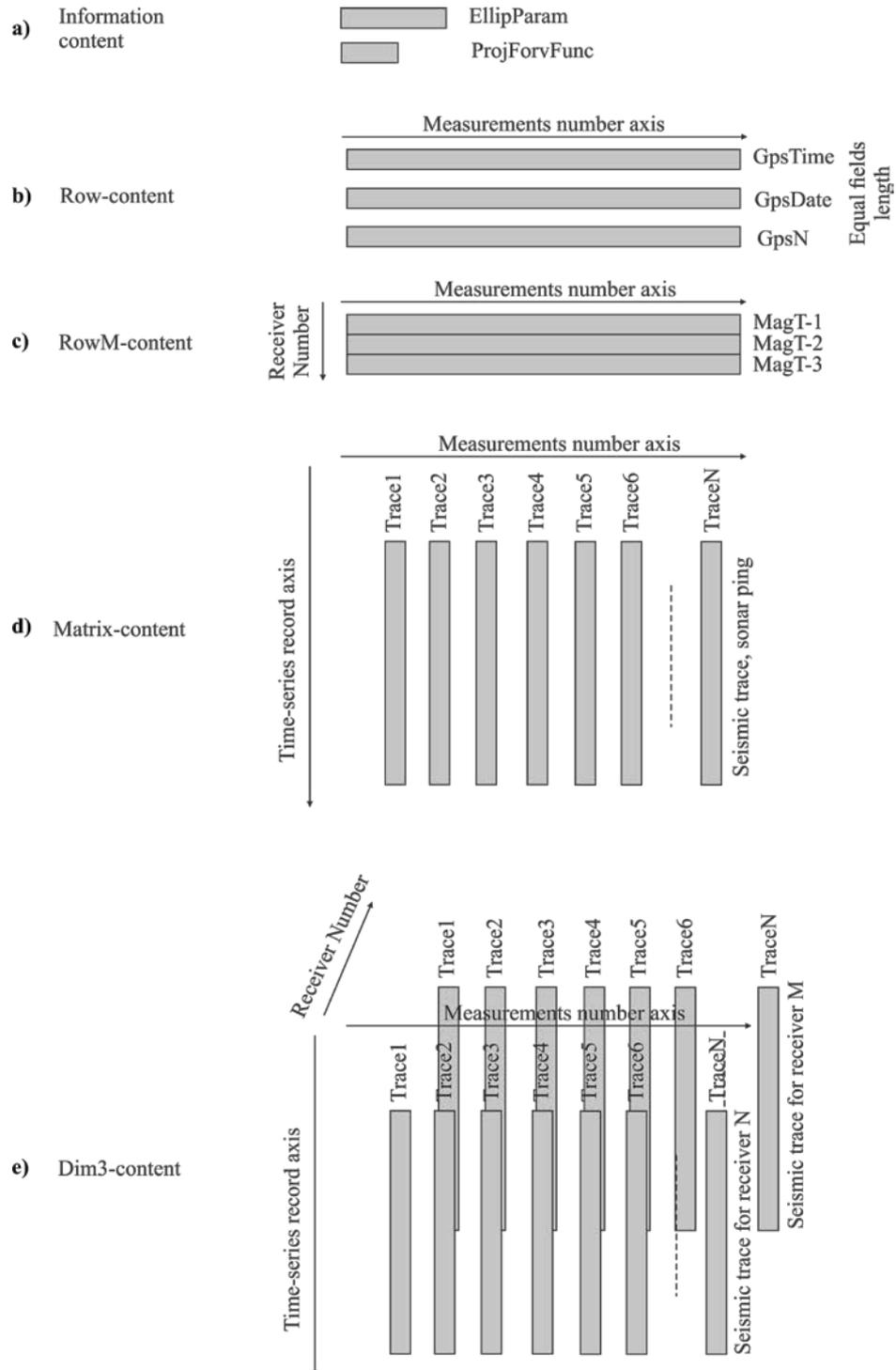


Figure 4 Measurements' data content types

## Dataset-content

**Dataset-content** is “composite” contents’ type (**Figure 5**). There are several variables: first variables type includes a number of Information/Row/RowM content components and formed as field’s vector; second variables type includes same number Matrix/Dim3 content components and formed as cell’s vector. Each vectors element is one “survey line”. Information/Row/RowM content’s fields are equal for each survey line.

