

GeOMLib

Библиотека для обработки данных морской инженерной геофизики. Основные понятия

Vol. 01-01

Дмитриев И.В.
29.12.2023

Содержание

1	Введение.....	4
2	Составляющие измерений геофизического поля.....	8
3	Стадии получения и обработки данных геофизической съемки.....	10
4	Уровни абстрактности данных.....	16
5	Формы представления данных («контейнеры»).....	18
6	Переменные-структуры и уровни их формализации.....	21
6.1	Обзор структур.....	21
6.2	Структуры со строго-заданными именами полей, постфиксы и префиксы.....	23
6.3	Вектор-структура.....	25
6.4	Основные переменные с заданными именами.....	26
6.4.1	SU (структура с настройками проекта).....	26
6.4.2	ML («картографические данные»).....	29
6.5	RS (фактические результаты измерений).....	30
6.5.1	PR (записи датчиков по профилю съемки).....	31
6.5.2	GL (результаты обработки геофизической съемки).....	33
7	Скрипты с графами обработки.....	34
7.1	Команды скриптов.....	34
7.2	Пример MatLab-сессии с расчетом геометрии, при съемке СВР.....	37
7.3	Журнал обработки профилей съемки.....	39
7.4	Команды-функции для скрипта.....	40
	Литература.....	41

Список таблиц

Table 1.1 Группы функций библиотеки GeOMLib.....	4
Table 6.1 Пример строго-заданных имен полей с данными навигации	23

Список рисунков

Figure 3.1 Блок-схема стадий получения и обработки данных геофизической съемки	10
Figure 5.1 Графическое изображение «форм представления» результатов измерений	20
Figure 6.1 Типы функций для работы с составными структурами	21
Figure 6.2 Пример значений полей переменной SU (расчет геометрии для съемки СВР)	28
Figure 6.3 Пример полей ячейки ML{1} (план линий съемки)	30
Figure 6.4 Пример полей PR{1} (расчет геометрии сейсмического источника)	32
Figure 6.5 Пример полей PR{1}.LLog (расчет геометрии сейсмического источника)	32
Figure 7.1 Пример содержимого журнала PR{n}.ProcLog (расчет геометрии съемки СВР).....	39

1 Введение

Библиотека функций предназначена для обработки данных морской инженерной геофизики (навигация, непрерывное сейсмическое профилирование, гидролокация бокового обзора, гидромагнитная съемка, геометрия для многоканальной сейсморазведки СВР/ССВР). Функции обработки реализованы в MatLab2018b (<https://mathworks.com/>) и адаптированы к свободно распространяемому ПО GNU Octave 8.4.0 (www.octave.org). Утилиты регистрации данных реализованы во Free Pascal (<https://www.freepascal.org/>). Сайт проекта: <https://ge0mlib.com/>.

Ge0mlib предоставляет удобное пространство (предопределенные поля, структуры переменных, базовые функции) для обработки геофизических и навигационных данных. Особенности библиотеки ge0mlib:

- быстрое создание сценариев и написание компактного кода;
- легкий доступ к каждому байту навигационных и геофизических данных;
- удобные структуры переменных для решения геофизических задач;
- все датчики могут быть обработаны как единый набор данных (например, поправки в данных MRU могут быть легко применены к координатам в файлах Seg-y, Jsf, Xtf и P1/90);
- полный доступ к коду функций и скриптов, алгоритмам и математике.

Библиотека включает несколько групп функций, описание которых приведено в [Table 1.1](#).

Table 1.1 Группы функций библиотеки Ge0MLib

Название	Код	Описание
	01	Базовые типы данных и функции библиотеки
	01-01	Концепции библиотеки Ge0MLib
gData	01-02	Работа с данными вида Matrix
gFields	01-03	Работа с данными вида Row, RowM
	02	Чтение и запись геофизических данных
gJsf	02-02	Работа с файлами Jsf (чтение, запись, работа с полями данных)
gXtf	02-03	Работа с файлами Xtf (чтение, запись, работа с полями данных)
gSgy	02-04	Работа с файлами Sgy (чтение, запись, работа с полями данных)
gAcad	02-05	Создание скриптов AutoCad
gP190	02-06	Работа с файлами P1/90 (чтение, запись, работа с полями данных)
gKml	02-07	Экспорт данных в формате Kml (Google Earth)
	03	Регистрация данных
gLog	02-01	Запись в файл потоков сообщений с последовательного порта, их привязка ко времени (FreePascal). Функции чтения записанных файлов
	04	Навигация
gNav	04-01	Работа с данными навигации и преобразования координат
gMap	04-02	Решение задач геометрии, построение треков аппаратуры (карт фактического материала), карт планируемых линий, пикирование горизонтов, построение графики

Название	Код	Описание
gWfr	04-04	Работа с матрицами и графическими файлами, имеющими координатную привязку (coordinate world-files); работа с XYZ-данными интерполированными к равномерной сети
	05	Батиметрия (однолучевые и многолучевые эхолоты)
	06	НСП (сейсмические профилографы)
gSgy	02-04	Работа с файлами Sgy (чтение, запись, работа с полями данных)
	07	Многоканальное сейсмическое профилирование (HR и UHR)
gUhr	07-01	Чтение логов сеймостанций и контроллеров сейсмических источников, расчет геометрии для съемок СВР с ССВР
	08	ГЛБО
	09	Магнитометрия
gMagy	09-01	Обработка данных морской магнитометрии

Лирическое вступление

Первые функции были написаны примерно в 2013 году, чтобы "поиграться" с форматом seg-y. Тогда уже существовала библиотека seg-y read, но в ней служебная информация (координаты, время и тд) располагалась в структуре с индексом для каждой трассы. Это было неудобно, – расписывание координат по сеймотрассам требовало или цикла или выполнения преобразования через массив ячеек. Поэтому функции чтения-записи seg-y были переписаны так, что каждому заголовку сеймотрассы соответствовал вектор-строка, который помещался в поле структуры.

Через полгода понадобилось "откорректировать" координаты в seg-y файлах и "подвинуть" сейсмические трассы под морское дно для большого объема данных SBP. Нюансов выполнения я не помню, помню, что дело было срочное, этим занималось три человека и процесс шел уже несколько дней; при этом использовалось несколько программ, среди которых были "тяжеловесные", в том числе ProMax SeeSpace. Написанный в тот вечер скрипт в MatLab, справился с этой задачей за ночь. Вероятно, сильно повезло, что он отработал все 15 часов без сбоев и произвел сильное впечатление "сверхскоростной обработкой". Впечатление было настолько сильное, что я какое-то время отдавал предпочтение среде MatLab даже перед специализированными программами обработки SBP.

Почему MatLab?

Если бы язык для библиотеки выбирался сегодня, то вероятнее это был бы Python в силу "раскрученности" и использования в GIS-системах, и, вероятнее всего, это было бы ошибкой. Но Python на тот момент был еще не распространен, а Fortran уже не распространен. В MatLab привлекало:

- (1) Язык был математическим, то есть не надо было искать специальную библиотеку для вычисления детерминанта матрицы, – сам синтаксис языка уже включал основную математику. Просто достаточно было написать $\det(a)$;

- (2) Любая переменная могла "быть" матрицей или вектором, соответственно отпадала необходимость в циклах для расчетов, вместо `for n=1 to 100, a(n)=a(n)+5;end`, достаточно было написать `a=a+5`;
- (3) Пункты 1-2 приводили к очень компактному коду. То, что в других языках заняло бы 10-20 строк, в MatLab требовало написания одного выражения. Компактность приводила к тому, что в самом коде виден "формализованный ход решения", а взгляд не увязает во вспомогательных конструкциях;
- (4) Язык не компилируемый, а интерпретируемый как тот же Python или ранее Basic. То есть исполняемыми/распространяемыми файлами являются не условные *.exe, а сам код программы. Командное окно языка выглядело чем-то вроде "продвинутого калькулятора", а простые программы на нем (с точки зрения инженера) близки к "последовательности действий на калькуляторе", которая просто записана в отдельный файл.

Чаще всего (4) воспринимается как минус. Однако есть ниши, в которых это является плюсом. Например, "проходные" задачи с простейшими расчетами и с неустойчивыми входным и выходным форматами данных. Пример: пусть при выполнении геофизической съемки внимательный on-line surveyor нашел 10 "недочетов" в формате логов с координатами; каждый раз он заботливо исправлял очередной недочет.. после чего, с чувством исполненного долга вывалил перед обработкой 50 файлов с идеально записанными логами и еще 100500 файлов с просто-логами в разных форматах. Для откомпилированной программы, в таких случаях требуется переформатирование входных данных вручную (например, в текстовом редакторе) или наличие в самой программе "блока форматирования" для входных данных (то есть меню с выбором разделителей или позиций данных с в строке, выбором какая "колонка" какой тип данных содержит и другими нюансами). Не будем погружаться в пучину ужаса с "блоками форматирования" или с 2млн кликов мышкой в Excel, или в чуть более сложную задачу – программирование "блока форматирования" для выходных данных. Скажем только, что проще всего для "проходной" задачи, – подкорректировать в самом коде строчку вида `"%f:%f:%f,%0.3f;%0.6f;%s/n/r"`, что более естественно выполняется для интерпретируемого, а не компилируемого языка (а еще проще – просто написать за 4 минуты скрипт и привести данные к единому формату).

Можно выделить четыре области применения для "программ, написанных инженером"

Каждая из четырех областей обладает своей спецификой:

- 1) Тривиальные простые случаи. Что бы автоматизировать простые действия, которые можно выполнить при помощи "универсальных программ", но такое выполнение заняло бы большое количество времени. Пример приведен выше; еще один пример – пересчет координат сейсотрасс на другой эллипсоид; выполнение без программирования: выгрузка координат из seg-у при помощи SeiSee в текстовый файл, переформатирование текстового файла в текстовом редакторе, пересчет координат в геодезическом калькуляторе и запись в текстовый

файл, переформатирование текстового файла в текстовом редакторе, загрузка координат в seg-у при помощи SeiSee.

- 2) Нетривиальные простые случаи. Когда формат данных достаточно редок, что бы его могли прочитать "универсальные программы", но сама задача не требует применения продвинутой математики и методов обработки. Пример, – ввод сдвига для отметок времени в Jsf-файле (ссылка на пример).
- 3) Сложные задачи. Требуют относительно сложных математических вычислений, с которыми могут справиться только специализированные библиотеки. Как правило "универсальные программы общего назначения" такие задачи не решают.. пока эти задачи со сложными вычислениями не переходят в категорию распространенных (как, например, происходит с GPT-чатом). Если вы имеете дело в такими задачами, это означает что вы недалеко от передней кромки прогресса и ПО общего назначения для этой области пока еще нет.. ну или просто ваша задача слишком не-мейнстримная. Пример библиотеки MatLab для-сложных-вычислений – Nav-lab; библиотека предназначена для вычисления пути AUV по инерциальным системам при потере сигнала USBL.
- 4) Новое знание. Когда надо создать что-то принципиально новое, для чего и нужен алгоритмический язык. Если у вас есть такое "новое", то вы где-то на переднем крае развития инженерной или научной мысли. Пример библиотеки – n-vector. Как правило такие библиотеки компактны по коду, но содержат то самое "новое", чего до этого в мире не существовало (или существовало в завуалированной форме).

