

Российская академия наук
Институт прикладной астрономии

Сообщения ИПА РАН

№ 143

И. Ф. Суркис

**Обработка РСДБ-наблюдений:
Программный пакет QUASAR.
III. Структура и схема функционирования**

Санкт-Петербург
2002

УДК 521.8

И. Ф. Суркис. Обработка РСДБ-наблюдений: Программный пакет QUASAR. III. Структура и схема функционирования.

Ключевые слова: РСДБ-данные, структура пакета QUASAR, состав оцениваемых параметров, многофункциональность, формат файлов.

Описывается структура и схема функционирования нового программного пакета массовой обработки РСДБ-наблюдений на глобальных сетях станций QUASAR, приводятся форматы входных, выходных и внутренних файлов.

I. F. Surkis. Processing of VLBI Observations: QUASAR Software. III. The Structure and Functioning Scheme.

Keywords: VLBI data, QUASAR software structure, estimating parameters set, multipurpose package, files formats.

The structure and functioning scheme of a new QUASAR software for processing of VLBI observations at the global networks are given. Also the formats of the input, output and internal files are presented.

Сообщения Института прикладной астрономии РАН № 143 – Санкт-Петербург, 2002. – 63 с.

Содержание

1. Структура и общая схема пакета	5
2. База исходных данных	11
2.1. Формат QDB файлов	14
2.2. Библиотека работы с QDB-форматом для C++	16
2.3. Библиотека работы с QDB-форматом для FORTRAN	17
2.4. Формат файлов коррекций	18
2.5. Формат файлов наблюдений	18
2.6. Система пополнения базы исходных данных новыми сериями	21
3. Программа average	21
4. Программа reduct	21
4.1. Блок редуцированных вычислений	23
4.2. Блок чтения каталогов	25
4.3. Блок чтения файлов коррекций и наблюдений	26
4.4. Блок чтения эфемерид	26
4.5. Блок записи результатов в файл редукиций	26
4.6. Управляющий блок	27
5. Системы оценивания	27
5.1. Системы сортировки и печати для оценивания методами СКК и МНК	28
5.2. Программа FormS построения и обращения матриц ковариаций	30
5.3. Односерийное оценивание методом СКК	31
5.4. Оценивание методом многопараметрического МНК	34
5.5. Многосерийное оценивание	34

5.6. Системы сортировки и печати на языке C++ для оценивания методами фильтрации Калмана и почасового оценивания	36
6. Дополнительные программы	38
6.1. Программа covarmod	38
6.2. Программа cvrt	38
6.3. Программа output	38
7. Графическая программа чистки	39
8. Управляющая программа	39
9. Утилита SetSeriesPath	41
10. Формат управляющих файлов	41
10.1. Файл path.dat	43
10.2. Файл task.dat	43
10.3. Файл reduct.dat	44
10.4. Файл statsour.dat	44
10.5. Файл der.dat	45
10.6. Файл signals.dat	45
10.7. Файл regular.dat	46
10.8. Файл parunits.dat	46
10.9. Файл extract.dat	46
10.10. Файл covarmod.dat	47
11. Формат файлов эфемерид	47
12. Формат файлов каталогов	48
13. Формат выходных файлов	50
14. Формат временных файлов	56
14.1. Двоичные файлы	58
14.2. Текстовые промежуточные файлы	58
Список литературы	63

1. Структура и общая схема пакета

Пакет предназначен для работы на персональных компьютерах под управлением операционной системы Windows-95. Он включает в себя набор вычислительных программ, базу исходных данных с системой ее пополнения из файлов Mark3-DBH или NGS форматов, систему графической чистки данных, управляющую программу, а также задаваемые а priori эфемериды, каталоги координат и некоторых других данных станций и источников. Пакет обеспечивает высокую гибкость при работе с различными моделями данных и различным набором постоянных параметров и стохастических сигналов. Гибкость заложена в конструкцию всех частей пакета. Параметры делятся на три группы: а) локальные параметры, относящиеся к каждой станции; б) параметры источников, относящиеся к каждому источнику; и в) глобальные параметры, относящиеся ко всей системе в целом. Каждый параметр может иметь три компонента: а) общий, постоянный в течение всего обрабатываемого массива наблюдений; б) суточный, постоянный в течение одной серии наблюдений и в) внутрисуточный (сигнальный), меняющийся от наблюдения к наблюдению. Общий и суточный компоненты представляются как тренды в виде разложений по полиномам Лежандра до 99 степени включительно, а сигнальные — как случайные последовательности, заданные для всех моментов наблюдений. Пакет построен таким образом, что алгоритмически можно задавать оценивание любого набора параметров с любым сочетанием компонентов. Для идентификации параметров вводится их 8-символьный идентификатор. Таким образом, каждый параметр полностью описывается его типом и идентификатором.

Система редукиций генерирует частные производные для некоторого набора параметров. Все эти параметры могут в дальнейшем оцениваться. При необходимости введения нового параметра редукиции дополняются лишь новым блоком, собственно вычисляющим частные производные, и в генерируемый список параметров добавляется новый параметр с указанием его типа. Перед оцениванием пользователь задает идентификаторы и степени трендов оцениваемых параметров; системы оценивания по этим идентификаторам формируют матрицы частных производных для общих, суточных и внутрисуточных компонентов параметров; после вычислений в выходных файлах автоматически ставятся имена параметров, их типы и имена соответствующих станций или

источников. В дальнейшем для краткости компоненты параметров также будут именоваться параметрами. Например, "суточные компоненты параметров" будет сокращаться до "суточные параметры". Схема потока данных представлена на рис. 1.

В качестве пакета массовой обработки наблюдений, программный пакет "QUASAR" имеет собственную базу данных исходных наблюдений (в дальнейшем именуется базой исходных данных). Для удобства переноса информации база разбита на две части: собственно наблюдения и так называемые коррекции, хранящие информацию о поправках к шкалам времени станций и перечень плохих наблюдений; каждая серия наблюдений хранится в отдельном файле. Таким образом, серия представляется одним файлом наблюдений и одним файлом коррекций. Файл наблюдений хранит всю информацию, полученную из файла NGS или Mark3-DBH формата; он не меняется в процессе работы. Такое разделение на изменяемую и неизменяемую части базы исходных данных позволяет: хранить на предназначенном только для чтения устройстве (например, CD-диске) объемные файлы наблюдений; вернуться к исходным данным наблюдений, очистив соответствующий файл коррекций; при переносе исходных наблюдений с компьютера на компьютер копировать только небольшие по объему файлы коррекций.

Система вычислений средних значений температур, давлений и влажностей станций **average** выбирает из базы исходных данных значения этих величин для каждого наблюдения, формирует средние арифметические значения и заносит в файл **average.txt**. Эти значения могут (в зависимости от настроек) использоваться редукциями в качестве опорных значений.

Система редукций, считывая данные из базы исходных данных, производит редукционные вычисления, формируя для каждой обрабатываемой серии наблюдений т.н. файл редукций, содержащий вектор разностей наблюдаемых и теоретических задержек (вектор **Loc**) и матрицу частных производных. В процессе редукционных вычислений возможно формирование т.н. файлов дополнительной информации. Эти файлы содержат данные, предназначенные для внешнего потребителя и не используемые пакетом. Состав информации задается вмешательством в программу редукций **reduct.for**. Файлы дополнительной информации имеют имя серии и расширение **add** и записываются в директорий выходных файлов.

Различные методы оценивания считывают файлы редукций, производят обработку данных и формируют выходные файлы (см.