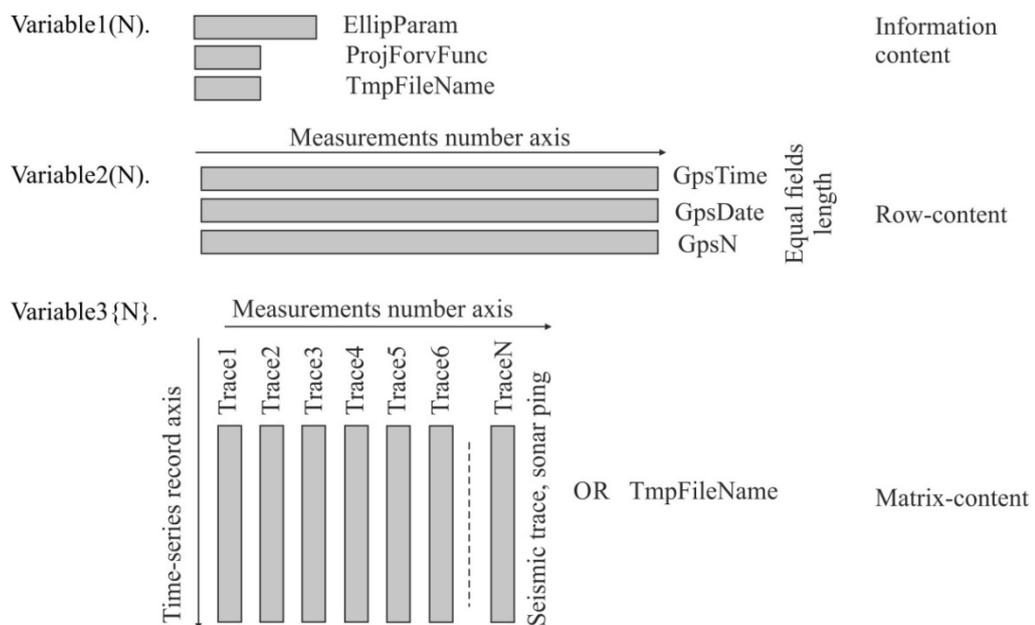
The Matrix/Dim3 content’s element can be saved as temp-file to hard drive for RAM-management. This case Matrix/Dim3 (cell’s vector element) and one field of Information content contains string with file name.

For example, for sgy-dataset we will have  $\text{Head}(1..10).\text{X}(1..\text{end})$ ,  $\text{Data}\{1..10\}$  where 1..10 – ten survey lines;

$\text{Head}(1).\text{X}(1..\text{end})$  – vector of X-coordinates for line number 1;

$\text{Head}(1).\text{X}(15)$  – X-coordinates of 15<sup>th</sup> short for line number 1;

$\text{Data}\{1..10\}$  – the cell of matrixes with traces data (or with temp-file names) for ten survey lines.



**Figure 5** Dataset content types

## Discussion: input functions types

The Information-content and Row content are realized as structures with fields; the field names are usually “predefined field names”, which “known” for procedures deal with structures (for example, functions for processing sgy-data are “understand” sgy-structures).

So, we have three types of functions by input/output:

- “**universal function**” with input parameter as vector, matrix, etc.;
- “**specified function**”, which “understand” structures with predefined names of fields (using Information and Row content);
- “**softly-specified function**”, with the fields names are transmitted to procedure.

## **5 Pre-defined structures field names**

### **5.1 General time reference fields**

### **5.2 General position/coordinates reference fields**

### **5.3 PolyLine structure**

### **5.4 Navigation's Sensors fields structure**

### **5.5 Pre-Processed Set fields structure**

## 6 Functions Glossary

### 6.1 gFields – Row content's functions

#### Manipulations with data structures fields

gFieldsRowGet – get data from structure, using field length as key;

gFieldsRowSet – set data to structure, using mask and field length as key;

gFieldsRowAppend – append rows contents for two structures;

gFieldsRowInsert – insert data to structure in defined position;

gFieldsRowInterp – interpolate pre-defined fields from time to another time (used interp1);

gFieldsCombine – append fields for two structures.

gFieldsCellsFindStr – find number of cell's vector element using string\_for\_find.