Библиотека ge0mlib ориентирована на применение в областях 1-2. При этом файлы библиотеки написаны для непрофессиональных программистов "в процедурном" стиле, без элементов ООП и с как-можно-более-простыми структурами данных и синтаксисом. Применение библиотеки облегчит реализацию кода для случаев 3-4, однако сама она не содержит каких-либо мега-алгоритмов или "прорывных" функций в области обработки данных.

В текущий момент функции адаптируются под GNU Octave. Он совместим с MatLab на уровне синтаксиса и не требует покупки лицензии. Просто ставим-и-работаем.. как в Python)

2 Составляющие измерений геофизического поля

Измерения геофизического поля характеризуются следующими «составляющими» (geophysical data components):

- временная привязка (синхронизация) измерений;
- координатная привязка (координаты, углы наклона, при необходимости – ускорения для всех объектов измерительной системы);
- значения измеренного физического поля (включая характеристики излучаемого импульса, при «активных» измерениях, а также аппаратурно-регистрируемые величины, характеризующие качество измерения геофизического поля).

Данные составляющие (в широком смысле), сводятся к описанию **распределения физического поля во времени и пространстве**. Каждая из «составляющих» формируется при помощи датчиков и алгоритмов, относящихся к определенной области специализации:

- 1) Данные с «**датчиков времени**». Обычно источником привязки являются навигационное оборудование или часы компьютера;
- 2) Данные с «**датчиков аттитюды/координат**» для объектов измерительной системы. Источником является навигационное оборудование;
- 3) Данные с «**датчиков физического поля**». Источником является геофизическое оборудование.

Таким образом, если цель геофизических измерений формулировать как «получение распределения физического поля во времени и пространстве», то результаты работы систем навигации, являются неотъемлемой частью измеренного геофизического поля. Плохое позиционирование геофизической аппаратуры, автоматически приводит к некондиционности результатов геофизической съемки.

Работа датчиков и измерительных систем осуществляется «во времени»; при этом, в большинстве случаев, время является основой для привязки измерений и синхронизации различных измерений между собой (с целью использования в дальнейших расчетах). В библиотеке GeOMLib реализовано два взаимосвязанных «**датчика времени**»:

- источником «глобального» времени являются данные GNSS, которые могут сопровождаться импульсом синхронизации PPS (в переменных, данный тип времени хранится в полях GpsTime, GpsDay и подобных);
- источником «локального» времени являются часы компьютера, выдающие разрядность значений до 0.001 или 0.000001 секунды (в переменных, данный тип времени хранится в полях CompTime, CompDay и подобных; запись данных с привязкой к компьютерному времени реализована при помощи группы функций gLog).

Часы компьютера используются для временной привязки внутри секундного интервала, задаваемого GNSS; в то же время, на основе GNSS можно определить уход компьютерных часов на длительных временных интервалах. Кроме того, использование двух источников времени позволяет осуществлять взаимный контроль устойчивости работы «датчиков времени».

Функции gLog используются для присвоения каждому первому байту потока сообщений, поступающих на компьютер через последовательный порт, временной метки. Если на компьютер поступает поток сообщений со временем GNSS-системы (далее – UTC), то все потоки сообщений, регистрируемые на данном компьютере, могут быть привязаны к UTC; таким способом синхронизируются записи, осуществляемые на разных компьютерах и не имеющие привязки к UTC в формате собственных передаваемых сообщений. Для точной привязки к UTC, необходима также регистрация времени прихода импульса PPS, по часам компьютера (импульс заводится не компьютер при помощи PPS-box).

«Датчики аттитюды/координат» используются для определения положения «датчиков физического поля». Задачи позиционирования, как правило, являются типовыми с хорошо проработанной и описанной математикой; примеры типовых задач:

- вычисление позиции для точек установки геофизических датчиков, жестко смонтированных к корпусу судна (включая штанги, выстрелы и иные устройства для закрепления оборудования); используются приемники GNSS, MRU/IMU, гироскопы, датчики осадки судна;
- вычисление позиции буксируемого оборудования на основе “Layback”; используется позиция точки буксировки, длина вытравленного кабеля, показания датчика глубины и альтиметра на буксируемом устройстве;
- вычисление позиции забортного оборудования при помощи USBL; используются показания приемников GNSS, MRU/IMU, гироскопов, датчиков осадки судна, системы USBL (время пробега акустического сигнала до маяка-ответчика и углы пеленга на маяк-ответчик).

При расчетах желательно применять расчет позиции по нескольким системам, для взаимного контроля и учета ошибок; например, грубые ошибки в показаниях USBL можно обнаружить при сопоставлении с длиной вытравленного кабеля, а значения «плохих» измерений можно пересчитать с помощью адаптивных расчетов по “Layback”.

В результате «определения положения датчиков физического поля» рассчитываются координаты датчика, при необходимости – углы наклона датчика, ускорения датчика и иные величины. Положение датчиков сохраняется в переменных, в полях со следующими именами: GpsLat, GpsLon, GpsE, GpsN, GpsH, Head, Pinch, Roll, CableLen, Depth, Altitude, WaterDepth и иных подобных.

3 Стадии получения и обработки данных геофизической съемки

Стадии, на которые разделяется процесс получения и обработки данных геофизической съемки, показаны на *Figure 3.1*. Эти стадии заложены в основу «последовательности обработки данных» в библиотеке GeOMLib. Описание каждой стадии приведено ниже.

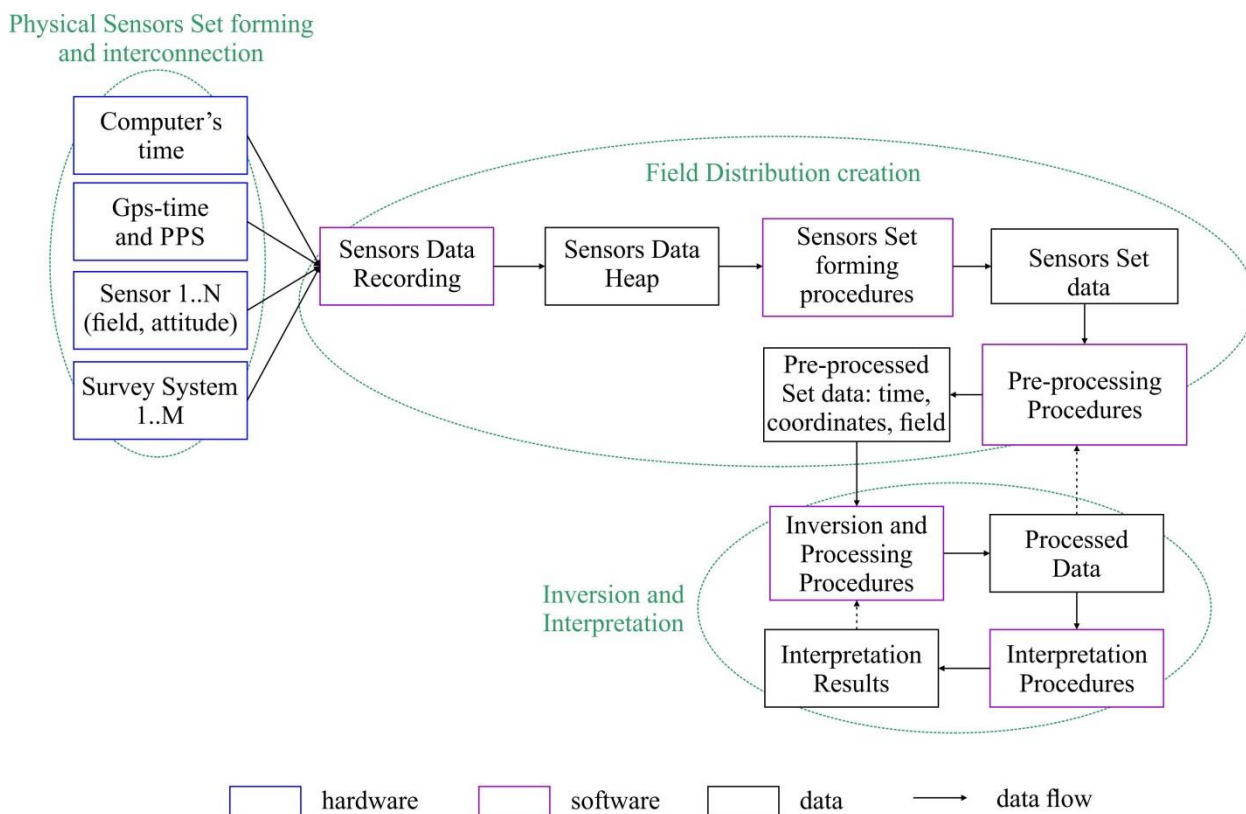


Figure 3.1 Блок-схема стадий получения и обработки данных геофизической съемки

(0) Подключение и установка датчиков (оборудования, аппаратурных комплексов) // The Sensors/Hardware set forming and interconnecting

На данной стадии, в соответствии с задачами геофизической съемки, выбирается геофизическая и навигационная аппаратура, производится ее установка, подключение, выполняется разводка потоков цифровой информации для обмена сообщениями между различными комплексами аппаратуры. Примеры задач геофизической съемки: проведение НСП для изучения приповерхностного геологического разреза; выполнение магнитной съемки для обнаружения локальных техногенных объектов (УХО).

Примеры датчиков/систем геофизического оборудования:

- датчики НСП, при условии, что излучатель и приемник выполнены как «одна точка»;
- датчики НСП, при условии, что сейсмический источник и приемники выполнены как «разные устройства»;
- датчики для сейсмической съемки с несколькими сейсмическими источниками и многоканальным сейсмическим приемником;

- датчики магнитометрической или градиентометрической (использование нескольких магнитных датчиков) съемки;
- датчики для съемки ГЛБО;
- датчики для съемки однолучевым или многолучевым эхолотом.

(1) Геофизическая съемка и запись показаний датчиков // *Data acquisition/recording*

На данной стадии производится запись потоков данных с датчиков, выполняемая при помощи программного обеспечения аппаратуры и/или gLog. Для дальнейшей обработки данных необходимо предусмотреть привязку измерений ко времени. Привязка осуществляется приписыванием к моменту измерения датчика «временных меток», генерируемых «датчиками времени» (или иных меток, позволяющих осуществить дельнейшую взаимную увязку данных). Результатом данной стадии является получение **фактического материала съемки (Sensors Data Heap)**, который содержит записи всех датчиков в цифровом формате и с привязкой ко времени и, чаще всего, представляет собой набор файлов с данными на цифровом носителе. Такой набор файлов также будем называть «первичными результатами съемки» (on-line survey results) или «сырыми данными» (Raw Data).