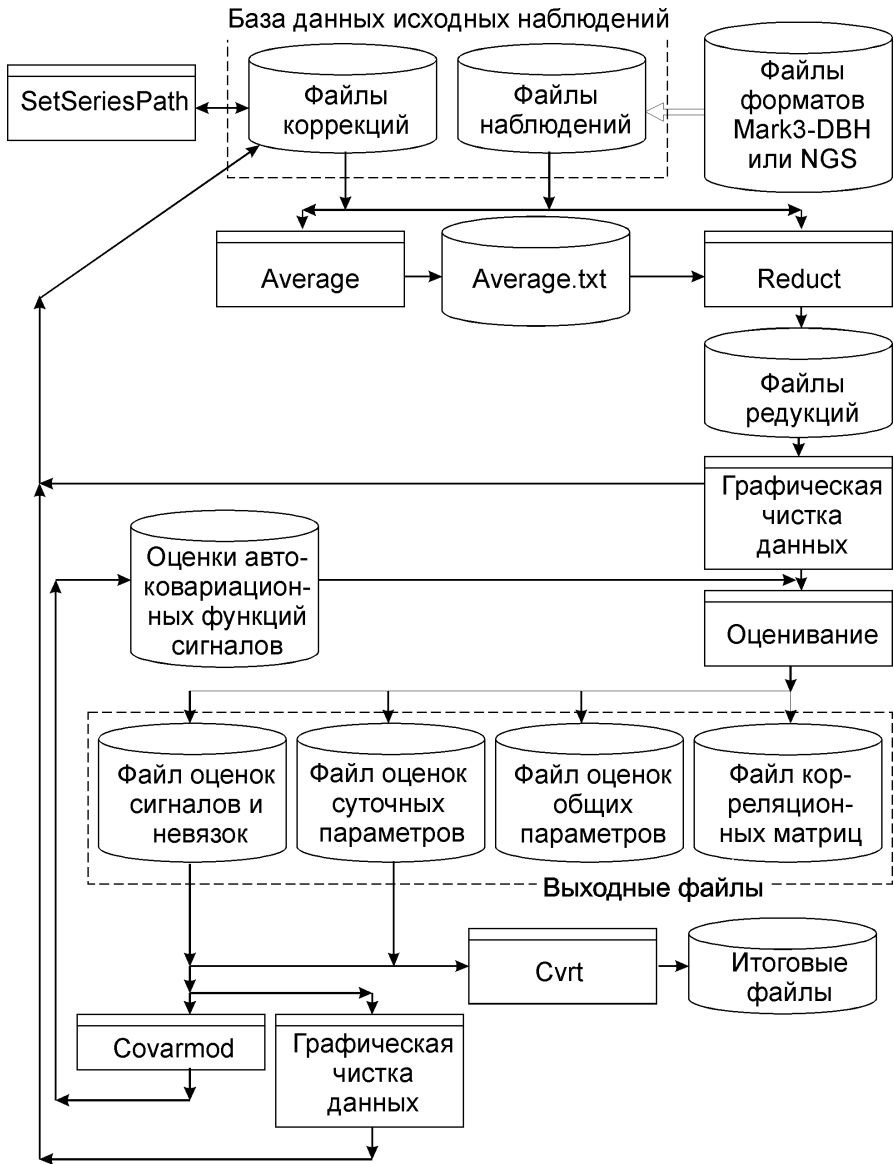


Рис. 1. Схема потока данных пакета.

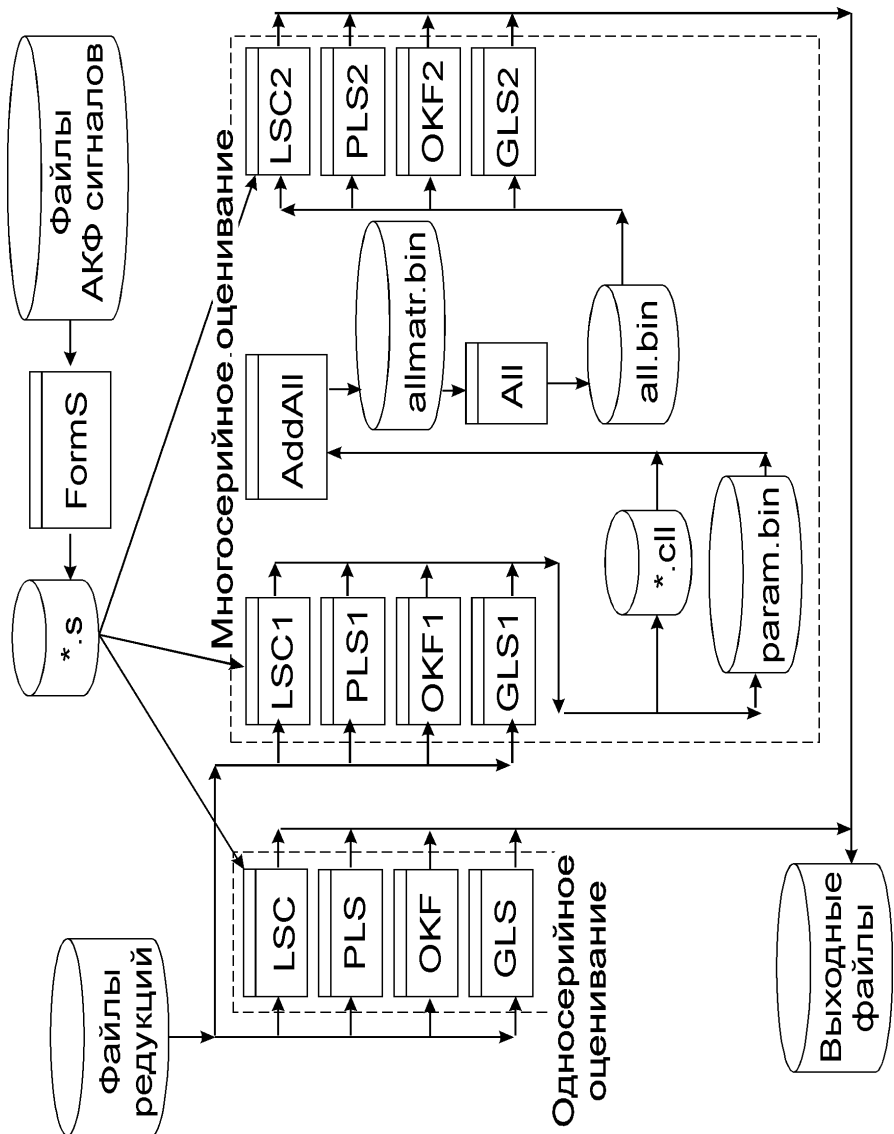


Рис. 2. Схема потока данных систем оценивания.

рис. 2). Односерийные методы оценивания реализованы каждый в виде отдельной программы: `lsc` — Least Squares Collocation (среднеквадратическая коллокация СКК), `pls` — Parametric Least Squares (многопараметрический МНК), `okf` — Optimal Kalman Filter (оптимальная фильтрация Калмана) и `gls` — Group Least Squares (многогрупповой МНК).

Многосерийное оценивание более сложно. Выполняются программы т.н. первого шага `lsc1`, `pls1`, `gls1`, `okf1`, формирующие временные файлы: файл `param.bin` с перечнем оцениваемых параметров и файлы с расширением `*.clc` и именем серии, хранящие нормальные системы общих и суточных параметров, а также некоторые другие данные. Программа `addall` производит объединение нормальных систем общих параметров каждой серии в единую систему и сохраняет ее во временном файле `allmatr.bin`. Программа `all`, обработав этот файл, оценивает общие параметры и выводит их в выходной файл и промежуточный файл `all.bin`. Наконец программы второго шага `lsc2`, `pls2`, `gls2`, `okf2` оценивают суточные и внутрисуточные параметры и формируют выходные файлы. Кроме того, перед выполнением одноили многосерийного СКК-оценивания должна выполняться программа `FormS` для создания используемых этим методом временных файлов с расширением `s`.

В результате оценивания формируются следующие выходные файлы: для всего совместно обрабатываемого ряда наблюдений — один файл, содержащий оценки общих параметров, и для каждой серии наблюдений — файл, содержащий оценки суточных параметров, файл с невязками и оценками внутрисуточных параметров и сигналов и файл с матрицами взаимных корреляций оценок суточных параметров. Содержание файлов подробнее описано в разделе "Формат выходных файлов". Файл оценок общих параметров имеет имя `all.txt`. Остальные имеют имя серии и следующие расширения: файл оценок суточных параметров `*.sng`, файл невязок — `*.sig`, и файл корреляционных матриц — `*.crm`.

Пользователь может выбрать директорий, в который записываются выходные файлы, при этом компьютер предлагает имена поддиректориев, соответствующие одному из четырех методов оценивания, например: `<quasar>\ results \ lsc_sig`, где `<quasar>` - корневой директорий пакета (см. таблицу 1).

Утилита `cvrt` читает файлы оценок суточных параметров и невязок и формирует два итоговых выходных файла: в первом `erp.txt` сведена

Таблица 1. Поддиректории выходных файлов.

Метод оценивания.	Односерийное оценивание.	Многосерийное оценивание.
Коллокация.	<code>lsc_sig</code>	<code>lsc_all</code>
Многопараметрический МНК.	<code>pls_sig</code>	<code>pls_all</code>
Фильтрация Калмана.	<code>okf_sig</code>	<code>okf_all</code>
Почасовое уравнивание.	<code>gls_sig</code>	<code>gls_all</code>

информация по оценкам и их ошибкам всех глобальных суточных параметров, во втором `rms.txt` — среднеквадратические отклонения (СКО) всех сигналов для всех обрабатываемых серий.