### 6.2 gData – Matrix content's functions

<https://ge0mlib.ru/g/gData.pdf>

#### Read/Write Data

gDataSave – save Data as size and IEEE-float bytes stream;

gDataLoad – load IEEE-float bytes stream to Data;

#### Data matrix manipulations

gDataTraceFilt – each own trace/ping filtration;

gDataTraceWeight – average traces/pings group;

gDataGainPL – traces/pings Gain;

gDataNormPL – normalized data-matrix (traces) using weight between polyline1 and polyline2;

gDataToPL – shifted data-matrix (traces) from polyline1 to polyline2;

gDataFillPL – fill data-matrix (traces) from Line1 to Line2 using number FillNum;

gDataCalcAttrib – calculate "attributes" for data-matrix (along traces/columns);

gDataResample – change “vertical axis sample interval” (create new point number using repeating or deleting);

#### Picking

gDataPLPickHandle – Handle Image/matrix picking (in pixels numbers).

gDataPLPickAuto – Auto Image/matrix picking (in pixels numbers).

#### MatLab Graphics

gDataDrawSection – draw Data matrix.

### 6.3 gNav – Navigation and coordinates manipulations

#### Coordinates/time/angle transformations

gNavAng2Ang – convert angle between formats;

gNavTime2Time – convert time between formats;

gNavDayCheck – check than date was changed in 00:00:00; correct bad date;

gNavDV2DV – convert distance and velocity between formats; (mile=1852m);

gNavEllipName2EllipParam – convert Ellipsoide Name to Ellipsoide Parameters [semi\_major\_axis first\_eccentricity];

gNavGeoc2Geoc1032 – convert geocentric coordinates to geocentric coordinates (7-parameters Helmert transformation);

gNavGeoc2Geoc1032inv – convert geocentric coordinates to geocentric coordinates (7-parameters Helmert transformation; direction strict inverse);

gNavGeoc2Geog – convert geocentric coordinates to geographic;

gNavGeog2Geoc – convert geographic coordinates to geocentric;

gNavGeog2ProjUtm – convert geographic coordinates to UTM/TM (EPSG 9807);

gNavProjUtm2Geog – convert UTM/TM (EPSG 9807) to geographic coordinates;

gNavGeog2ProjLambAz – convert Geographic coordinates to Lambert Azimuthal Equal Area projection (EPSG 9820);

gNavProjLambAz2Geog – convert Lambert Azimuthal Equal Area projection coordinates to Geographic (EPSG 9820);

gNavCoord2Coord – convert coordinates using SensorNavigation datum to ProjectNavigation datum (TargCodes can be used; see Figure 17);

gNavArc2Len – calculate length in meters along Latitude and Longitude arc in degrees;

gNavLen2Arc – calculate Latitude and Longitude arc in degrees from length in meters;

gNavCartesianTrans – Calculate scale and angles for Geographic-to-Cartesian coordinates transformation;

gNavReducingHeadingToProjUtm – Scale and Angle estimated between Lines in Geographic and Transverse Mercator (EPSG 9807) Projection.

#### **Coordinates/time manipulations**

gNavCoordDerepeat – remove and interpolate repeating position;

gNavTrackMadeGood2D – 2D-track "sophisticates" smoothing and direction calculation;

gNavTrackDirection2D – 2D-track "direction" calculate;

gNavTrackToLineStat – distance and another stats between Track and planned Survey Line from .LinePlan field.

gNavEuler – calculate 3D Euler matrixes for coordinates rotations;

gNavEulerLever – LeverArm's coordinates calculates using Head,Pitch,Roll;

gNavCoordNods – Nods coordinates calculation using follow Model: Metacenter Lever; Gps Lever/Data; MRU Lever/Data; Nods Levers;

gNavLayback – calculate "layback" position using streamer angle.