(2) Процедуры и результаты первичной обработки // *The Sensors Set data forming*

Первым шагом в обработке данных является «первичная обработка» и получение первично-обработанных данных (Sensors Set). При этом для фактического материала, полученного с каждого датчика, выполняется следующая последовательность **процедур первичной обработки**:

- (а) Импорт данных из файлов (который может сопровождаться поиском файлов внутри каталога, по заданному времени записи файла). Импорт в Ge0MLib, осуществляется в ячейки переменных RS или PR (см. описание ниже).
- (б) Привязка данных, полученных с различных датчиков, к единому времени (часам компьютера или времени GNSS) и обрезка «данных с датчика» по заданному временному интервалу. Временной интервал, как правило, заключен между временем начала и временем окончания профиля геофизической съемки, которое регистрируется в Журнале оператора. В Ge0MLib, для размещения «данных с записью датчика по профилю съемки» используются поля переменной PR (см. описание ниже).
- (с) Удаление «плохих» измерений из фактического материала (сырых записей датчика). Природа «плохих» измерений, как правило, не связана с физикой измеряемого явления, а является отражением сторонних процессов (например, скачка питающего напряжения на измерительном приборе). Можно выделить два типа измерений, определяемых как «плохие»:
 - штормовые выбросы (spikes) и иные «особенности в поведении записи датчика во времени», которые не могут быть следствием физики измеряемой величины;
 - значения с прикрепленной меткой «данные плохого качества»; такого типа метки (как логические, так и числовые) могут быть предусмотрены самим датчиком и являются

отражением аппаратурного контроля процесса изменений или могут быть получены при обработке, в результате расчетов;

(d) Выполнение процедур сглаживания/осреднения/фильтрации измерений, основанные на характеристиках датчика; примеры:

- уменьшение нормально-распределенного высокочастотного шума,
- сглаживание данных с учетом выходного разрешения датчиков (что бы, при дальнейшей обработке, избежать скачкообразных изменений показаний датчика),
- учет амплитудно-частотных характеристик датчика (фильтрация, для удаления компонент сигнала вне указанного диапазона частот, как заведомого «шума»), например, для приемников сейсмического сигнала,
- пересчет из единиц АЦП или напряжения в физические величины: применение калибровочных коэффициентов, учет нелинейности датчика.

Таким образом, на стадии первичной обработки к результатам измерения датчика применяются его «специфические характеристики»; при получении Результатов первичной обработки необходимо стремиться к приведению измерений к виду «записей идеального/абстрактного датчика», не обладающих специфическими характеристиками/особенностями присущими конкретной физической реализации измерительного датчика.

В библиотеке GeOMLib, на стадии первичной обработки задействованы две переменные:

RS – массив ячеек, каждая ячейка которого включает запись одной системы оборудования (датчиков, прочитанных из одного файла);

PR – массив ячеек, каждая ячейка которого включает записи всех систем оборудования на профиле съемки (данные из нескольких файлов); при этом, записи обрезаны по границам времени съемки профиля.

В большинстве случаев, (c) и (d) не выполняются для всего времени записи некоторого датчика, а применяются именно ко времени прохождения геофизического профиля. В этом случае, требуемые участки записи «вырезаются» из RS, помещаются в PR{n} (для профиля съемки номер “n”) и после этого обрабатываются по процедурам (c) и (d).

Важной особенностью является то, в процессе дальнейшей обработки используются единые Результаты первичной обработки датчиков для всех аппаратурных комплексов и выполняется это в рамках одного программного инструмента. Например, если при позиционировании ГЛБО будут внесены поправки в установочные углы USBL, то пересчет позиции ГЛБО будет достаточно просто «перенести» и на позицию магнитометра, который буксировался в связке (piggy-back) с ГЛБО. Противоположной является ситуация, когда первичная обработка и предварительная обработки (см. описание ниже), производятся в разных программах по разным алгоритмам (в результате из одного и того же фактического материала могут быть, например, вычислены существенно различающиеся позиции для ГЛБО и буксируемого в связке с ним магнитометра).

(3) Процедуры и результаты предварительной обработки // The Pre-processing Procedures and Pre-processed Set data forming

Процедуры предварительной обработки, направлены на вычисление «распределения геофизического поля во времени и пространстве»; при этом данные множества датчиков, путем расчетов, «сжимаются» до данных вида «координаты-время-значение поля». Обычно, при вычислениях используются только данные измеренные на профилях съемки (помещенные в ячейки переменной PR).

Поскольку «специфические характеристики» датчиков были учтены на предыдущем этапе обработки, то во время предварительной обработки вычисления производятся с абстрактными «результатами измерений», которые характеризуются абстрактной «ошибкой измерений».

Следует отметить, что формулировка «предварительная обработка» относится именно к области геофизики; в процессе предварительной обработки решаются прямая и обратная навигационные задачи, что по отношению к навигации является конечным результатом обработки.

Пример задачи предварительной обработки – вычисление координат приемника-излучателя НСП, закрепленного на корпусе судна, в моменты «отстрела» (присваивание меток со временем и координатами трассам НСП).

Для результатов первичной обработки выполняется следующая последовательность **процедур предварительной обработки:**

- (a) Расчет положения (аттитюды) датчиков геофизического поля на основе навигационных датчиков;
- (b) Интерполяция данных (датчиков геофизики и навигации) к «целевому времени», то есть по времени измерений «основного/целевого» датчика, который регистрирует интересующее физическое поле;
- (c) Поиск и устранение противоречий в модели предварительной обработки. Пример 1: поиск грубых ошибок в показаниях USBL, при помощи сопоставления расстояния до маяка-ответчика с длиной вытравленного кабеля, измеренного про помощи счетчика кабеля. Пример 2: поиск участков, где длина вытравленного кабеля оказывается меньше глубины погружения буксируемого устройства, определяемой по датчику давления (обычно бывает при накоплении ошибки счетчика кабеля);
- (d) Расчет результатов (данных) предварительной обработки, на основе результатов (данных) первичной обработки. При этом, результатом предварительной обработки является «распределение поля», которое содержит: значения измеренного физического поля, время измерения физического поля, координаты и иные характеристики пространственного положения датчика физического поля.
- (e) Выполнение дополнительных вычислений, связанных с моделью геофизического поля и процессов (например, учет поправок за вариации магнитного поля Земли, при магнитной съемке);

В большинстве случаев, на этапе предварительной обработки, решается именно задача позиционирования датчиков целевого геофизического оборудования. Для этого для каждого профиля геофизической съемки (ячейка переменной PR) создается «набор измерений», в котором ко времени целевого датчика, интерполируются значения всех остальных доступных датчиков, и далее, вычисления производятся уже по этому «набору измерений». При наличии нескольких геофизических аппаратных комплексов, может быть создано несколько «наборов измерений», для каждого аппаратного комплекса (например, для комплексов ГЛБО, НСП, магнитометрии). В отдельных случаях, возможно приведение данных полей к единому «квантованию» по времени или по координатам.

(4) Процедуры и результаты обработки геофизических данных // The Processing Procedures and Processed Data forming

Процедуры обработки геофизических данных основываются на математической модели, заданной уравнениями изучаемого физического поля (то есть, с распределением изучаемого поля во времени и пространстве). Задача процедур обработки геофизических данных – получение решения обратной геофизической задачи. Результаты обработки представляют собой распределение/характеристики источников физического поля, или иные величины, получаемые в результате решения обратной геофизической задачи.

Результаты обработки размещаются в ячейках переменной GL.

(5) Процедуры и результаты интерпретации // The Interpretation Procedures and Interpretation Results forming

Процедуры интерпретации (результатов обработки геофизического поля) переводят результаты обработки из области «характеристик источников физического поля» в область знаний инженерной геологии, с соответствующим «истолкованием результатов обработки».

Шаги (3)-(4) и (4)-(5) могут выполняться многократно, для получения наилучшего разрешения в рамках выбранных физико-математической и физико-геологической моделей среды.

Основной областью приложения библиотеки Ge0MLib, являются стадии (1), (2) и (3). Эти стадии характеризуются взаимопроникновением специализации/областей знания навигации и геофизики. Использование «инструментов программирования графов обработки» именно для данных стадий, связано с многовариантностью фактического материала и процедур обработки: применяется множество различных типов датчиков; существует множество вариантов подключения датчиков, организации обмена цифровой информацией между ними, настроек датчиков; существует множество вариантов обработки результатов измерений, полученных с датчиков и взаимного контроля измерений различных датчиков.

Подход «программирование графов обработки» противоположен использованию «статических многоцелевых программ», основанных на «многоуровневых меню» (когда

многообразие методов решения задачи реализуется через увеличение и усложнение вкладок, меню и чек-боксов). Применение «программирования графов» является не только более гибким, но и, в конечном итоге, делает процесс обработки более контролируемым, чем подход «многоуровневых меню».

Кроме того, использование отдельного программного обеспечения «для навигации» и «для геофизики» создает определенные препятствия, связанные с импортом-экспортом данных между программами и со скоростью выполнения цикла «расчет навигации геофизических датчиков – контроль согласованности предварительных результатов геофизической съемки поля по профилям».

4 Уровни абстрактности данных

На основе описанных выше «Стадий получения и обработки данных геофизической съемки» можно выделить несколько уровней связанных с «**абстрагированием**» результатов измерений и расчетов:

- 1) Фактические результаты измерений с датчиков (Sensors Set), являются данными «физически реализованных датчиков»; при этом **датчики характеризуются специфическими характеристиками и особенностями**, которые находят отражение в измеренных данных. На стадии первичной обработки необходимо учесть все «специфические характеристики». Например, для данных альтиметра: удаляются выбросы, вносятся калибровочные коэффициенты (перевод измерений АЦП в высоту), учитывается скорость звука на момент измерений, выполняется сглаживание устраняющее квантование по шкале глубин (в результате которого глубина изменяется скачкообразно), выполняется синхронизация измерений альтиметра ко времени UTC; в результате получают данные «абстрактного/идеального альтиметра».
- 2) Процедуры предварительной обработки, имеют дело с «**данными абстрактных/идеальных датчиков**» (без специфических особенностей и с максимально удаленным влиянием сторонних процессов и помех).
- 3) Данные «идеальных датчиков» используются процедурами предварительной обработки для расчета «**распределения физического поля во времени и пространстве**», которое характеризуется координатной привязкой измерения, меткой времени для измерения, измеренным значением физического поля. При этом производится абстрагирование от «идеальных датчиков» на уровень «распределения поля».
- 4) Процедуры обработки основаны на уравнениях физических полей, которые описывают распределение полей от «источника поля». В результате решения обратной геофизической задачи получают «результаты обработки геофизических данных». В широком смысле, результаты обработки несут информацию о **распределении и характеристиках источников физического поля** (в существовании множества возможных комбинаций источников заключается неопределенность решения обратной геофизической задачи). Примером «источников» может быть положение магнитных масс (при UXO), положение отражающих или преломляющих сейсмических границ (при сейсморазведке). Существуют различные уровни «приближения модели среды» и «глубины обработки» при решении обратной задачи (например, для отраженных сейсмических волн могут быть применены различные типы миграции, может быть учтена анизотропия среды, исследования могут быть поставлены как для выполнения динамического, так и кинематического анализа и т.д.). Уровни «приближения модели среды и глубины обработки» выбираются в зависимости от задач съемки и свойств инженерных/геологических объектов, для изучения которых проводятся исследования.

5) Процедуры интерпретации имеют дело с переводом информации из области знаний о физическом поле (распределении и свойствах «источников поля») в область знаний об инженерных/геологических объектах (**распределении и свойствах вещества разреза**).

5 Формы представления данных («контейнеры»)

Данные, используемые функциями библиотеки GeOMLib, разделены на несколько типов по **форме представления** содержимого данных (content types). Выделены следующие формы представления: Information content (информационные поля); Row-content (ряд измерений); RowM-content (набор рядов); Matrix-content (ряд «серий измерений»); MatrixM-content (набор рядов «серий измерений»).