Утилита `output` применяется при работе пакета в режиме международной службы вращения Земли IERS. Она читает выходные файлы, указанные в таблице 1, интерполирует на момент наблюдений значения ПВЗ из опорных рядов IERS (EOP)C04 или NEOS, осуществляет редукицию поправок нутации к модели IAU(1980) и формирует выходные файлы в принятом формате IERS.

Система графической чистки данных может использоваться как после редукиционной обработки, работа с вектором L_{o-c} , так и по результатам оценивания, используя вектор невязок. В первом случае убираются возникающие иногда нерегулярности работы стандарта частоты водородного лазера на одной из станций, во втором — отбраковываются плохие наблюдения, например, по критерию "трех сигм". Результаты работы системы записываются в базу исходных данных и требуют повторного прогона редукиций и оценивания.

Для отличия устаревших вычислений вводится специальный индекс — номер версии серии. Он хранится в файле коррекций и копируется оттуда в файл редукиций и выходные файлы. Каждая модификация файла коррекций программой графической чистки увеличивает его на единицу. Первоначально или после очистки файла коррекций этот индекс равен нулю.

Для стохастических методов оценивания — среднеквадратической коллокации (СКК) и фильтрации Калмана — требуется переоценивать автоковариационные функции сигналов для каждой серии наблюдений. Это осуществляется утилитой `covarmod`. После ее работы оценивание

параметров повторяется. Для идентификации этого процесса утилитой вводится еще один индекс — номер итерации, отражаемый в выходных файлах. Первоначально он равен нулю, каждая последующая итерация увеличивает его на единицу.

В состав пакета также входят включенная в управляющую программу система просмотра корреляционных матриц и утилита **SetSeriesPath** для перемещения файлов базы исходных данных. Все вычислительные блоки реализованы в виде отдельных программ — консольных приложений, запускаемых управляющей программой — приложением Windows-95. Передача информации от управляющей программы к вычислительным осуществляется через специальные текстовые файлы (управляющие файлы). Такая организация позволяет пользователю, не знакомому с программированием под Windows-95, тем не менее вносить изменения в вычислительные блоки.

Системы пополнения базы данных файлами форматов Mark3-DBH или NGS, графической чистки данных и просмотра матриц взаимных корреляций встроены в управляющую программу пакета.

Пакет имеет довольно легкую настройку путей к различным его компонентам. Все его директории перечислены в таблице 2.

Выходные файлы записываются в поддиректории (см. таблицу 1) директория `c:\quasar\results`.

Три ненастраиваемых поддиректория должны фиксированно находиться внутри корневого директория пакета (хотя сам директорий вместе со всем содержимым может быть перенесен в другое место). Остальные можно свободно переносить на другое место (кроме файлов наблюдений; в случае их перенесения необходимо воспользоваться утилитой **SetSeriesPath**). После перенесения нужно обновить информацию об их расположении в управляющей программе (или вручную отредактировать файл `quasar.dat`, находящийся в поддиректории `cpp`).

Пакет выполнен на двух языках программирования. Большая часть вычислительных блоков написана на языке FORTRAN-90, некоторые системы оценивания и управляющая программа — на языке C++.

2. База исходных данных

Взаимодействие базы исходных данных пакета с другими системами показана на рис. 3.

Таблица 2. Директории пакета.

Значение, предлагаемое при установке.	Назначение директория.	Настраивается.
C:\quasar	Корневой директорий пакета.	Да
C:\quasar\catalog	Каталоги координат и других данных станций и источников.	Нет
C:\quasar\ephem	Эфемериды.	Нет
C:\quasar\cpp	Директорий управляющей программы.	Нет
C:\quasar\for	Директорий вычислительных программ.	Да
c:\quasar\correct	Директорий файлов коррекций.	Да
c:\quasar\series	Директорий, куда по умолчанию записываются новые файлы наблюдений.	Да
c:\quasar\results	Директорий для записи результатов вычислений.	Да
C:\quasar\temp	Директорий для хранения промежуточных данных.	Да



Рис. 3. База исходных данных и ее взаимодействие с другими программами.

При проектировании формата базы исходных данных учитывались следующие соображения. Для компактности данные необходимо хранить в двоичном виде. Однако использовать файлы фиксированной структуры нецелесообразно: состав данных наблюдений может меняться в сторону появления новых типов информации. Требовался формат гибкого хранения информации, позволяющий хранить для каждого наблюдения большой набор данных и выбирать нужные, обеспечив совместимость версий, при которой более поздняя версия пакета могла бы читать файлы более ранней версии. За основу такого формата файла был взят формат Mark3-DBH, удовлетворяющий этим требованиям, но имеющий большую избыточность структуры. Был разработан собственный формат QDB на основе упрощенного Mark3-DBH, и написаны процедуры его чтения/записи на языках программирования FORTRAN и C++. Далее была разработана система чтения/записи файлов коррекций, файлов исходных данных и объединение информации.

2.1. Формат QDB файлов

QDB-формат предназначен для хранения различных данных в виде одномерных массивов чисел; каждое такое данное может быть найдено по уникальному 8-символьному идентификатору. При этом все данные разбиваются на две группы: постоянные для всех наблюдений (например, участвовавшие в серии имена станций и источников) и меняющиеся от наблюдения к наблюдению (например, задержки). Соответственно, данные первой группы (в дальнейшем — постоянные данные) представлены один раз, а второй (в дальнейшем — данные наблюдений) — по разу для каждого наблюдения.

Файл состоит из трех частей (см. рис. 4): заголовка, содержания и блоков собственно данных. Заголовок содержит шапку, идентифицирующую QDB-формат, номер версии формата и число наблюдений. Номер версии требуется для обеспечения совместимости форматов от более ранних к более поздним. Он вместе с шапкой хранится в 8-символьной строке вида: "QDB V 1". Число наблюдений хранится в двоичном виде как 4-байтное целое со знаком.

Содержание разбито на две части: для постоянных данных и данных наблюдений. Каждая часть начинается 4-байтным целым со знаком — количеством данных в части; затем идет описание данных. Каждое данное представляется 16-байтовой записью, состоящей из 8-символьного идентификатора и трех целых чисел со знаком: 2-байтовым



Рис. 4. Структура QDB-файлов.

типом данного, 2-байтовой размерностью данного и 4-байтовым смещением.

Тип данных может принимать следующие значения:

- 0 — 8-байтовое вещественное число;
- 1 — 4-байтовое целое со знаком;
- 2 — 2-байтовое целое со знаком;
- 3 — символьное данное.

Могут также храниться и булевские значения — как символьные данные, где истинность означает ненулевой символ.

Размерность данного означает размерность его массива — количество чисел в каждой записи этого данного. Смещение выражает сдвиг в байтах данного относительно начала каждого блока данных.

Третья часть файла содержит собственно данные. Она разбита на блоки: один блок для постоянных данных и по одному блоку для данных каждого наблюдения. Данные в блоках идут в том порядке, в котором они указаны в содержании. Каждое данное хранится в виде одномерного массива, размерность, тип и смещение которого указаны в его содержании.

Блоки данных нумеруются следующим образом: блок постоянных данных имеет нулевой номер, блоки наблюдений — номер, соответствующий номеру наблюдения.

Используемая библиотека процедур на языке FORTRAN требует, чтобы длина файла была кратна двум килобайтам. Поэтому в конце файла идут дополнительные байты, дополняя его длину до кратного 2Kb значения. Они не несут информации.

2.2. Библиотека работы с QDB-форматом для C++

Библиотека представлена классом TQDB, описанным в файлах `qdb.h` и `qdb.cpp`. Так как размер файла базы данных сравнительно невелик — всего сотни килобайт данных для серии в несколько тысяч наблюдений, — то самым рациональным режимом при чтении файла оказывается его полная загрузка в оперативную память и выборка из него нужных данных, а при записи — формирование всего файла в оперативной памяти и последующая запись на диск.

Класс TQDB содержит следующие функции для чтения файла:

- `void Load(const char* aFileName)` — загрузка файла *aFileName* в оперативную память;
- `unsigned GetData(const char *aName, unsigned iType, unsigned iSize, void* Buffer)` — извлечение данного *aName* типа *iType* максимальной размерности *iSize* в буфер *Buffer* из текущего наблюдения;
- `void NextObsv()` — переход к следующему наблюдению;
- `void ToObsv(unsigned I)` — переход к наблюдению номер *I*.