### **6.4 gMap – Geometrics tasks decision, trackplots/pipelines/lineplanning, picking, graphics**

#### **Geometrics tasks decision: minimal distance, cross, normal**

`gMapGeomPoints2DMinDist` – minimal distance search between points;

`gMapGeomSegments2DCross` – find cross point for segments;

`gMapGeomPointsSegments2DNormal` – find normal from point to segments;

`gMapGeomPointsPolyline2DNormal` – find normal from point to polyline;

`gMapGeomShiftZaxisPL` – Minimized mean difference (Z-axis) for each PL in PLs'-cross-points; shift PLs, using  $PL(n).pZ=PL(n).pZ+difference$

`gMapGeomLineDirect2D` – robust calculation a,b for  $y=ax+b$ ;

`gMapGeomPolyline2Line` – create Polyline points projection to Line in 2D-plane (3 projection's methods); create uniform points distance;

### **Trackplots / pipelines / lineplanning**

`gMapPLReadTxt` – read LinePlaning line's or PipeLine's coordinates from txt-file;

`gMapPLWriteTxt` – write Poly lines structure to txt-file;

`gFieldsDimFindStr` – find PL number using .PLName;

`gMapPLDraw` – draw figure with poly lines from PL-structure;

`gMapPL2AcadExport` – export Poly lines structure to AutoCad script;

### **Picking**

`gMapPickHandleNan` – set to Nan points on curve using handle picking.

### **Draw graphics**

`gMapTickLabel` – remove exponent from Tick Labels;

`gMapCursorCallback` – create tips with current position and additional data.

## **6.5 gLog – Serial data logging/read/write**

### **Data Logging (Free Pascal)**

`gComLog v.04 / gComLogOv v.01` – serial port logging;

`gComCC` – cable counter keyboard emulation.

### **Read/Write created by gComLog**

`gLogGpGguRead` – read \$GPGGU data from files;

`gLogGpGgaRead` – read \$GPGGA data from files;

`gLogGpGllRead` – read \$GPGLL data from files;

`gLogGpHdtRead` – read \$GPHDT data from files;

`gLogGpVtgRead` – read \$GPVTG data from files;

`gLogGpZdaRead` – read \$GPZDA data from files;

`gLogGpGstRead` – read \$GPGST data from files;

`gLogTss1Read` – read Tss1 structure from files;

`gLogFixRead` – read fixes structure from files created;

`gLogCableCounterRead` – read Cable Counter structure (T-Count, C-Max, gComCC, etc);

`gLogGeoSenseNavFixRead` – read \$CUSTOM data from log files Navigation-to-Geo-Resource;

`gLogGeoEelNavFixRead` – read data from log files Navigation-to-GeoEel;

## **Time stamp analysis and synchronization**

**gLogGpsCompTimeDelta** – calculate difference between Computer and GPS time;

**gLogGpsDayCalc** – calculate GpsDay using CompDay, CompTime, GpsTime;

**gLogCompTimeToTimeStamp** – change time in CompTime field to Time-Stamp time (nearest, forw, back);

## **Database**

**gLogFilesFind** – create Files List using start and final date/time;

**gLogFileFormatFix** – analyze strings from position-formated file.

## **6.6 gJsf – Jsf files read/write/manipulations**

*<https://ge0mlib.ru/g/gJsf.pdf>*

**gJsfHeaderRead** – Read JsfHead structure from \*.jsf file.

**gJsfHeaderWrite** – Write JsfHead structure to New \*.jsf file.

**gJsf0080Read** – Read [Head,Data] from JsfHead.fName (\*.jsf) file for Message Type 0080 (SBP or SSS Data Message).

**gJsf0080Write** – Write [JsfHead,Head,Data] to \*.jsf file for Message Type 0080 (SBP or SSS Data Message).

**gJsf0082Read** – Read [Head,Data] from JsfHead.fName (\*.jsf) file for Message Type 0082 (Side Scan Data Message).

**gJsf0086Read** – Read [Head,Data] from JsfHead.fName (\*.jsf) file for Message Type 0086 (4400-SAS Processed Data; 990-0000048-1000\_Revision:1.7/Nov2006 used).