Графически, особенности «форм представления данных» («контейнеров» для различных типов информационного содержимого) показаны на *Figure 5.1*. Описание форм представления приведено ниже.

Information-content

По смыслу, форма содержит **«информацию-описание»** для характеристик датчиков, режимов измерения, дополнительную информацию к «блокам данных» (сырым данным датчиков, данным первичной или предварительной обработки). Пример Information content: характеристики геодезического эллипсоида, калибровочные коэффициенты датчика глубины, время начала и конца линии съемки. Переменная с Information-content, наиболее часто имеет тип “structure”.

Row-content

Форма содержит **ряд результатов измерений, выполненных вдоль оси времени** (привязанных к «временным отметкам»). Каждый ряд измерений размещается в своем «поле переменной». Несколько полей характеризующих один блок данных, имеют одинаковую длину. Примеры содержимого рядов измерений (Row-content): широта, долгота, температура воды, угол крена, полный вектор магнитного поля. В библиотеке присутствует определенное число предварительно заданных имен полей переменных, для основных видов измерений, которые обрабатываются функциями библиотеки GeOMLib. Для ряда данных, каждое измерение представлено скалярным значением (одно число, это одно измерение одного датчика). Таким способом можно представлять любые серии «скалярных» измерений во времени, например: данные GNSS, данные IMU/MRU, данные датчиков полного вектора магнитного поля или компонент магнитного поля.

RowM-content

Если мы имеем несколько одинаковых и синхронизированных датчиков, то их можно представить в виде матрицы, ряды которой содержат **«набор рядов измерений для нескольких одинаковых датчиков»**, выполненных вдоль оси времени; при этом каждая строка будет соответствовать измерениям отдельного датчика.

Как альтернативный вариант, измерения каждого датчика можно разместить в отдельном поле переменной, со своим именем, но такое размещение представляется избыточным и неудобным для доступа к данным при вычислениях. Примером RowM-content являются измерения, магнитного поля, выполненные при помощи магнитного градиентометра,

оборудованного несколькими синхронизированными и однотипными датчиками магнитного поля (в количестве от двух до семи).

Matrix-content

Данная форма представления данных содержит **ряд измерений, выполненных вдоль оси времени; при этом, каждое измерение представлено «записью временной серии с одного датчика»**, которая, для удобства обработки, записывается в колонку матрицы. В виде Matrix-content представляются данные с датчиков, выдающих на выходе «серию измерений». Пример записей Matrix-content, это сейсмические разрезы (каждое измерение – сейсмическая трасса) или сонограммы ГЛБО (каждое измерение соответствует записи одной посылки ГЛБО). Другие характеристики сейсмических трасс (координаты, отметки времени) содержатся в структурах с Row-content; при этом количество столбцов в Матрице (Matrix-content) и количество элементов в полях со Строками (Row-content) одинаково.

MatrixM-content

Если мы имеем несколько одинаковых датчиков, формирующих данные типа Matrix-content, то приняв номер датчика за третью размерность, получим трехмерный массив данных. Примером таких данных может служить запись с сейсмической многоканальной косы; при этом данные располагаются в трех осях: 1) время отстрела или номер отстрела; 2) время записи сейсмического сигнала с гидрофонов или номер записанного отсчета сейсмического сигнала; 3) номер группы сейсмоприемников, с которых производилась запись.

Другие характеристики сейсмических трасс (координаты, отметки времени) содержатся в структурах с Row-content (например, координаты сейсмического источника) или RowM-content (например, координаты гидрофонов).

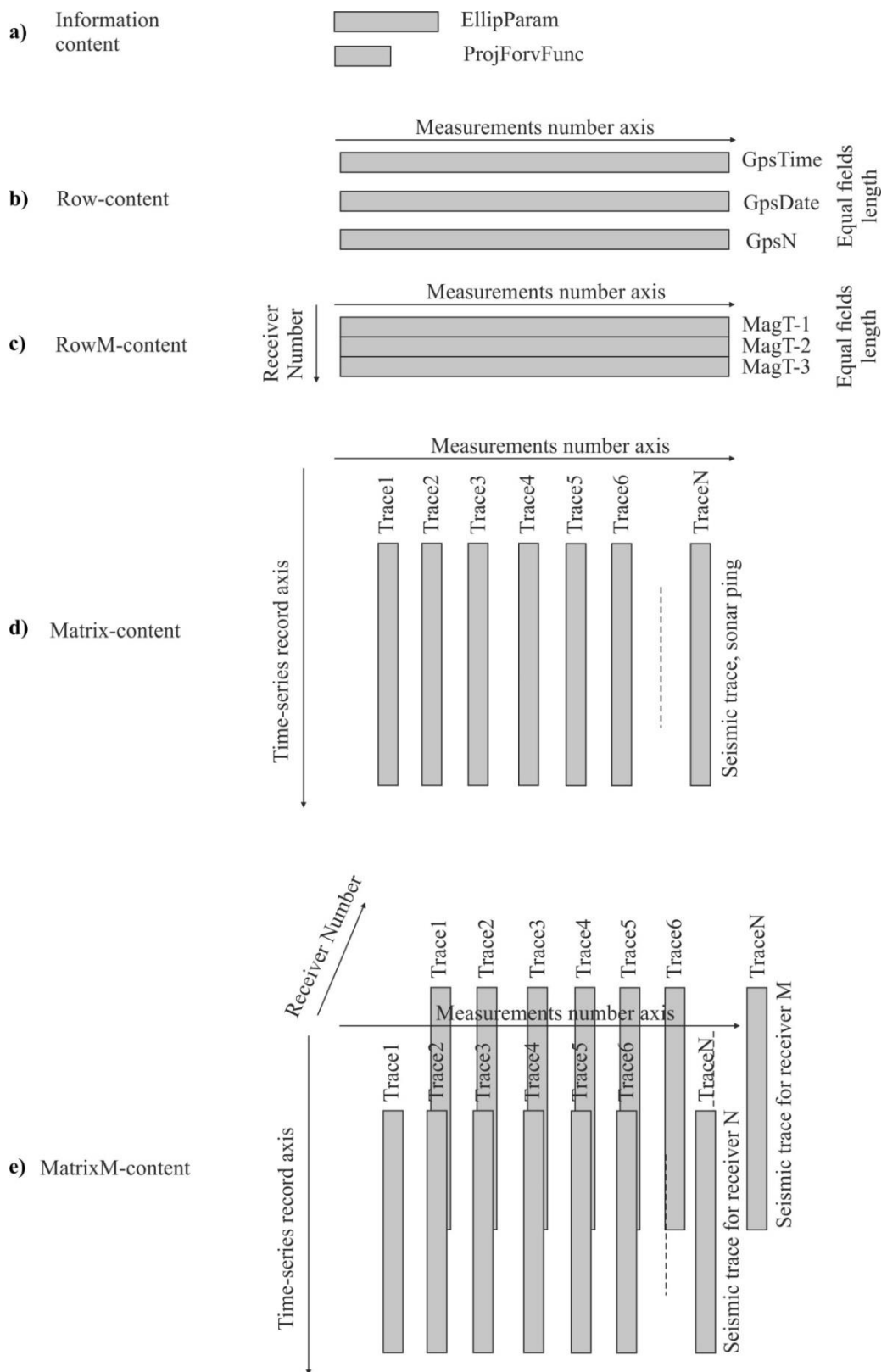


Figure 5.1 Графическое изображение «форм представления» результатов измерений

6 Переменные-структуры и уровни их формализации

6.1 Обзор структур

Данные (как массив чисел) в соответствии с их содержимым и способом измерения, могут быть представлены в виде пяти «форм представления» («контейнеров» для различных типов содержимого): Information content; Row-content; RowM-content; Matrix-content; MatrixM-content (см. *Формы представления данных*). Из отмеченных «контейнеров» формируются более сложные **составные структуры**; такие структуры включают четыре уровня формализации (постепенного усложнения правил формирования структур). Переменные, используемые функциями и скриптами, как входные параметры, также могут быть четырех уровней формализации; также можно говорить об **уровне формализации функций обработки данных** (*Figure 6.1*).

(1) **Произвольное содержимое.** Первый уровень формализации включает данные без предопределенной «смысловой нагрузки», которые можно рассматривать как векторы, скаляры, строки и т.д. Такие данные используются «**универсальными**» функциями (ничего не «знающими» о физической привязке содержимого входных параметров).

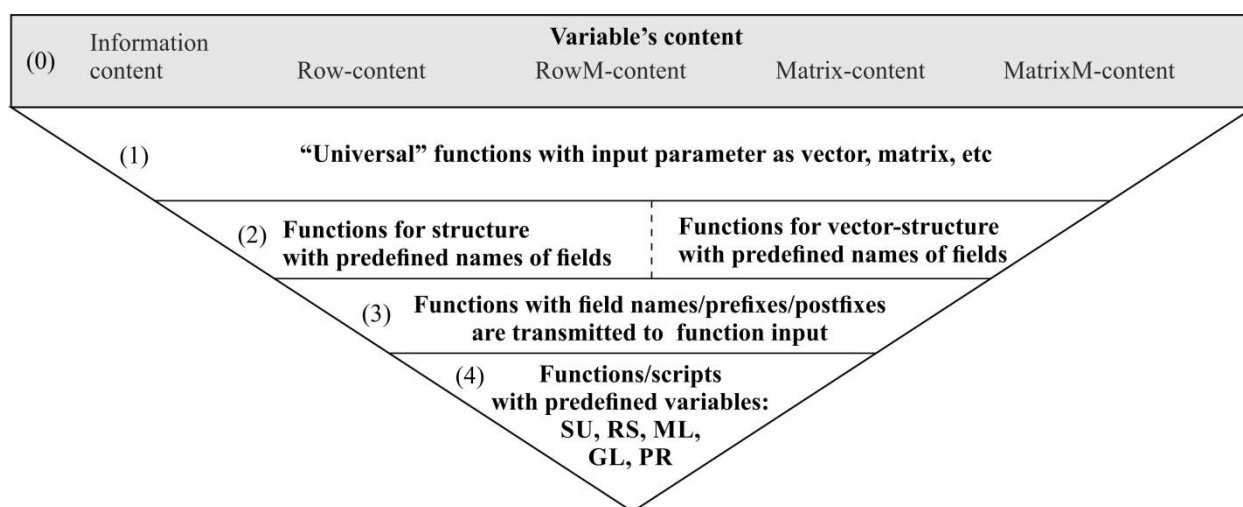


Figure 6.1 Типы функций для работы с составными структурами

(2.1) **Структуры со строго-заданными именами полей.** Второй уровень формализации используется, как правило, при чтении данных из файла; для него характерно размещение данных в полях переменной со «строго-заданными именами». При этом имена полей определяют «смысловую нагрузку данных», которые в этих полях содержатся (координаты, курс, температура и т.д.). Функции, которые используют структуры со строго-заданными именами как входные параметры, «знают» имена полей и «смысловую нагрузку» физических величин, содержащихся в полях; функции выполняют расчеты с этими полями, размещая результаты расчетов в других структурах со «строго-заданными именами». При этом каждая

структура, характеризуется **только одним уровнем полей**, в которых содержатся «контейнеры с данными» (не допускаются поля с несколькими уровнями вложенности). Например, для магнитометра это будут следующие поля: значения модуля полного вектора, signal strength, давление (глубина), альтитуда, время измерения магнитного поля.