Класс содержит следующие функции для формирования и записи файла:

- `void AddName(const char* A, bool bObsv, unsigned AType, unsigned ASize)` — добавить в содержание файла новое данное с именем *A*, типом *AType*, размерностью *ASize* к данным наблюдений, если *bObsv* истинно, иначе — к постоянным данным;
- `void SetCommonSize()` — завершить формирование содержания;
- `void AddObsv()` — добавить новое наблюдение;
- `unsigned PutData(const char *aName, unsigned iType, unsigned iSize, void* Buffer)` — запись данного *aName* типа *iType* максимальной размерности *iSize* из буфера *Buffer* в текущее наблюдение;
- `void Save(const char* aFileName)` — сохранить в файл *aFileName*.

Некоторые вспомогательные процедуры используются и при чтении, и при записи:

- `void ToCommonBlock()` — переход к началу файла (блоку постоянных данных);
- `inline unsigned Size(void) const` — возвращает число наблюдений;
- `unsigned Curr(void) const` — возвращает номер текущего наблюдения.

При чтении файла процедура `Load` загружает все блоки в оперативную память. Процедурой `GetData` необходимо извлечь все постоянные данные, затем последовательно для каждого наблюдения процедурой `NextObsv` переходить к очередному наблюдению и извлекать данные наблюдений процедурой `GetData`. Возможно пользование процедурами `ToObsv` и `ToCommonBlock`.

При формировании нового файла, сначала объявляется переменная типа `TQDB`, создавая в оперативной памяти новый пустой файл. Затем процедурой `AddName` формируется содержание файла; этот процесс завершается вызовом процедуры `SetCommonSize`. Система готова к приему постоянных данных, который производится процедурой `PutData`. Далее для каждого наблюдения вызывается процедура `AddObsv`, и после этого процедурой `PutData` заполняются данные этого наблюдения.

2.3. Библиотека работы с QDB-форматом для FORTRAN

Библиотека представлена рядом процедур, находящихся в файле `qdb.for`. Она позволяет одновременно работать только с одним фай-

лом. Библиотека реализует простой алгоритм работы с файлом: в оперативной памяти одновременно хранится только содержание файла и один блок данных. Это предполагает только последовательный доступ к наблюдениям.

В состав входят следующие процедуры:

- `OpenQDB` — открытие файла, указываются имя файла и режим чтения/записи, при чтении возвращается число наблюдений;
- `CloseQDB` — закрывается файл;
- `AddQDBName` — при записи добавляется запись в содержание;
- `NextQDBObsv` — переход к следующему наблюдению;
- `RdWrQDBChar` — чтение/запись массивов символов;
- `RdWrQDBInt2` — чтение/запись массивов типа `integer*2`;
- `RdWrQDBInt4` — чтение/запись массивов типа `integer*4`;
- `RdWrQDBReal` — чтение/запись массивов типа `real*8`;
- `RdWrQDBInt2S` — чтение/запись скаляра типа `integer*2`;
- `RdWrQDBInt4S` — чтение/запись скаляра типа `integer*4`;
- `RdWrQDBRealS` — чтение/запись скаляра типа `real*8`;

Система работает аналогично созданной для языка C++. Сначала вызовом `OpenQDB` открывается файл. Далее, при формировании нового файла идет формирование содержания процедурой `AddQDBName`. Затем идет чтение/запись постоянных данных процедурами `RdWrQDBChar`, `RdWrQDBInt2`, `RdWrQDBInt4`, `RdWrQDBReal`, `RdWrQDBInt2S`, `RdWrQDBInt4S`, `RdWrQDBRealS`. Затем для каждого наблюдения вызывается `NextQDBObsv` и читаются/пишутся данные наблюдения. По окончании работы файл закрывается процедурой `CloseQDB`.

2.4. Формат файлов коррекций

Файл содержит данные, указанные в таблице 3.

2.5. Формат файлов наблюдений

В настоящее время файл содержит данные, указанные в таблицах 4, 5. Возможно, набор данных будет дополнен.

Для удобства идентификаторы данных взяты из стандарта файлов формата Mark3-DBH.

Таблица 3. Содержание файла коррекций.

Идентификатор	Тип и размерность	Назначение
Постоянные данные		
VERSION	4-байтовое целое	версия серии
PATH	80-символьная строка	путь к файлу исходных данных и его имя
Данные наблюдений		
DELETED	булевская переменная	Признак плохого наблюдения
JUMP	вещественное	Величина поправки к синхронизации часов
RMS	вещественное	RMS наблюдения

Таблица 4. Содержание файла наблюдений. Постоянные данные.

Идентификатор	Тип и размерность	Назначение
JUL DATE	4-байтовое целое	Модифицированная юлианская дата.
TIME12	Два вещественных числа	Время первого и последнего наблюдения в долях суток
# SITES	2-байтовое целое	Количество станций <i>iStat</i>
SITNAMES	Массив $8 * iStat$ символов	Имена станций
AXISTYPS	Массив $4 * iStat$ символов	Типы монтаровок станций
AXISOFFS	Массив $iStat$ вещественных чисел	Величины антенных выносов
# STARS	2-байтовое целое	Количество источников <i>iSour</i>
STRNAMES	Массив $8 * iSour$ символов	Имена источников

Таблица 5. Содержание файла наблюдений. Данные наблюдений.

Идентификатор	Тип и размерность	Назначение
STATSTAR	Три 2-байтовых числа	Номера участвовавших в наблюдении станций и источника
UTJ	Вещественное число	Время наблюдения
DELUFLAG	2-х байтовое целое	Признак качества наблюдения — 0, если хорошее
ION CODE	2-х байтовое целое	Признак качества измерения ионосферной задержки — 0, если хорошая
DEL OBSV	Вещественное число	Измеренная задержка
DELSIGMA		Ошибка задержки
RAT OBSV		Измеренная частота интерференции
RATSIGMA		Ошибка частоты интерференции
ION DEL		Измеренная ионосферная задержка
ION RAT		Ошибка ионосферной задержки
ION DELS		Измеренная ионосферная частота интерференции
ION RATS		Ошибка ионосферной частоты интерференции
CABL DEL		Два вещественных числа
TEMP C	Температура воздуха станций	
ATM PRES	Атмосферное давление станций	
REL HUM	Влажность воздуха станций	

2.6. Система пополнения базы исходных данных новыми сериями

Основная часть этой системы находится в файлах `read.h` и `read.cpp`. Там находится процедура `InputNewSeries` для добавления файлов форматов NGS и Mark3-DBH. Процедура вызывает процедуру `NGSQDB` для декодировки NGS файлов, либо `MK3QDB` — для файлов Mark3-DBH. Форматы этих файлов описаны в [4] (NGS) и [1] — Mark3-DBH. Для чтения файлов Mark3-DBH используется класс `TMk3` (файлы `mk3proc.h` и `mk3proc.cpp`).

Осуществляется также добавление уже готовых файлов наблюдений либо пар из файла коррекций и файла наблюдений. Эти операции, а также управление процессом обновления базы исходных данных производится из модуля `Unit5`.

3. Программа `average`

Программа находится в файле `average.for`. Подобно программе `reduct`, она считывает из управляющих файлов `task.dat` и `statsour.dat` списки обрабатываемых серий, станций и источников, читает каждую серию в базе исходных данных и вычисляет средние арифметические значения температуры, давления и влажности для всех имеющихся в списке станций. Полученные величины записываются в файл `average.txt` директория для временных файлов.