**gJsf0182Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 0182 (System Information Message).

**gJsf0182Write** – Write [JsfHead,Head] to \*.jsf file for Message Type 0182 (System Information Message).

**gJsf0426Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 0426 (File Timestamp Message; 0004824\_REV\_1.18 used).

**gJsf0428Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 0428 (File Padding Message; 0004824\_REV\_1.18 used).

**gJsf2000Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2000 (Sonar Virtual Ports Data).

**gJsf2000Write** – Write [JsfHead,Head] to \*.jsf file for Message Type 2000 (Sonar Virtual Ports Data).

**gJsf2002Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2002 (NMEA String).

**gJsf2002Write** – Write [JsfHead,Head] to \*.jsf file for Message Type 2002 (NMEA String).

**gJsf2020Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2020 (Pitch Roll Data).

**gJsf2020Write** – Write [JsfHead,Head] to \*.jsf file for Message Type 2020 (Pitch Roll Data).

**gJsf2060Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2060 (Pressure Sensor).

**gJsf2060Write** – Write [JsfHead,Head] to \*.jsf file for Message Type 2060 (Pressure Sensor).

**gJsf2080Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2080 (Doppler Velocity Log Data).

**gJsf2090Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2090 (Situation Message).

**gJsf2091Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2091 (Situation Comprehensive Message, version 2; 0004824\_REV\_1.18 used).

**gJsf2100Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2100 (Cable Counter Data Message).

**gJsf2101Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2101 (Kilometer of Pipe Data; 0004824\_REV\_1.18 used).

**gJsf2111Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 2111 (Container Timestamp Message).

**gJsf3000Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3000 (Bathymetric Data Message Type; 0014932\_REV\_D March 2016 used).

**gJsf3001Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3001 (Attitude Message Type; 0014932\_REV\_D March 2016 used).

**gJsf3002Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3002 (Pressure Message Type; 0014932\_REV\_D March 2016 used).

**gJsf3003Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3003 (Altitude Message Type; 0014932\_REV\_D March 2016 used).

**gJsf3004Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3004 (Position Message Type ; 0014932\_REV\_D March 2016 used).

**gJsf3005Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 3005 (Status Message Type; 0014932\_REV\_D March 2016 used).

**gJsf9001Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 9001 (DISCOVER II General Prefix Message; 0004824\_REV\_1.18 used).

**gJsf9002Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 9002 (DISCOVER II Situation Data Message; 0004824\_REV\_1.18 used).

**gJsf9003Read** – Read Head from JsfHead.fName (\*.jsf) file for Message Type 9003 ( DISCOVER II Acoustic Prefix Message; 0004824\_REV\_1.18 used).

**gJsf0000Read** – Read Head and Data from JsfHead.fName (\*.jsf) file for Private Messages Types.

## **6.7 gXtf – Xtf files read/write/manipulations**

### **Xtf files read and write**

**gXtfHeaderRead** – Read XtfHead structure from \*.xtf file;

**gXtfHeaderWrite** – Read XtfHead structure to New \*.xtf file;

**gXtf000Read** – Read Message Type 000;

**gXtf000Write** – Write Message Type 000;

**gXtfFilesAppendByHeader** – Append Xtf files.

gXtf000Dir2PLRead – read coordinates form Directory with \*.xtf to PL-structure;

gXtf000Xtf2JpMat – convert sgy to jpeg-file and 7z arcive with headers;

gXtf000JpMat2Xtf – create sgy from jpeg-file and 7z arcive with headers;

#### **Xtf structure manipulations**

gXtfFileSetMessOff – mark messages typed in Head to Delete from Xtf structure;

gXtf000DeleteSlant – delete slant-range borders from Data in Xtf000 structure;

gXtf000SlantCorrect – convert two SSS channels to “belt”;

gXtf000BeltPoints2Pings – re-calculate points from “belt” to SSS-pings;

gXtf000PingsShiftToPoints – re-calculate coordinates and angles of pings to points-framework (move SSS to MBES);

#### **MatLab Graphics**

gXtf000DrawSection – draw SSS channels.