(2.2) Вектор-структура со строго-заданными именами полей. Несколько структур (в которых размещены данные одинаковых датчиков), которые содержат одинаковые поля со строго-заданными именами, могут быть объединены в вектор-структуру. При этом каждый элемент вектора содержит структуры с одинаковыми именами полей, в которых расположены одинаковые «контейнеры» с формой представления данных. Например, в вектор-структуру могут быть загружены несколько sgu-файлов (НСП-профилирование). Каждый элемент вектора-структуры допускает только один уровень полей с данными.

(3) Структуры с префиксами и постфиксами в именах полей. Третий уровень формализации подобен второму, но включает две особенности:

- функции обработки могут использовать имена полей в качестве входных параметров;
- существует система префиксов и постфиксов, которые добавляются к строго-заданным именам полей и также могут использоваться как входные параметры функций.

Существуют специальные правила добавления префиксов и постфиксов, которые позволяют формировать разные поля с одинаковой «смысловой нагрузкой». Например, [GpsE, GpsN], являются строго-заданными именами полей для прямоугольных координат, тогда: [TrGpsE, TrGpsN] будут имена полей для точки буксировки заборного оборудования; [UsblGpsE, UsblGpsN] – координаты маяка-ответчика USBL, закрепленного на буксировочном кабеле; [SssGpsE, SssGpsN] – координаты ГЛБО, расположенного в пяти метрах за маяком-ответчиком. Структуры с префиксами и постфиксами в именах полей, в основном используются при обработке данных (первичной и предварительной); используя данный механизм, оператор-обработчик может создать несколько полей с «одинаковой смысловой нагрузкой» и использовать эти поля в функциях графа обработки.

(4) Основные переменные проекта. Четвертый уровень формализации характеризуется наличием переменных с фиксированными именами. К таким переменным относятся:

- SU – структура со строго-заданными полями, содержащая информацию об установках проекта (Information content),
- ML – вектор ячеек с «картографическими слоями»,
- RS – вектор ячеек с фактическим материалом съемки, то есть с сырыми данными датчиков (одна ячейка соответствует одному датчику или аппаратурному комплексу; обычно ячейка содержит запись для всего времени съемки),
- PR – вектор ячеек с профилями съемки (одна ячейка содержит записи нескольких датчиков или аппаратурных комплексов для одного профиля съемки),
- GL – вектор ячеек, используемый для размещения результатов обработки геофизических данных; причем в каждой ячейке размещаются данные одного метода обработки.

При работе с «Основными переменными проекта», используются следующие правила:

- SU содержит структуру, описывающую установки проекта (уровень формализации 2.1).
- Каждая ячейка ML, RS, PR содержит свой объект: измерения одного датчика; измерения датчиков, на одном профиле; картографический объект и т.д.; каждый объект расположен в переменной соответствующей смысловому содержанию объекта.
- Каждая ячейка переменной ML включает один объект с конечным результатом обработки данных, являющихся «внешними данными» по отношению к проекту (уровень формализации 2).
- Каждая ячейка переменной RS содержит сырые данные по одному датчику или аппаратурному комплексу (уровень формализации 2).
- Каждая ячейка переменной PR содержит записи нескольких датчиков (аппаратурных комплексов) по профилю съемки (уровень формализации 3). Таким образом, для переменной PR, мы имеем два уровня вложенности имен полей: первый – имя датчика или аппаратурного комплекса (например, магнитометр); второй – «имя подсистемы датчика» (давление, альтиметр, магнитное поле и т.д.).
- Каждая ячейка переменной GL содержит «Результаты одного метода обработки геофизических данных». Ячейка содержит некоторое число полей, причем уровень вложенности полей определяется методом обработки геофизических данных и описывается в документации к соответствующему методу обработки.

6.2 Структуры со строго-заданными именами полей, постфиксы и префиксы

Сырые результаты изменений датчиков или аппаратурных комплексов, импортируемые в MatLab из файлов, размещаются в структурах со строго-заданными именами полей, для каждого типа данных. Имена полей, принятые для координат и иных данных навигации, показаны в [Table 6.1](#). Строго-заданные имена полей «известны» функциям импорта данных; считанные структуры размещаются в ячейках переменной RS (или PR).

Строго-заданные имена полей могут быть изменены, путем добавления постфиксов или префиксов, при обработке сырых данных функциями обработки данных (первичной или предварительной).

Table 6.1 Пример строго-заданных имен полей с данными навигации

Имя поля	Описание имен полей переменной
GpsLat	Широта (обычно WGS84); DD.DDDDDD.
GpsLon	Долгота (обычно WGS84); DDD.DDDDDD.
GpsAltSea	Высота над уровнем моря по GNSS.
GpsHgtGeoid	Превышение над уровнем геоида по GNSS.
GpsHead	Истинный курс по показаниям дифференциальной GPS/гироскопа/MRU.
GpsE	Прямоугольная проекция, Восток.
GpsN	Прямоугольная проекция, Север.

Имя поля	Описание имен полей переменной
GpsH	Высота, полученная при пересчете координат с эллипсоида на эллипсоид (используется для однозначности обратного пересчета).
GpsKp	Идентификатор точки; может быть номером измерения, КП или иным.
GpsZ	Координаты по вертикальной оси геодезического датума (могут применяться при задания позиции трубопроводов, буксируемых устройств и т.д.).
Altitude	Альтитуда (обычно определяется при помощи альтиметра на буксируемом устройстве).
Depth	Заглубление (обычно определяется датчиком давления на буксируемом устройстве).
Head	Курс на прямоугольной проекции.
WaterDepth	Глубина моря в текущей точке (может быть «измеренной для текущего момента или привязанной к системе отсчета MLS, LAT, Baltic, и т.д.).
<u>XXXXRaw</u>	Постфикс для сырых данных (без применения сглаживания, удаления помех и иных процедур); здесь XXXX – имя какого-либо поля переменной.

Постфикс “Raw”

Для сохранения не обработанных данных датчика используется механизм присвоения именам полей постфикса “Raw”. Поля с постфиксом Raw, автоматически создаются функциями первичной обработки данных. Например, сырые данные из полей именами GpsE и GpsN (прямоугольные координаты), будут сохранены в полях с именами GpsERaw и GpsNRaw, перед тем, как значения в полях GpsE и GpsN подвергнутся процедуре «удаления штормовых выбросов и сглаживанию». При повторном запуске процедуры «удаления штормовых выбросов и сглаживания», для обработки будут использованы сырые данные из полей GpsERaw и GpsNRaw.

Обычно постфикс “Raw” используется только для «сырых данных»; данные датчиков, которые интерполируются ко времени целевого-датчика, не могут сохранить поля с этим постфиксом, в силу изменения размера данных при интерполяции (при этом данные в полях с постфиксом Raw, по смыслу, становятся «обработанными»). Использование постфикса для полей с «результатами расчетов» не является логичным.

В общем случае, описанный механизм «присвоения постфиксов» может использоваться для любой функции: исходные данные, перед обработкой функцией сохраняются в полях, с добавлением к имени специального постфикса; при последующих запусках функции обрабатываются «исходные» (относительно первого запуска) данные. Постфикс “Raw” оказался хорошим решением для «сырых данных», поскольку все сырые данные проходят процедуры первичной обработки, связанные с удалением выбросов, сглаживанием и т.д., а также не имеется данных «предшествующих» сырым.

Префиксы строго-заданных имен полей

Если данные, которые необходимо разместить в полях одной переменной, ранее размещены в полях с одинаковыми строго-заданными именами (при этом данные несут одинаковую смысловую нагрузку), то они могут размещаться в полях с использованием «префикса для имени поля».

Например, пусть координаты точки буксировки оборудования (при загрузке данных располагаются в полях со строго-заданными именами GpsE, GpsN и GpsH) интерполируются ко времени измерения магнитного поля и копируются в переменную содержащую запись магнитометра; тогда, при копировании к имени полей может быть добавлен префикс Tr (TrGpsE, TrGpsN и TrGpsH). По аналогии, для координат буксируемого тела, при копировании может быть использован префикс Tb (TbGpsE, TbGpsN и TbGpsH); для координат нескольких датчиков магнитного поля (размещенных на буксируемом устройстве) может быть использован префикс Ts (TsGpsE, TsGpsN и TsGpsH). Такой механизм добавления префиксов, позволяет включить в поля переменной одного датчика множество полей с одинаковой смысловой нагрузкой (а также, исходно, с одинаковыми именами полей). При добавлении префиксов имена полей сохраняют строго-заданную часть.

6.3 Вектор-структура

Данные с некоторого датчика, загруженные из нескольких файлов в несколько структур со строго-заданными полями, могут быть представлены как вектор-структура. При этом экспортированные данные из каждого файла, размещаются в одном элементе вектора-структуры.

Например, для вектора-структуры A, будем иметь следующий вид записи полей:

```
-- A(1..n).GpsLat;  
-- A(1..n).GpsLon;  
-- A(1..n).GpsAltSea.
```

Вектор-структура обычно применяется при загрузке нескольких файлов с однотипными данными (которые размещаются в полях с одинаковыми именами); также для вектора-структуры обычно написаны функции экспорта данных. Например, мы можем загрузить несколько файлов P1/90 в вектор-структуру, а после этого, экспортировать трек с координатами точки измерения, для каждого файла, в скрипт AutoCad.

Другой пример: мы можем загрузить несколько Sgy-файлов в вектор-структуру. Каждый Sgy-файл импортируется в MatLab в три переменные SgyHead (бинарный, текстовый и расширенный текстовый заголовки), Head (заголовки трасс) and Data (матрица с данными по трассам). При этом будут сформированы два вектора-структуры и один вектор ячеек:

```
-- SgyHead(1..n);  
-- Head(1..n);  
-- Data{1..n}.
```

SgyHead(m) и Head(m) являются элементами вектора-структуры со строго-заданными полями. Data{m} является элементом вектора ячеек; ячейки содержат сейсмограмму с данными для каждой сейсмической трассы или имя tmp-файла с сохраненной сейсмограммой (смотри описание функций gSgy).

Файлы, загружаемые в вектор-структуру, могут быть не связаны с отдельными профилями геофизической съемки (в отличие от вектора ячеек PR). Например, если мы выполняем

непрерывную съемку НСП (при этом запись не останавливается ни на разворотах между линиями съемки, ни на переходах внутри площадки), то мы можем загрузить все Sgy-файлы съемки (обычно программное обеспечение, при превышении объема файла с записываемыми данными, начинает запись в новый файл) в вектор-структуру. Далее, например, из вектора-структуры по времени съемки можно будет «вырезать» данные для каждого профиля съемки и данные, записанные на разворотах и переходах.

По «доступу к данным» вектор-структура является хорошим решением для загрузки данных на этапе «Первичной обработки» (загрузка в ячейку переменной RS).