4. Программа `reduct`

Схема потока данных системы редукций приведена на рис. 5. Система редукций использует следующие данные:

- Перечень настроечных параметров из файла `reduct.dat`;
- Перечень обрабатываемых станций и источников из файла `statsour.dat`;
- Перечень обрабатываемых серий из файла `task.dat`;
- Исходные данные из базы данных;
- Сведения о скачках часов и плохих наблюдениях из файлов коррекций;
- Эфемериды Земли, Луны, Солнца и планет, а также опорные значения EOP из каталога эфемерид;

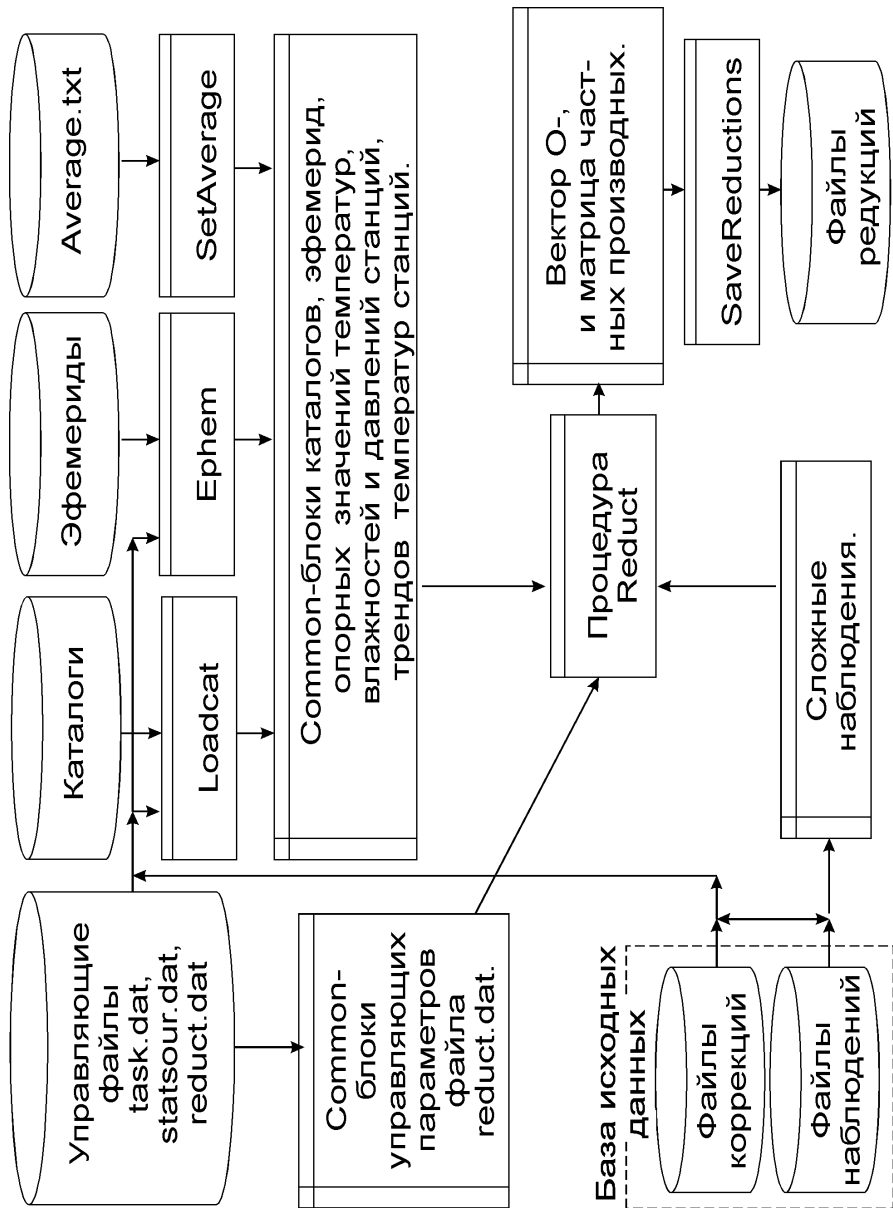


Рис. 5. Схема потока данных редукций.

- Каталоги координат станций и источников; параметры атмосферных и океанических нагрузок, послеледниковых поднятий, температурных деформаций антенн; данные о структуре источников.

Результатом работы является файл редукиций с расширением `red`, составляемый для каждой серии и включающий в себя:

- Количество и имена станций и источников;
- Количество наблюдений;
- Для каждого наблюдения номера станций и источника;
- Вектор моментов наблюдений в MJD;
- Вектор задержек O-C;
- Вектор ошибок наблюдений;
- Матрицу частных производных с именами и типами параметров.

Система редукиций состоит из следующих частей:

- Собственно блок редукиционных вычислений;
- Блок чтения каталогов;
- Блок чтения файлов коррекций и файлов наблюдений;
- Блок чтения эфемерид;
- Блок записи результатов в файл редукиций;
- Управляющий блок.

4.1. Блок редукиционных вычислений

Блок редукиционных вычислений включает в себя процедуру `reduct` из файла `reduct.for` и вызываемые ею процедуры из файла `red_sub.for`. Блок спроектирован так, что одновременно обрабатываются данные одного "сложного" наблюдения, т.е. всех синхронных наблюдений нескольких станций одного и того же источника. Такая схема, принятая также в пакете `Ossam`, существенно ускоряет вычисления. Однако до начала вычислений требуется произвести сортировку поступающих простых наблюдений. Эту задачу решает блок чтения исходных данных. Так как алгоритмы оценивания работают только с простыми наблюдениями, то после редукиций необходимо объединить все сложные наблюдения в единую цепь простых, что и делает блок записи.

Процедура `reduct` вызывается в цикле для каждого сложного наблюдения. Перед этим циклом необходимо вызывается процедура `Param`

из файла `red_sub.for` для инициализации используемых в вычислениях констант. Если формируются файлы дополнительной информации, то вызывается также процедура `GetAddNames`. После завершения цикла используется процедура `GetXNames` для установки количества, типов и имен столбцов матрицы частных производных.

Процедура `reduct` получает следующие данные, не зависящие от номера наблюдения, через `common`-блоки:

1. Каталоги координат станций и источников;
2. Эфемериды;
3. Опорные системы ПВЗ;
4. Мировые константы;
5. Настраечные параметры.

Остальные данные, зависящие от номера наблюдения, поступают через обращение к процедуре:

1. Модифицированные юлианская дата и время наблюдения;
2. Номер наблюдаемого источника по каталогу;
3. Количество станций в данном сложном наблюдении;
4. Для каждой станции — номер станции по каталогу, температура, влажность, давление и кабельная задержка в момент наблюдения;
5. Количество баз в данном сложном наблюдении;
6. Для каждой базы — номера ее первой и второй станции в списке станций этого сложного наблюдения, наблюдаемые задержка и частота интерференции с ошибками, их ионосферные поправки с ошибками.

В результате работы процедуры `reduct` выдает следующие данные для каждого простого наблюдения:

- Разность типа О-С для задержки с ее ошибкой;
- Строку матрицы частных производных для задержки;
- Разность типа О-С для частоты интерференции с ее ошибкой;
- Строку матрицы частных производных для частоты интерференции.

Если настроенный параметр `bAdd` истинен, то генерируется также дополнительный текстовый файл с расширением `add`, в который выводится отладочная информация. Подробно алгоритм редукций описан в [2].

4.2. Блок чтения каталогов

При формировании задания на проведение редуцированных вычислений формируется также список используемых станций и источников. Система чтения каталогов для этих источников читает данные из каталогов и заносит в common-блоки для использования вычислительным блоком.

Физически блок чтения каталогов находится в файле `loadcat.for` и представлен процедурой `LoadCatalog` и рядом вызываемых ею процедур. Этим блоком производятся следующие действия:

1. Из файла `sta_voc.dat` читаются 9-символьный код станции (для идентификации в файле координат станций `station.dat`), величина поправки к выносу антенны *CorOff* (сам тип антенны и исходная величина ее выноса поступают из базы исходных данных), и коэффициент вертикального прогиба земной коры вследствие атмосферной нагрузки в районе станции *CoATL*;
2. Из файла `station.dat` читаются координаты и скорости станций;
3. Станции, координаты которых не найдены в файлах `sta_voc.dat` или `station.dat`, исключаются из списка обрабатываемых станций;
4. Из файла `sour_voc.dat` для источников, представленных в этом списке псевдонимами, читаются стандартные цифровые имена;
5. Из файла `source.dat` читаются координаты источников;
6. Источники, координаты которых не найдены в файле `source.dat` и их псевдонимы в файле `sour_voc.dat`, исключаются из списка обрабатываемых источников;
7. В зависимости от настройки параметра *iStruct* (из файла настроек редуциций `reduct.dat`) читаются данные о структуре источников либо для многокомпонентной модели из файлов `*.mod` поддиректория `Struct`, поступая в common-блок *SourStruct*, либо для двухкомпонентной модели из файла `Struct`, поступая в переменную *StructMod* common-блока *Sources*;
8. Из файла `ocload.dat` загружаются данные об океанических нагрузках в районе станций;
9. Из файла `thermal.dat` загружаются данные о температурных деформациях антенн;
10. Из сформированного программой `Average` файла `average.txt` читаются данные о средних значениях давления, температуры и влажности на станциях (выполняющая чтение процедура `SetAverage` вызывается отдельно из управляющего блока).