## **6.8 gSgy – Sgy files read/write/manipulations**

### **Seg-y files read and write**

gSgyRead – Read \*.sgy file;

gSgyWrite– Write \*.sgy file;

gSgyDatasetImport – add segy-files from file-list or folder to Dataset;

gSgyDatasetPack – save data-matrixes from Data to temporary files (temp-files names are put in Data);

gSgyDatasetExport – exported segy-files from Dataset to specified folder;

gSgyJpMatWrite – convert sgy to jpeg-file and mat-file with headers;

gSgyJpMatRead – create sgy from jpeg-file and mat-file with headers;

### **Seg-y structure manipulations**

gSgyDTEN\_on – create fields GpsDate, GpsTime, GpsE, GpsN in Project’s datum from sgy-fields which in Sensor’s datum;

gSgyDTEN\_off – create sgy-fields in Sensor’s datum from GpsDate, GpsTime, GpsE, GpsN in Project’s datum;

gSgyDirLines2PL – read coordinates form Directory with \*.sgy to PL-structure;

gSgyDatasetLines2PL – put coordinates from Dataset to PL-structure;

gSgyDataset2P190Dir – read sgy-structures form Directory, and create P190 files;

gSgyAppend – append Seg-y lines from Dataset;

gSgyAppendFiles – append Seg-y files to [SgyHead,Head,Data];

gSgySetTraceOff – delete traces from segy-structure;

gSgySetEndRecTime – change segy's Traces Length;

gSgySetDelayRecTime – change segy's DelayRecordingTime for each trace;

gSgyReSample – change “vertical axis samplpe interval” (create new point number using repeating or deleting);

gSgyShiftZero – shift sgy-section's start-time. Apply shifting's part to DelayRecTime; use gDataToPL for another shifting part.

gSgyDataToPL – shift sgy-section's from PL1 to PL2. Apply shifting's part to DelayRecTime; use gDataToPL for another shifting part.

gSgyUnificate – create Unificate data sets: Head.SampleNumber=const (SgyHead.FixedLengthTraceFlag=1); Head.DelayRecordingTime=const; time\_step=const (interp Head and insert Nan to Data); (flag for HOLES)

gSgyHeadNanInterpSBP – interpolate Sgy-header structure from vector1 to vector2;

gSgyTime2DepthPL – convert time-section to depth section using PL-structure as peaking horizons; horizons velocity is constant; based on gDataToPL and gDataResample.

gSgyProcessingLogging(on/off) – write processing steps to text header (like to buffer);

### **Modelling**

gSgyPipeDiffraction – calculate hodograph from pipe.

### **MatLab Graphics**

gSgyDrawSection – draw Segys' Data matrix; create tips with hyperbola (diffraction point);

gSgyDraw3Section – draw Segys' Data matrix; create tips with hyperbola (diffraction point).

gSgyPLProjToSection – calculate PL's projection to Sgy-section (example: well projection);

gSgyDrawPLProj – draw PL's projection to Sgy-section (example: well projection).

## **6.9 gMagy – Magnetometers data read/write/manipulations**

### **Magnetometers files read/write**

gMagyG880IntRead – read MagLog Geometrics program INT-file;

gMagyG882IntRead – read MagLog Geometrics program INT-file;

gMagyG88MagRead – read MagLog Geometrics program MAG-file;

gMagyG88MagWrite –

gMagyG88tvMagRead – read MagLog Geometrics program MAG-file for 2 sensors (TVG);

gMagyG88tvMagWrite –

gMagyG88GpsRead – read MagLog Geometrics program GPS-file;

gMagyG88GpsWrite –

### **Magnetic field modeling**

gMagyNormal – create normal magnetic field vector;

gMagyMonopol – calculate magnetic field from monopole;

gMagyDipol – calculate magnetic field from dipole;

gMagyCylinder – calculate magnetic field from horizontal dipoles chain;

gMagySurf – calculate magnetic field from grid;

gMagyModelScript –

### **MatLab Graphics**

**gMagyDrawWiggle** – draw Magy wiggle on 2D plane;

**gMagyDrawWiggleNan** – draw Magy wiggle (with out-of-data by mask) on 2D plane;

**gMagyCursorCallback** – display and copy to clipboard the data tips (points) position.

## 6.10 gUhr – Ultra High Resolution Seismic data read/write/manipulations

**gUhrGeoEel\_NavLogRead** – Read Navigation log from GeoEel station software.

**gUhrGeoSense\_DirRenameShotId2FFID** – Rename Seg-d files in DirName, using FFID and FIX from Log-file was created Geo-Sense LWStreamer's aquisition program (\*.log)