6.4 Основные переменные с заданными именами

Данные «проекта обработки геофизических данных» размещаются в следующих переменных с заданными именами:

SU – структура с основными настройками проекта;

ML – вектор ячеек с «картографическими слоями» (сторонние «статические» данные, которые не предполагается обрабатывать); например: координаты объектов инфраструктуры на площадке работ, положение береговой линии, координаты устьев скважин, координаты трубопроводов и кабелей, данные батиметрии для разных лет съемки;

RS – вектор ячеек с сырыми данными датчиков (данные одного датчика или аппаратурного комплекса в каждой ячейке);

PR – вектор ячеек с данными датчиков «по профилю съемки» (содержит несколько датчиков для одного профиля; в процессе предварительной обработки данные разных датчиков будут интерполироваться ко времени измерений «целевым датчиком» на каждом профиле);

GL – вектор ячеек с «результатами обработки геофизической съемки» (данные в GL получаются как результат обработки данных из переменной PR); примеры: результаты пикирования профилей НСП (сейсмические горизонты); результаты пикирования магнитных аномалий, оценки положения магнитных масс и их величин.

В основном, переменные SU, PR, RS, ML и GL, не используются как параметры функций для оптимизации использования памяти (при крайней необходимости, можно определить эти переменные как «глобальные», но широко применять данный подход не рекомендуется). В функции, в качестве параметров, передается только содержимое одной ячейки каждой из переменных; при этом части графа обработки желательно выносить в функции выполняя ряд условий (см. [Команды-функции для скрипта](#)).

6.4.1 SU (структура с настройками проекта)

Переменная с «настройками проекта» содержит информацию с общими настройками для всего проекта и настройками общего характера, применяемыми к каждому профилю съемки. Для задания значений полей переменной SU, эти значения прописываются непосредственно в коде

скрипта; графические меню или ini-файлы для этой цели не используются, поскольку усложняют код и уменьшают наглядность задания полей переменной.

Данные, размещенные в переменной SU, используются двумя путями. «Не-изменяемые установки проекта» хранятся в самой переменной; скрипт обработки, для их чтения, обращается к полям переменной. Установки, которые могут меняться от профиля к профилю, копируются из SU в поле «установок профиля» PR{n}.LLog.

Примеры полей с «не-изменяемыми установками проекта» в SU:

```
SU.RootDir='d:\2019\009_Grad'; % корневая папка проекта;
SU.LLog=1; % Номер ячейки RS, где хранится Log-файл и именами линий, временем начала и
           конца съемки линии и иными параметрами;
SU.LPlan=1; % Номер ячейки ML, в которой хранится План линий съемки;
SU.Bott=2; % Номер ячейки ML, в которой хранится Цифровая модель рельефа морского дна;
SU.NavS=struct('EllipParam',[6378137 sqrt(2.* 1/298.257-(1/298.257).^2)],'EllipTransParam',[6.102269
              -10.161 -5.186 -0.050 0.0701 0.0583 2.0293e-
              6], 'EllipForvTransFunc','gNavGeoc2Geoc1032', 'EllipRevTransFunc',
              'gNavGeoc2Geoc1032inv','TargCode',2); % Датум, в котором работают навигационные
           системы при выполнении съемки;
SU.NavP=struct('EllipParam',[6378137 sqrt(2.* 1/298.257-(1/298.257).^2)],'ProjParam',[0 121 0.9999
              250000 0], 'ProjForvFunc', 'gNavGeog2ProjUtm', 'ProjRevFunc', 'gNavProjUtm2Geog',
              'TargCode',6); % датум (прямоугольная проекция) в которой производится обработка
           навигации;
SU.CompTimeLocShift=8*3600; % сдвиг местного времени относительно мирового в секундах;
SU.MagNorm=[44000 -3.60 35.50]; % локально-нормальное магнитное поле [T(nT) D(deg) I(deg)];
SU.SgyFormatForced=1; % принудительный формат экспорта для sgy-файлов (формат трассе); если
           параметр пустой, то файлы записываются в том же формате, в котором были считаны;
SU.SgyReelPos=1:4; % номера символов с «префиксом профиля съемки» в имени профиля съемки;
SU.Job=25; % Идентификатор проекта, предоставляемый заказчиком (JIN).
```

Примеры полей SU, значения которых могут изменяться от профиля к профилю (расчет геометрии при съемке СВР):

```
SU.WaterVelocity=1470; % скорость звука в воде с м/с;
SU.NameP190GunHead='P190_GunHeader.txt';
SU.NameP190Cmp1Head='P190_Cmp1Header.txt';
SU.NameSgyTxt='Sgy_TextHeader.txt';
SU.SpInterval=6.25; % интервал отстрела профиля;
SU.GunTpNod=[]; % позиция точки буксировки сейсмических источников, относительно
           метacentра судна;
SU.GunDist=54; % длина буксировочного кабеля сейсмоисточников от точки буксировки, до
           «акустического центра»;
```

```

SU.StTpNod=[]; % точка буксировки сейсмической косы, относительно метацентра судна;
SU.StBuoyDist=1418.4; SU.StBuoyNum=numel(SU.StBuoyDist); % длина сейсмической косы от
    точки буксировки до концевого буя // количество буев на косе;
SU.StBirdDist=[62.2 162.2 262.2 362.2 462.2 562.2 662.2 762.2 862.2 962.2 1062.2 1162.2 1262.2];
    SU.StBirdNum=numel(SU.StBirdDist); % расстояние от точки буксировки до
    контроллеров глубины («птиц») рассчитанное вдоль длины сейсмической косы //
    количество контроллеров глубины на косе;
SU.StChDist=(0:192-1).*6.25+64.3;SU.StChNum=numel(SU.StChDist); % расстояние от точки
    буксировки до центров групп сейсмических приемников на косе // номер
    сейсмических каналов на косе;
SU.StAuxIdenitafactionCode=[9 3 3 3]; SU.StAuxNum=numel(SU.StAuxIdenitafactionCode); % Байты
    29-30 заголовка Sgy, содержащие TraceIdenitafactionCode для дополнительных каналов
    сейсмостанции; количество дополнительных каналов;
SU.StAllNum=SU.StChNum+SU.StAuxNum; % общее количество каналов на сейсмической косе;
SU.StAllGain=[zeros(size(SU.StChDist))+8 16 16 16 16]; % усиление для сейсмических каналов.

```

Field	Value	Min	Max	Class
RootDir	'd:\UHR_GeoEel\'			char
LPlan	1	1	1	double
NavS	1x1 struct			struct
NavP	1x1 struct			struct
CompTimeLocShift	39600	39600	39600	double
SgyFormatForced	1	1	1	double
SgyReelPos	[1 2 3 4]	1	4	double
Job	25	25	25	double
WaterVelocity	1470	1470	1470	double
NameP190GunHead	'P190_GunHeader.txt'			char
NameP190Cmp1Head	'P190_Cmp1Header.txt'			char
NameSgyTxt	'Sgy_TextHeader.txt'			char
SpInterval	6.2500	6.2500	6.2500	double
GunTpNod	[]			double
GunDist	54	54	54	double
StTpNod	[]			double
StBuoyDist	1.4184e+03	1.4184e+03	1.4184e+03	double
StBuoyNum	1	1	1	double
StBirdDist	1x13 double	62.2000	1.2622e+03	double
StBirdNum	13	13	13	double
StChDist	1x192 double	64.3000	1.2581e+03	double
StChNum	192	192	192	double
StAuxIdenitafactionCode	[9 3 3 3]	3	9	double
StAuxNum	4	4	4	double
StAllNum	196	196	196	double
StAllGain	1x196 double	8	16	double

Figure 6.2 Пример значений полей переменной SU (расчет геометрии для съемки СВР)

6.4.2 ML («картографические данные»)

Переменная ML является вектором ячеек; в каждой ячейке переменной размещается один объект уровня формализации 2, с «картографическими данными». Такой объект содержит внешние данные по отношению к текущему проекту (данные, которые «окончательно обработаны» и для которых не предполагается дополнительной обработки или изменения в рамках проекта). Примеры: координаты объектов инфраструктуры, координаты береговой линии, положение устьев скважин, координаты и иные характеристики трубопроводов или кабелей, данные батиметрии (конечная обработка).

В настоящий момент возможно использование следующих объектов, размещаемых в ячейках ML:

- векторные объекты (такие как структура PolyLine) – план линий съемки, положение объектов и иные маркеры на площадке работ;
- изображения с картографической привязкой (изображение батиметрической съемки, мозаика ГЛБО, изображение Backscatter);
- данные на равномерной сетке (grids), например – данные батиметрии используемые для приведения сейсмического дна к единому уровню.

Чаще всего «картографические данные» используются для отрисовки или для экспорта. На (Figure 6.3) показаны поля объекта, содержащего план линий съемки (узлы поли-линий, типа PolyLine, с координатами линий съемки, размещены в поле LP). Для данного объекта заданы функции отрисовки в графическом окне MatLab и экспорта в скрипт AutoCad. Однако, «картографические данные» могут использоваться и для вычислений; например, такие данные как координаты-заглубление-диаметр трубопровода, могут быть использованы для моделирования гиперболы на разрезе НСП при пересечении трубопровода профилем геофизической съемки (подразумевается профиль съемки, выполняемый в рамках обрабатываемого проекта).

Задание объектов для ячеек переменной ML, и их загрузка, выполняется в коде скрипта обработки (так же как и для переменной SU).

Пример задания полей переменной ML, для загрузки плана линий съемки:

```
ML{SU.LPlan}.fRec='LinePlan20190916.txt'; % имя файла с планом линий съемки, расположенного  
в корневой папке проекта;  
ML{SU.LPlan}.fRecFormat='LinePlan'; % формат объекта, размещенного в ячейке ML;  
ML{SU.LPlan}.NDraw=2; % порядок отрисовки (NaN если объект не предназначен для отрисовки).
```

Пример задания полей переменной ML, для загрузки данных батиметрии:

```
ML{SU.Bott}.fRec='3961_EGS#2.pts'; % имя файла с данными батиметрии, расположенного в  
папке проекта;  
ML{SU.Bott}.fRecFormat='Xyz2Triang'; % формат объекта, размещенного в ячейке ML;  
ML{SU.Bott}.NDraw=NaN; % порядок отрисовки (NaN если объект не предназначен для  
отрисовки).
```

Field	Value	Min	Max	Class
fRec	'LPlan_2DHR.txt'			char
fRecFormat	'LinePlan'			char
NDraw	2	2	2	double
LP	1x74 struct			struct

Figure 6.3 Пример полей ячейки ML{1} (план линий съемки)

6.5 RS (фактические результаты измерений)

Переменная RS является вектором ячеек, который содержит «сырые данные» с датчиков (данные, записанные с датчиков без какой-либо обработки). В каждой ячейке находится информация необходимая для загрузки (импорта) значений из файла, результаты импорта, иная информация необходимая для обработки показаний датчика. Соответственно, для каждого типа датчиков необходимо задание своего собственного набора полей; имена полей, используемые при загрузке «сырых данных» датчиков являются строго-заданными (уровень формализации 2).

Задание датчиков для ячеек переменной RS, и их загрузка, выполняется в коде скрипта обработки (так же как и для переменных SU и ML).