4.3. Блок чтения файлов коррекций и наблюдений

Данный блок вызывается в цикле для каждой серии, читает исходные данные и сортирует их по сложным наблюдениям. Для работы блоку требуются имя серии и список обрабатываемых станций и источников. Блок представлен процедурами, находящимися в файле `qdbrdwr.for`. Эти процедуры используют систему доступа к базе исходных данных пакета, содержащуюся в файле `qdb.for` (подробнее см. раздел "База исходных данных").

4.4. Блок чтения эфемерид

Блок использует два настроечных параметра — *iEop*, указывающий, какой из опорных рядов ПВЗ (IERS (EOP) C04 или NEOS) будет использоваться, и *iLagr* — параметр, используемый в настройке интерполяции методом Лагранжа. При этом реальное количество табличных значений эфемерид для данной серии будет определяться следующим образом: $iLagr * 2 + N$, где N — число табличных значений эфемериды, попавшее внутрь серии, то есть берутся все точки, попадающие внутрь серии и по *iLagr* точек до первого и после последнего наблюдений серии.

Блок состоит из двух частей. Одна из них — инициализирующая работу блока процедура `InitEphem` в файле `loadcat.for` — вызывается один раз в начале работы редукиций. При этом считываются форматы данных, длины строк, начальные моменты времени и интервалы времени между точками всех рядов. Вторая часть блока — процедура `Ephem` из того же файла — собственно читает эфемериды и вызывается для каждой серии наблюдений; ее входные параметры — моменты первого и последнего наблюдений серии и передаваемое через `common`-блок число *iLagr*. Файлы эфемерид читаются как файлы прямого доступа: вычисляется число точек от первой до читаемых, которое затем умножается на длину строки в байтах и осуществляется сдвиг начала считывания; затем каждая точка (строка в файле) читается в символьную переменную и декодируется.

4.5. Блок записи результатов в файл редукиций

Блок состоит из процедуры `SaveReductions` и ряда вызываемых ею процедур, находящихся в файле `mainred.for`, а также `RdWrReduct`

из файла `accdata.for`, собственно записывающей/читающей файлы редуцированных. Блок вызывается для каждой серии данных и осуществляет следующие действия:

1. В списках станций и источников оставляются только те, которые реально участвовали в наблюдениях в данной серии;
2. Запрашиваются имена и типы столбцов в матрице частных производных (вызывается процедура `GetXNames` из файла `reduct.for`), удаляются пустые столбцы;
3. Процедурой `RdWrReduct` вся информация записывается в файл редуцированных.

4.6. Управляющий блок

Процедуры управляющего блока находятся в файле `mainred.for`. Управляющий блок производит следующие действия:

1. Инициализирует пути, читает список обрабатываемых серий, источников, станций и параметров редуцированных;
2. Вызывает процедуру чтения каталогов, а также читает средние значения температур, давлений и влажностей для всех станций;
3. Инициализирует систему чтения эфемерид;
4. Инициализирует используемые в редуцированных константы;

Далее для каждой серии производит следующие действия:

1. Читает файл коррекций и файл наблюдений;
2. Определив момент первого и последнего наблюдений серии, читает эфемериды;
3. Вычисляет тренд второго порядка температуры для каждой станции;
4. В цикле для каждого сложного наблюдения вызывает процедуру `reduct`;
5. Формирует файл редуцированных (см. блок записи результатов);

5. Системы оценивания

Системы оценивания унифицированы по блокам сортировки (извлечения данных из файлов редуцированных и формирования матриц частных производных) и вывода результатов, отличаясь лишь собственно вычислительными блоками.

Однако часть систем оценивания — односерийная и многосерийная коллокация, многопараметрический МНК — выполнены на языке программирования FORTRAN, а другая часть — фильтрация Калмана и почасовое оценивание — на C++. Поэтому разработаны два комплекта библиотек процедур сортировки и печати для каждого из языков программирования.

Система сортировки данных запускается перед оцениванием параметров каким-либо из указанных выше способов и после считывания файла редуций формирует матрицы частных производных для общих, суточных и внутрисуточных компонентов параметров. Для внутрисуточных компонентов формируются также их автоковариационные функции — из управляющего файла `signals.dat` или из файла автокорреляционных функций для данной серии. Список оцениваемых параметров и режимы их оценивания берутся из управляющего файла `der.dat`. Система печати для каждой серии выводит файл оценок поправок к суточным параметрам, файл оценок невязок и внутрисуточных параметров; при многосерийной обработке создается еще и файл оценок общих параметров. Теория методов оценивания подробно изложена в [3].

5.1. Системы сортировки и печати для оценивания методами СКК и МНК

Процедуры сортировки находятся в файле `sortpar.for`, печати — в файле `printing.par`. Система сортировки разрабатывалась в первую очередь для использования в методе СКК, требующем объемных вычислений с большими матрицами, и поэтому основной упор делался на повышение быстродействия вычислений при сохранении гибкости пакета. С этой целью все матрицы частных производных формируются, содержатся и участвуют в вычислениях в так называемом сжатом виде. В каждой их строке хранятся только ненулевые элементы: частные производные глобальных параметров, параметров, участвовавших в данном наблюдении источника, и двух станций. Это позволяет сократить число столбцов в матрице в несколько раз при оценивании параметров станций и в десятки раз при оценивании параметров радиоисточников. "Сжатые" и "развернутые" матрицы частных производных изображены на рис. 6.

Для хранения информации о составе оцениваемых параметров предусмотрены два common-блока — *Parameters* для общих и суточных



а)



б)

Рис. 6. Развернутая (а) и сжатая (б) матрицы частных производных. Нулевые элементы показаны белым цветом, ненулевые — черным.

параметров и *Signals* — для сигналов. Их переменные хранят информацию о количестве глобальных параметров, параметров для каждого источника и каждой станции, имена параметров, а для общих и суточных параметров — максимальную степень их тренда. Процедура `SortParams`, вызываемая перед началом работы, устанавливает эти переменные, считывая файл `der.dat`.

Далее для каждой обрабатываемой серии считывается файл редукций со всеми частными производными, которые генерируются редукциями. Процедурой `SortSerParams` из этих частных производных формируются сжатые матрицы со всем набором параметров и всеми необходимыми трендами. Процедура `LoadSignals` считывает автоковариационные функции сигналов. При этом сначала делается попытка считать функции, полученные именно для этой серии — из файла с именем серии и расширением `cov`, находящимся в директории временных файлов. Если такового нет либо он содержит не все требуемые сигналы, используется априорные функции из управляющего файла

`signals.dat`. Автоковариационные функции хранятся в common-блоке *Signals* в виде дисперсии и массива значений параметров корреляционной функции. Процедура `CorrFunc` файла `quas_sub.for` производит для требуемого сигнала и момента времени вычисление ковариационной функции заданного вида (см. [3]).

Иногда требуется производить оценивание, используя вместо вектора разностей (O-C) либо вектор невязок, либо вектор суммы невязок и внутрисуточных сигналов. Значения этих векторов берутся из ранее полученных оценок сигналов и невязок считыванием информации из файла `*.sig`. Эта операция выполняется внутри процедуры чтения редуциций `RdWrReduct` (файл `accdata.for`).

Печать оценок параметров осуществляется процедурой `SaveParameters` с использованием информации о необходимых единицах измерения из управляющего файла `parunits.dat`. Все оценки параметров печатаются в метрах и, если параметр имеет другие единицы измерения, — также и в этих единицах. Процедура производит печать в уже открытый файл и не закрывает его. Это делают вызывающие ее процедуры `SaveLocal` печати оценок суточных и внутрисуточных параметров и `SaveCommon` — печати общих параметров.

5.2. Программа `FormS` построения и обращения матриц ковариаций

Эта программа используется обоими (одно- и многосерийным) методами СКК-оценивания, фактически являясь их составной частью. Она рассчитывает для каждой серии наблюдений используемую в СКК матрицу $S = CKC' + Q$, где C — матрица частных производных сигналов, K — матрица автоковариаций сигналов, Q — матрица ковариаций невязок, а также обратную матрицу S^{-1} (см. [3]). Процедуры программы находятся в файле `forms.for`.