**gUhrGeoSense\_NavLogRead** – Read "RAW\_LOG-Input 0.log" from MultiTrace station software; messages were formed as "\$CUSTOM,20170906,221035.94,453,513681.21,8006048.81,50.16,65.2".

**gUhrGeoSense\_StreamerLogRead** – Read data from Log-file was created Geo-Sense LWStreamer's aquisition progtram (\*.log)

## 6.11 gGrav – Gravimeters data read/write/manipulations

### Gravity field modeling

**gGravSphere** – calculate gravity field form ball;

**gGravCylinder** – calculate gravity field form horizontal cylinder;

**gGravSurf** – calculate gravity field form grid.

## 6.12 gP190 – P190 files read/write/manipulations

<https://ge0mlib.ru/g/gP190.pdf>

**gP190Read** – Read P190 Type\_1 (Grid or Geographical coordinates) without Item 16 (RecordId='R': Receiver group records for 3-d offshore surveys).

**gP190Write** – Write P190 Type\_1 (Grid or Geographical coordinates) without Item 16 (RecordId='R': Receiver group records for 3-d offshore surveys).

## 6.13 gWfr – Images with coordinate world-files and XYZ-grid-files manipulations

<https://ge0mlib.ru/g/gWfr.pdf>

**gWfrRead** – Read raster-image with world-file.

**gWfrGeotiffRead** – Read Geotiff-image file and copy values to world-file's structure from tiff's geo-tags.

**gWfrWrite** – Write raster-image with world-file.

**gWfrDraw** – Draw image with world-file's data.

**gWfrIm2Xyz** – Convert image to XYZ-grid-data.

**gWfrXyz2Mat** – Convert XYZ-grid-data to matrix-image.

**gWfrMat2Im** – Convert matrix-image to raster-image.

**gWfrIm2Mat** – Convert raster-image to matrix-image.

**gWfrCut** – Cut rectangular zone from image.

**gWfrAdd** – Adds/cut borders to image.

**gWfrResize** – Resize image.

**gWfrCombineColors** – Combine colors/palettes for raster-image1 and raster-image2

**gWfrMergeData** – Merge Image1 and Image2 using world-data.

## 6.14 gAcad – AutoCad scripts creation

<https://ge0mlib.ru/g/gAcad.pdf>

**gAcadColor** – Set color to AutoCad script file.

**gAcadZoom** – Write "zoom" to AutoCad script file.

**gAcadLayerMake** – Write "make layer (create and set active)" to AutoCad script file.

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

**gAcadText** – Write "write texts in X(:),Y(:) coordinates, with size FontSize(:) and angle TextRotAngle(:)" to AutoCad script file.

**gAcadPline** – Write "draw one Pline in X(:),Y(:) coordinates" to AutoCad script file.

**gAcadPolygon** – Write "draw one Pline with Hatch in X(:),Y(:) coordinates" to AutoCad script file. The Pline is auto closed (first point will be added to end).

**gAcadBitmapInsert** – Write "insert bitmap (from fReff path) in XY coordinates" to AutoCad script file.

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

**gAcadGraphToMap** – Write "draw one TrackPlot in X(:),Y(:) coordinates with Graph (relative track)" to AutoCad script file.

**gAcadWiggleToMap** – Write "draw one WiggleToMap in X(:),Y(:) coordinates" to AutoCad script file.

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

## 6.15 gZ – Ancillary functions

**gZTxtRead** – read from text file with Title strings and formatted data columns;

**gZTxtWrite** – write to text file with Title strings and formatted data columns;

## Citation