Пример задания полей переменной RS, для загрузки «Журнала оператора», заполненного для магнитометрической съемки (ссылка на журнал содержится в SU.LLog=1):
 RS{SU.LLog}.fRec='Log_mag.txt'; % файл с журналом оператора, расположенный в корневой папке проекта;

RS{SU.LLog}.fRecFormat='gLLog'; % формат датчика;

RS{SU.LLog}.fgLogDelim=[]; % неприменимо.

Пример задания полей переменной RS, для загрузки «координат буксируемого объекта» (загружается в ячейку номер два):

RS{2}.fRec='000_TB\'; % имя папки, которая содержит файлы с координатами буксируемого объекта (расположена в корневой папке проекта);

RS{2}.fRecFormat='gLogGPGGA'; % формат датчика;

RS{2}.fgLogDelim=['~' '']; % разделители для времени компьютера, используемые при записи потока данных при помощи утилиты gLog;

RS{2}.Period=1; % период прихода сообщений с позицией (NaN, если не применимо).

Пример задания полей переменной RS, для загрузки «данных градиентометра SeaSpy2» (загружается в ячейку номер семь):

RS{7}.fRec='000_Mag3\'; % имя папки, которая содержит файлы с данными градиентометра (расположена в корневой папке проекта);

```

RS{7}.fRecFormat='gLogSeaSpyGrad'; % формат датчика (примеры форматов других
магнитометров/градиентометров: MagLogG88Magy, MagLogG88Grad, gLogSeaSpyMagy,
gLogSeaSpyGrad);
RS{7}.fgLogDelim=['~' '']; % разделители для времени компьютера, используемые при записи
потока данных при помощи утилиты gLog;
RS{7}.Period=0.5; % период измерений, выполняемых градиентометром;
RS{7}.ID=[999 999 999;999 nan 999]; % серийные номера магнитных датчиков, альтиметров,
датчиков глубины [magnetic_sensor1_SN, altimeter1_SN, depth_sensor1_SN; ....;
magnetic_sensorN_SN, altimeterN_SN, depth_sensorN_SN]; использование серийных номеров
преследует две цели: (1) идентификация используемого оборудования; (2) определение
датчиков, которые установлены в текущей конфигурации оборудования (nan – означает, что
датчик не установлен);
RS{7}.Coef=[1 0 1 0;1 0 1 0]; % линейные коэффициенты (сдвиг шкалы и усиление) для перевода
единиц АЦП в значения альтитуды и глубины, для каждого альтиметра и датчика глубины
[Altimeter1Scale Altimeter1Bias Depth_sensor1Scale Depth_sensor1Bias; ....; AltimeterNScale
AltimeterNBias Depth_sensorNScale Depth_sensorNBias].

```

6.5.1 PR (записи датчиков по профилю съемки)

Переменная RS является вектором ячеек, каждая ячейка содержит записи датчиков, обрезанные по времени съемки отдельного профиля съемки. Кроме того, ячейка содержит запись «целевого геофизического датчика», ко времени которого, при предварительной обработке, интерполируются значения всех иных датчиков. Пример полей ячейки, включающей несколько датчиков, показан на (*Figure 6.4*).

Для каждой ячейки мы имеем два уровня вложенности полей: на первом уровне располагаются имена датчиков или аппаратурных комплексов (например «магнитометр»), а на втором – записи измерений с датчика и подсистем датчика (для магнитометра – глубина, альтитуда, магнитное поле и т.д.).

Каждая ячейка содержит поле “LLog” с общей информацией о профиле съемки (*Figure 6.5*). Данная информация включает два вида данных:

(1) поля с информацией о самом профиле съемки (получаемой, например, из Журнала оператора):

- LLog.LName – имя профиля,
- LHead – направление прохождения запланированной линии профиля,
- CompDay, CompTime – отметки времени по часам компьютера для начала и конца профиля (обычно используется с утилитой gLog),
- GpsDay, GpsTime – отметки времени по часам GNSS для начала и конца профиля (используется для записей датчиков, включающих время GNSS),

(2) поля, скопированные из переменной SU как «изменяемые характеристики для профиля съемки».

Field	Value	Min	Max	Class
LLog	1x1 struct			struct
ProcLog	1x45 char			char
Bathy	1x1 struct			struct
StTp	1x1 struct			struct
StBuoy	1x1 struct			struct
StNav	1x1 struct			struct
StDpt	1x1 struct			struct
StTens	1x1 struct			struct
GunTp	1x1 struct			struct
GunLog	1x1 struct			struct
BirdLog	1x1 struct			struct
SgyHead	1x1 struct			struct
Head	1x1 struct			struct

Figure 6.4 Пример полей PR{1} (расчет геометрии сейсмического источника)

Field	Value	Min	Max	Class
WaterVelocity	1470	1470	1470	double
NameP190GunHead	'P190_GunHeader.txt'			char
NameP190Cmp1Head	'P190_Cmp1Header.txt'			char
NameSgyTxt	'Sgy_TextHeader.txt'			char
SpInterval	6.2500	6.2500	6.2500	double
GunTpNod	[]			double
GunDist	54	54	54	double
StTpNod	[]			double
StBuoyDist	1.4184e+03	1.4184e+03	1.4184e+03	double
StBuoyNum	1	1	1	double
StBirdDist	1x13 double	62.2000	1.2622e+03	double
StBirdNum	13	13	13	double
StChDist	1x192 double	64.3000	1.2581e+03	double
StChNum	192	192	192	double
StAuxIdenitificationCode	[9 3 3 3]	3	9	double
StAuxNum	4	4	4	double
StAllNum	196	196	196	double
StAllGain	1x196 double	8	16	double
LName	'0006_C_L_HR_29'			char
LHead	26	26	26	double
ShotNum	354	354	354	double
CompDay	[NaN NaN]	NaN	NaN	double
CompTime	[NaN NaN]	NaN	NaN	double
GpsDay	[737585 737585]	737585	737585	double
GpsTime	[7.3648e+04 7.4720e+04]	7.3648e+04	7.4720e+04	double

Figure 6.5 Пример полей PR{1}.LLog (расчет геометрии сейсмического источника)

6.5.2 GL (результаты обработки геофизической съемки)

Переменная GL является вектор ячеек; ячейки содержат «результаты обработки геофизических данных», которые создаются процедурами обработки, при обработке данных переменной PR. Пример результатов обработки: результаты пикирования сейсмограмм НСП (сейсмические горизонты), результаты пикирования магнитных аномалий и оценки расположения и магнитной массы источников аномалий.

Каждая ячейка содержит поля, название и степень вложенности которых, определены в соответствии с описанием (документацией) к конкретному методу обработки.

7 Скрипты с графами обработки

7.1 Команды скриптов

Для удобства использования и «контроля» графы обработки реализуются в MatLab как скрипты (данный подход, также позволяет прилагать к обработанным данным граф в виде одного файла). Графический интерфейс, для ввода переменных SU, ML и RS не используется (например, меню и текстовые формы); значение полей определяется непосредственно в коде скрипта.

Библиотека GeOMlib использует **специальный формат команд для скриптов**:

```
{'ComName',Par1,Par2,...,ParN,ParDef1,...ParDefN};ScriptName;
```

где

{....} – данные, передаваемые в скрипт (имя команды скрипта и параметры скрипта);

ComName – имя команды скрипта;

Par1..ParN – обязательные параметры для выполнения команды;

ParDef1..ParDefN – дополнительные параметры для выполнения команды; если данные параметры пропущены, то устанавливаются «параметры по умолчанию» прописанные в коде скрипта (или коде команды-функции для скрипта);

ScriptName – имя скрипта, который необходимо выполнить.

Пример команды на запуск скрипта: {'PR_Set','0006_C_L_HR_29',26,1};gUhrGeoEelGeomR02;

Код скрипта работает с переменными SU, PR, RS, ML и GL. Другие переменные, расположенные в основной области MatLab, могут быть использованы как параметры команды, передаваемой в скрипт.

Предпочтительно использовать в теле скрипта переменные, имена которых заканчиваются символом «_»; таким образом, переменные, заданные вне скрипта, не будут изменены или удалены при его работе. Дополнительные переменные, используемые при расчетах в теле скрипта, рекомендуется удалять в последней линии команды скрипта.

Скрипт для решения задачи «Подготовка распределения физического поля» (“Field Distribution creation”; *Figure 3.1*) импортирует данные из файлов с записями датчиков и формирует Предварительно-обработанные данные. Для этого, в теле скрипта выполняются следующие шаги, реализованные как одна или несколько команд:

- (1) Задание и загрузка полей переменной SU – установка параметров проекта в коде скрипта;
- (2) Загрузка данных в ячейки ML – загружаются «внешние данные» не требующие обработки в теле скрипта. Данный код является формализованным, с известными объектами, форматами и функциями.
- (3) Загрузка «сырых данных» датчиков в ячейки переменной RS (или PR) – показания каждого датчика (или аппаратурного комплекса) загружается в одну ячейку переменной RS, при этом параметры и функция загрузки прописываются в коде скрипта.

Данный код является формализованным с известными объектами, форматами и функциями.

Ячейки RS используются, если имеется запись датчиков не привязанная к началу и окончанию определенных профилей съемки (например, непрерывная запись датчиков на все время съемки); тогда, на одном из следующих шагов записи датчиков будут «обрезаны» ко времени начала и конца профилей и помещены в ячейки переменной PR. Если запись показаний с датчиков осуществлялась только во время съемки линий, тогда эти показания могут быть сразу загружены в ячейки переменной PR (без использования RS).

- (4) «Вырезание» данных и помещение в ячейки переменной PR – используя время начала и конца съемки профиля, вырезаются фрагменты записи датчиков и помещаются в ячейки переменной PR; каждая ячейка переменной соответствует времени регистрации данных на профиле съемки. При задании имени сенсора (имени поля) в ячейке PR, оператор-обработчик может использовать «общепринятое» имя, или придумать имя самостоятельно. Данный код является формализованным, с известными объектами, форматами и функциями, за исключением задания «имен датчиков».
- (5) Первичный контроль качества и обработка измерений с датчиков – обычно первичный контроль включает: контроль периода измерений (для датчиков, которые производят измерения с определенной частотой), удаление выбросов в данных, выявление и удаление трендов «иной природы» (например, рысканье курса судна может приводить к синхронным «осцилляциям» на данных позиционирования буксируемого устройства по USBL), фильтрация и осреднения для высокочастотного случайного шума. Обычно, код «первичного контроля и обработки» является формализованным, с известными объектами, форматами и функциями. На данной стадии, сырые данные датчиков сохраняются в полях с добавлением к именам постфикса “Row”; вместо сырых данных, в соответствующие поля, записываются данные прошедшие процедуры первичной обработки. Также для каждого сенсора создается QC-флаг с результатами маркирования измерений при контроле качества. Основной задачей на данном шаге является переход от «данных физической реализации датчика со специфическими особенностями» к «данным идеального/абстрактного датчика» (см. *Уровни абстрактности данных*).
- (6) Интерполяция первично-обработанных измерений датчиков ко времени «целевого датчика» – на данном шаге данные всех датчиков интерполируются ко времени датчика с записью целевого геофизического поля. В результате получается набор данных с синхронными «измерениями» различных датчиков. Данный код является формализованным, с известными объектами, форматами и функциями.