Программа работает только с сигналами. Для каждой серии процедурой `FormS` формируется сжатая матрица частных производных сигналов C , а процедурой `ProcS` — матрица S , затем эта матрица обращается стандартной процедурой `dsinv`, взятой из библиотек стандартных программ, и, наконец, матрицы C и S^{-1} записываются во временный файл процедурой `RdWrS` из файла `accdata.for`.

СКК-алгоритм получает оценки сигналов и учитывает их влияние на каждое наблюдение, что делает этот алгоритм довольно громоздким. В связи с этим основное внимание при разработке программы

уделялось устранению операций с нулевыми элементами матриц частных производных и функций взаимных корреляций наблюдений; эти довольно большие даже в сжатом виде матрицы содержат значительную часть нулевых элементов, и попытка их обработать в развернутом виде (размерностью в десятки тысяч строк и столбцов) сделала бы невозможной реализацию СКК-алгоритмов.

В самом деле, матрица \mathbf{C} имеет размер $N \times M$, где N — число наблюдений в серии, M — число сигналов, матрица \mathbf{K} — имеет размер $M \times M$. Реально же матрица \mathbf{K} является блочно-диагональной, т.к. различные сигналы считаются некоррелированными друг с другом, а матрица \mathbf{C} — состоящей из блоков диагональных матриц, поскольку на каждое наблюдение оказывают воздействие только сигналы этого наблюдения. Матрица \mathbf{Q} — диагональная, т.к. невязки считаются некоррелированными. Матрицы \mathbf{C} и \mathbf{K} изображены схематически на рис. 7. Целесообразно хранить матрицу \mathbf{C} в виде M векторов и \mathbf{Q} в виде вектора длиной N . Матрица \mathbf{K} задается в виде корреляционной функции для каждого сигнала и вычисляется непосредственно перед использованием.

Кроме того, легко убедиться, что значение элемента d_{ij} матрицы \mathbf{CKC}' можно вычислить по формуле: $d_{ij} = \sum_{m=1}^M c_{m,i} c_{m,j} k_{m,i,j}$, где $c_{m,i}$, $c_{m,j}$ — значения матрицы частных производных, $k_{m,i,j}$ — значения автоковариаций i -го и j -го наблюдений m -го сигнала. При этом нет необходимости проводить суммирование по всем M сигналам: произведение $c_{m,i} c_{m,j}$ будет не равно нулю, только если m -й сигнал участвует и в i -м и в j -м наблюдениях. Это справедливо для всех глобальных сигналов, а для сигналов станций и источников — только если станция (источник) участвовали в каждом из наблюдений. Таким образом, устраняется обработка нулевых элементов.

Все это позволяет сделать вычисления достаточно быстрыми и не требующими хранения в оперативной памяти более чем одной матрицы размерности $N \times N$.

5.3. Односерийное оценивание методом СКК

Процедуры метода содержатся в файле `lsc.for`. В цикле для каждой серии производится чтение файла редукиций и файла с матрицей \mathbf{S}^{-1} , вызываются процедуры сортировки данных. Затем вычисляются матрица и вектор нормальной системы суточных параметров.

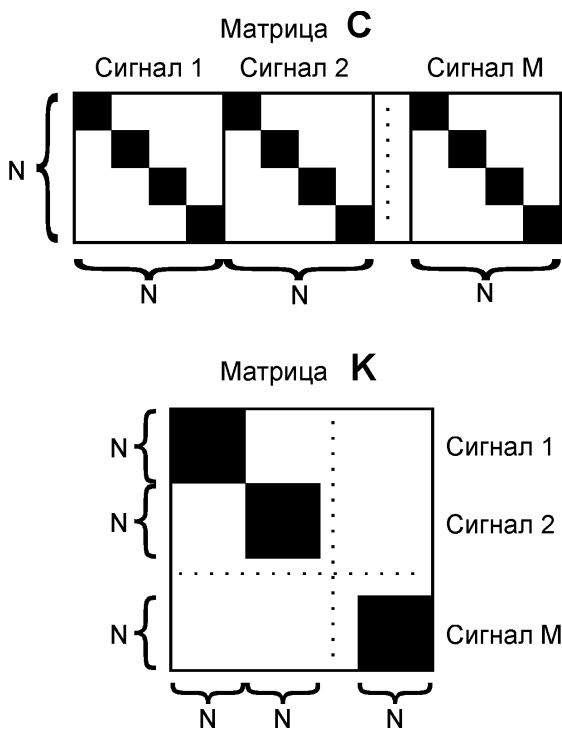


Рис. 7. Матрицы частных производных сигналов и их автоковариаций. Нулевые элементы показаны белым цветом, ненулевые — черным.

В случае, если в управляющем файле `task.dat` задается режим регуляризации, он исполняется. Для этого вызывается процедура `Regularization` из файла `quas_sub.for`, которая читает управляющий файл `regular.dat`, выбирает оттуда данные для вычисления параметров регуляризации на заданную дату наблюдений, преобразует эти данные в метры, используя единицы измерений из файла `parunits.dat`, вычисляет параметры стохастической регуляризации и прибавляет их к соответствующим диагональным элементам матрицы нормальной системы.

Процедурой `SeriesConstraints` из файла `sortpar.for` накладываются условия (ограничения) на суточные параметры. В настоящий момент используются следующие ограничения: сумма всех взаимных трендов часов (параметр с идентификатором `CLCK-off`) равна нулю.

Эти условия используются при решении системы нормальных уравнений способом Лагранжа (см. [3]).

В результате решения нормальной системы уравнений находятся суточные параметры и их ошибки. После этого, если в управляющем файле `task.dat` задается оценивание сигналов, вызываются процедуры `CalcS` и `CorrectY` из файла `quas_sub.for`. Процедура `CalcS` вычисляет оценки сигналов. При этом используются сокращения вычислений, аналогичные сокращениям в процедуре `ProcS` программы `FormS` (см. выше). Также вычисляются средние арифметические значения сигналов и их дисперсии. Затем сигналы центрируются (из них вычитаются средние значения) и вычисляется вектор влияния оценок сигналов $\hat{\mathbf{s}}$ на наблюдения: $\mathbf{L}_s = \mathbf{C}\hat{\mathbf{s}}$. Процедурой `CorrectY` полученные средние значения сигналов прибавляются к соответствующим суточным параметрам.

В случае, если в управляющем файле `task.dat` задается перевзвешивание наблюдений, оно выполняется.

Наконец формируется выходной файл оценок суточных параметров и — если сигналы вычислялись — файл невязок и оценок внутрисуточных параметров.

Программа имеет следующую оптимизацию вычислений. В ходе работы требуется вычислять промежуточную матрицу $\mathbf{T} = \mathbf{B}'\mathbf{S}^{-1}$ и матрицу нормальной системы $\mathbf{G} = \mathbf{T}\mathbf{B}$, где \mathbf{B} — сжатая матрица частных производных суточных параметров размерности $N \times M$, \mathbf{S}^{-1} — вычисленная программой `FormS` матрица размерности $N \times N$, симметричная и хранящаяся в виде одномерного массива, содержащего ее верхний треугольник, N — число наблюдений, M — число суточных параметров.

Вычисление матрицы \mathbf{T} осуществляется процедурой `MultQ()` из файла `matr.for`. По определению, элемент матрицы \mathbf{T} t_{ij} вычисляется по формуле: $t_{ij} = \sum_{k=1}^N b_{k,i} s_{k,j}$, где $b_{k,i}$, $s_{k,j}$ — элементы матриц \mathbf{B} и \mathbf{S}^{-1} соответственно, что предполагает алгоритм из трех вложенных циклов — по $i = 1, M$, $j = 1, N$ и $k = 1, N$. Это было бы эффективно, если бы матрица \mathbf{S}^{-1} хранилась как обычный двумерный массив. Однако от этой симметричной матрицы хранится лишь ее верхний треугольник в одномерном массиве. Потребовалось бы либо разворачивать ее в двумерный массив — что с учетом ее размеров не всегда возможно, либо — каждый раз вычислять смещение к элементу $s_{k,j}$, что не произвительно. Был применен другой алгоритм: выполнять цикл $i = 1, M$

как самый внутренний. Тогда элемент $s_{k,j}$ читается только один раз. Кроме того, у матриц \mathbf{B} и \mathbf{T} доступ осуществляется к расположенным подряд элементам (с этой целью матрица \mathbf{B} хранится как транспонированная), что оптимизирует работу процессора. Далее, поскольку матрица \mathbf{B} является упакованной, то цикл $k = 1, N$ проходит только по ее ненулевым элементам. Для унификации системы весь алгоритм распаковки находится в одной процедуре `UnPackP` файла `quas_sub.for`. Процедура, получив номер наблюдения и через `common`-блок — номера источников и наблюдавших станций, выдает для матрицы частных производных общих или суточных параметров количество ненулевых элементов и в виде целочисленного массива места, куда они должны встать в "развернутом" виде. Этот процесс используется в процедуре `MultT` файла `matr.for` при вычислении $\mathbf{G} = \mathbf{T}\mathbf{B}$. Таким образом удается избежать работы с нулевыми элементами.