- (7) Формирование набора предварительно-обработанных данных – на данном шаге производятся расчеты с использованием значений всех синхронизированных датчиков, для формирования «Результатов предварительной обработки» (см. *Уровни абстрактности данных*). В общем случае, результатом предварительной обработки является распределение физического поля – значения физического поля, имеющие координатную привязку и привязку по времени. Пример 1: для измерений магнитного градиентометра создаем «данные с распределением поля», которые содержат XYZ-координаты каждого датчика магнитного поля, время измерения и значение измеренного магнитного поля (а также специфические для метода величины: информация о качестве замера магнитного поля signal strength и информация о «свойствах окружающей среды», например, заглубление и альтитуда датчика магнитного поля, значение вариаций магнитного поля Земли и т.д.). Пример 2: для многоканальной сейсморазведки создаем «данные с распределением поля», которые включают XYZ-координаты источников и приемников на момент выстрела, время выстрела, записи сейсмического сигнала для каждого приемника (а также специфические для метода величины: ошибка синхронизации и характеристика энергии выстрела для каждого источника, записи гидрофонов ближней и дальней зоны, заглубление источника и приемников и т.д.). **Данный код должен быть написан и формализован для каждого нового аппаратного комплекса**; для «регулярной съемки» код является формализованным, с известными объектами, форматами и функциями.
- (7) Экспорт предварительно-обработанных данных и расчет оценок качества – предварительно-обработанные данные экспортируются в файлы специального формата, для дальнейшего использования в программах обработки геофизических данных (выполнения процедур обработки геофизического поля и интерпретации; см. *Уровни абстрактности данных*). Значения оценки качества и графики с оценками качества данных, подготавливаются как часть описания предварительно-обработанных данных. Пример экспортируемых данных: файлы P1/90 с координатами, Sgy-файлы с подгруженными навигационными данными, база данных с «результатами магнитной съемки» для загрузки в Oasis Montaj. **Данный код должен быть откорректирован для каждой новой «задачи экспорта»**; для «регулярной съемки» код является формализованным, с известными объектами, форматами и функциями.
- (8) Загрузка и сохранение переменных SU, RS, PR – данный шаг необходим для загрузки и сохранения результатов работы скрипта.

К описанным основным шагам, при обработке данных, могут быть добавлены дополнительные шаги. Например, на основе высокочастотных записей с датчиков, размещенных в переменной RS{...}, могут быть рассчитаны показания дополнительного «виртуального» датчика,

которые будут использованы на последующих этапах. Тогда, процедуры «Первичный контроль качества и обработка измерений» могут быть применены непосредственно к записям в ячейках RS (а не PR, как в описанном выше графе). Однако, в большинстве случаев, могут быть использованы шаги «типового скрипта», приведенные выше.

7.2 Пример MatLab-сессии с расчетом геометрии, при съемке СВР

Ниже приведен пример команд, для расчета геометрии сейсмической косы при съемке СВР (текст является частью “quick start guide for gUhrGeoEelGeomR02 script commands”):

```
{'SU_Set'};gUhrGeoEelGeomR02;
```

Загрузка «основных настроек проекта».

```
{'ML_Set'};gUhrGeoEelGeomR02;
```

Задание настроек и загрузка «картографических слоев». Для данного скрипта загружается только «План линий съемки».

```
{'Main_Load'};gUhrGeoEelGeomR02;
```

Загружается переменная PS с результатами предыдущей работы скрипта (или создается вектор пустых ячеек PR, указанной длины).

```
{'PR_Set','0006_C_L_HR_29',26,3};gUhrGeoEelGeomR02;
```

Загружаются записи датчиков для профиля съемки. При этом, задается имя линии 0006_C_L_HR_29; направление прохода профиля 26 градусов; данные загружаются в ячейка номер 3.

Загружаются следующие логи и файлы с заголовками:

- 1) положение точки буксировки сейсмической косы с периодом 1 секунда;
- 2) положение точки буксировки сейсмического кластера с периодом 1 секунда;
- 3) положение концевой буя с периодом 1 секунда;
- 4) данные контроллеров глубины («птичек»; используется лог, экспортированный из Qinsy);
- 5) три текстовых лога сейсмической станции GeoEel (Nav for shots log, Depth log, Tension log);
- 6) текстовый лог GunLink (содержит время выстрела, заглубление кластера, атмосферное давление, давление в пневматической трассе источников, ошибки времени синхронизации сейсмических источников и т.д.);
- 7) «пояс» с данными, снятыми многолучевым эхолотом, при прохождении профиля;
- 8) Sgy-файл, сформированный сейсмической станцией GeoEel;
- 9) текстовый файл с шаблоном заголовка для P1/90;
- 10) текстовый файл с шаблоном для текстового заголовка файла Sgy.

```
{'Fix&Shot_QC',3};gUhrGeoEelGeomR02;
```

Визуальный контроль пропущенных пикетов отстрела. Выполняется для профиля, загруженного в ячейку №3 переменной PR.

{'Time_QC','StTp','Gps',1,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание времени для поля 'StTp' (точка буксировки сейсмической косы). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 1 (нет сглаживания). Команда скрипта выполняется для ячеек с 1 по 3.

{'Time_QC','StBuoy','Gps',1,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание времени для поля 'StBuoy' (положение концевого буя сейсмической косы). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 1 (нет сглаживания). Команда скрипта выполняется для ячеек с 1 по 3.

{'Time_QC','GunTp','Gps',1,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание времени для поля 'GunTp' (точка буксировки сейсмического кластера). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 1 (нет сглаживания). Команда скрипта выполняется для ячеек с 1 по 3.

{'Coord_QC','StTp','Gps',10,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание координат для поля 'StTp' (точка буксировки сейсмической косы). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 10. Команда скрипта выполняется для ячеек с 1 по 3.

{'Coord_QC','StBuoy','Gps',10,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание координат для поля 'StBuoy' (положение концевого буя сейсмической косы). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 10. Команда скрипта выполняется для ячеек с 1 по 3.

{'Coord_QC','GunTp','Gps',10,1:3};gUhrGeoEelGeomR02;

Удаление выбросов и сглаживание координат для поля 'GunTp' (точка буксировки сейсмического кластера). Постфикс поля со временем 'Gps' (обычно, может быть 'Gps' или 'Comp'). Окно сглаживания 10. Команда скрипта выполняется для ячеек с 1 по 3.

{'PR_CreateSet',1:3};gUhrGeoEelGeomR02;

Создание набора данных, в котором значения всех датчиков синхронизированы к моментам отстрела. Команда скрипта выполняется для ячеек с 1 по 3.

{'OutPromaxGeom',1:3};gUhrGeoEelGeomR02;

Создание файла с «таблицей» для загрузки в ProMax (таблица с «геометрией сейсмического источника»). Команда скрипта выполняется для ячеек с 1 по 3.

{'OutP190Geom','Gun',2018,1:3,10,'S'};gUhrGeoEelGeomR02;

Создание файлов P1/90, скриптов AutoCad и файла «каталог координат» для позиции сейсмического источника. Год 2018; команда скрипта выполняется для ячеек с 1 по 3; шаг каталога координат 10; идентификатор записи в файле P1/90 'S' (источник).

{'OutP190Geom','Cmp1',2018,1:3,10,'C'};gUhrGeoEelGeomR02;

Создание файлов P1/90, скриптов AutoCad и файла «каталог координат» для позиции ОСТ1. Год 2018; команда скрипта выполняется для ячеек с 1 по 3; шаг каталога координат 10; идентификатор записи в файле P1/90 'C' (общая средняя точка).

```
{'OutSgyGeom',1:3};gUhrGeoEelGeomR02;
```

Экспорт файла Sgy с добавленной навигацией. Команда скрипта выполняется для ячеек с 1 по 3 (записывается три файла)

```
{'Main_Save'};gUhrGeoEelGeomR02;
```

Сохранение переменной PR.

7.3 Журнал обработки профилей съемки

Для сохранения команд, которые были отправлены скрипту при обработке для каждого профиля, используется поле с именем “ProcLog”. Поле PR{n}.ProcLog содержит все команды, которые были отправлены скрипту для обработки профиля “n”. Журнал представляет собой набор строк, с разделителем строк LF (0Aхh).

Пример содержимого PR{n}.ProcLog показан на *Figure 7.1*. Журнал содержит сессию, подобную описанной выше (*Пример MatLab-сессии с расчетом геометрии, при съемке*).

```
>> PR(1, 1).ProcLog
ans =
PR_Get# LName=0006_C_L_HR_29; LHead=26
FixNum#ShotsNumQC#
Time_QC# fmm=StTp; fnpt=Gps; TimeSmooth=1; nn=1; BitTQH=1; f1TQH=0
Coord_QC# fmm=StTp; fnpt=Gps; PosSmooth=1; BitCQH=2; f1TQH=0
Time_QC# fmm=StBuoy; fnpt=Gps; TimeSmooth=1; nn=1; BitTQH=1; f1TQH=0
Coord_QC# fmm=StBuoy; fnpt=Gps; PosSmooth=1; BitCQH=2; f1TQH=0
Time_QC# fmm=GunTp; fnpt=Gps; TimeSmooth=1; nn=1; BitTQH=1; f1TQH=0
Coord_QC# fmm=GunTp; fnpt=Gps; PosSmooth=1; BitCQH=2; f1TQH=0
PR_CreateSet#
OutPromaxGunGeom# SrcPattern0=1; Static0=0
OutP190Geom# CoordinateFieldPrefix=Gun; P190HeadYear=2018; CatStep=10; RecordId=5; Spare1= ; VesselId=1; SourceId=1; OtherId=1; Spare2= ; WCode=NNMM
OutP190Geom# CoordinateFieldPrefix=Cmp1; P190HeadYear=2018; CatStep=10; RecordId=C; Spare1= ; VesselId=1; SourceId=1; OtherId=1; Spare2= ; WCode=NNMM
OutSgyGeom#
```

Figure 7.1 Пример содержимого журнала PR{n}.ProcLog (расчет геометрии съемки СВР)

Код, формирующий строки журнала обработки PR{n}.ProcLog, должен быть добавлен, в скрипт в конце кода команды, перед удалением использованных дополнительных переменных.

Журнал обработки не является инструментом «полного контроля за процессом обработки». В нем не фиксируются манипуляции с данными, которые могут быть совершены в командном окне MatLab (они записываются в журнале сессии MatLab) или в теле скрипта, при изменении его кода (данные операции не фиксируются нигде). Журнал оператора является полезным инструментом для «запоминания» стадии обработки профиля и параметров, переданных командам скрипта. Команды скрипту, не связанные с обработкой профилей (ячеек переменной PR) не записываются в PR{n}.ProcLog.

7.4 Команды-функции для скрипта

Некоторые команды скрипта реализуются при помощи кода, универсального для разных датчиков, примером такой команды является “TimeQC” (контроль качества поля со временем). Такие участки кода перемещаются в так называемые «команды-функции» – код данных функций написан таким образом, что может быть легко скопирован назад в тело скрипта. Это достигается использованием специфических переменных, имен переменных и «специфическим стилем написания кода функций».

Реализация команд-функций позволяет, с одной стороны, упростить код скрипта и уменьшить количество строк в коде, с другой стороны, если мы нуждаемся в «нестандартной обработке» в рамках команды-функции, то мы можем поместить ее код назад в тело скрипта и изменить его под текущие задачи.

Литература