5.4. Оценивание методом многопараметрического МНК

Процедуры метода содержатся в файле `pls.for`. Программа по сути скопирована с программы `lsc` и сильно упрощена за счет того, что вместо симметричной матрицы \mathbf{S} в алгоритме фигурирует диагональная матрица \mathbf{Q} , а сигналы отсутствуют.

5.5. Многосерийное оценивание

Многосерийное оценивание похоже на односерийное, отличаясь наличием блока оценивания общих параметров. В результате и вся система усложняется. Каждый метод реализуется двумя уникальными программами и двумя общими для всех методов `addall` и `add`.

Программы первого шага `lsc1`, `pls1`, `okf1`, `mkh1` для каждой серии вычисляют и сохраняют в файлах с расширением `clc` и именем серии в директории для временных файлов следующие промежуточные величины: матрицы и вектора нормальных систем для общих параметров, матрицы и вектора для расчета суточных и внутрисуточных параметров. Программа `addall` объединяет данные всех серий, `add` — вычисляет оценки общих параметров и формирует их файлы — выходной текстовый с именем `all.txt` и временный двоичный `all.bin` для использования программами второго шага `lsc2`, `pls2`, `okf2`, `mkh2`, которые для каждой серии завершают вычисления, производя оценки суточных параметров и сигналов.

Программы первого и второго шагов основные свои элементы заимствуют из программ односерийного оценивания. Единственная особенность — учет программой второго шага полученных оценок общих параметров. Их список формируется с учетом станций и источников, участвовавших во всех одновременно обрабатываемых сериях (т.н. полные списки источников, станций, общих параметров); однако сама программа имеет списки станций и источников и, соответственно, требует набора оценок общих параметров только для текущей серии наблюдений (т.н. короткие списки). Специальная процедура `SetX` файла `quas_sub.for` (FORTRAN) или файлов `sortpar.h`, `sortpar.cpp` (C++) производит преобразование от полного списка общих параметров к текущему.

Программа `addall` получает матрицы \mathbf{P}_k и вектора \mathbf{p}_k нормальной системы общих параметров для каждой k -й серии и должна выполнить суммирование $\mathbf{P} = \sum_k \mathbf{P}_k$, $\mathbf{p} = \sum_k \mathbf{p}_k$. Однако нормальные системы для каждой серии сформированы для короткого списка, а в результате суммирования требуется получить полный. Это выполняется процедурой `SummX` файла `addall.for`. Сначала происходит просмотр всех серий и составляется полный список всех станций, источников и, соответственно, параметров. Заносятся нули во все элементы матрицы \mathbf{P} и вектора \mathbf{p} . Далее для каждой серии ищется соответствие каждого ее общего параметра общему параметру полного списка, после чего все элементы матрицы \mathbf{P}_k и вектора \mathbf{p}_k добавляются к соответствующим элементам матрицы \mathbf{P} и вектора \mathbf{p} .

Получение суммарной нормальной системы — относительно трудоёмкий процесс, а последующие действия часто приходится выполнять несколько раз. Поэтому целесообразно было выделить суммирование в отдельную программу. Дальнейшие действия выполняются программой `all`. Она сама накладывает ограничения, определяя наличие в списке параметров координат источников и прямоугольных или топоцентрических координат. В процессе используются априорно заданные координаты источников и станций, которые берутся из каталогов (см. редукции, чтение каталогов координат станций и источников). Эти действия выполняются процедурами `SetLimits` и `AppendLimits` файла `all.for` (там же находятся и другие уникальные процедуры этой программы).

На практике не для всех общих параметров могут быть получены оценки. Например, нельзя получить поправки к координатам источника, который наблюдался только один раз; если источник наблюдался несколько раз, но плохо, например, низко над горизонтом, ошиб-

ки его оценок будут велики и негативно повлияют на другие, "хорошие" параметры. Поэтому может возникнуть необходимость исключить часть параметров из нормальной системы до ее решения. В настоящий момент имеется возможность исключения любого параметра, любой станции, любого источника, и скоростей любой станции. С этой целью вводится управляющий файл `extract.dat`, содержащий имена исключаемых параметров.

После считывания этого файла программа с помощью процедур `ExtractGlobal`, `ExtractStation` и `ExtractSource` исключает из полного списка соответствующие параметры, строки и столбцы из матрицы P и строки из вектора p . При считывании оценок общих параметров программами второго шага многосерийного оценивания оценки общих параметров имеющихся в данной серии станций и источников приравниваются к нулю. Однако, для скоростей станций этот способ не подходит, так как в пакете все станции обязаны иметь одинаковое количество параметров. Поэтому скорости станций исключаются из нормальной системы уравнений непосредственно перед обращением её матрицы, и сразу после обращения в соответствующие места матрицы вставляются нулевые строки и столбцы. В результате оценки исключаемых скоростей станций, их ошибки и корреляции присутствуют во всех выходных файлах в виде нулевых чисел.

Перед решением системы дополнительно записывается выходной файл, уже содержащий номера и имена параметров и имеющий нули вместо значений их оценок и ошибок. Если система не разрешима, будет выдан список номеров неопределяемых параметров; восстановив по этому файлу имена "плохих" станций и источников, их добавляют в файл `extract.dat` и повторяют работу программы. Так же поступают, если система разрешается, но часть параметров имеет большие ошибки.

5.6. Системы сортировки и печати на языке C++ для оценивания методами фильтрации Калмана и почасового оценивания

При программировании алгоритмов оценивания на языке C++ рационально было бы использовать библиотеку матрично-векторной алгебры, которая перегружала бы арифметические операторы, ускоряя процесс программирования. Требовалась библиотека программ для работы с большими векторами и матрицами — до нескольких тысяч строк и столбцов, достаточно производительная — операции не должны вы-

полняться существенно медленнее, чем такие же операции, написанные в стандарте языка С, и — бесплатная. К сожалению, такую библиотеку найти не удалось, поэтому была создана собственная. Она содержится в файлах `matrix.h` и `matrix.cpp`. Системы сортировки и печати выполнены на базе этой библиотеки.

Фильтрация Калмана и почасовое оценивание требуют гораздо меньше вычислительного ресурса, чем СКК-оценивание. Поэтому эта система сортировки существенно проще системы, предназначенной для метода СКК. Все матрицы частных производных формируются и обрабатываются как "развернутые". Система сортировки содержится в файлах `sortpar.h`, `sortpar.cpp` и `readred.cpp` (последний содержит процедуру чтения файла редуций).

Система делает доступной три класса: `TSeries`, `TParamCompressed` и `TParameters`, а также три процедуры: `ReadDerDat`, `SortParams` и `SeriesConstraints`. Класс `TSeries` содержит информацию об одиночной серии данных и включает конструктор ее чтения из файла редуций. Он включает доступные для потребителя ссылки на вектора *Time* моментов времени наблюдений, *Loc* измеренных задержек и *Err* ошибок наблюдений, количество и имена станций и источников. При чтении файла редуций также выдается хранившаяся в нем матрица частных производных с сопутствующими параметрами, необходимая для сортировки. Для почасового оценивания имеется процедура `Hrs`, разбивающая серию на ряд групп одинаковой длительности и возвращающая массив номеров наблюдений — границ этих групп (см. [3]).

Классы `TParamCompressed` и `TParameters` хранят массив оцениваемых параметров — либо общих, либо суточных, либо внутрисуточных. `TParamCompressed` хранит их в сжатом виде (глобальные параметры и набор параметров для станций и для источников) и имеет процедуру распаковки `Uncompress` — получения развернутого массива параметров `TParameters` для серии `TSeries`. При этом каждый параметр включает всю необходимую информацию — имя, тип (глобальный/станций/источников), степень тренда, а если список не сжатый — номер станции и источника. Кроме того, класс `TParamCompressed` хранит количество глобальных параметров и параметров для каждой станции и источника.

Процедура `ReadDerDat` читает управляющий файл `der.dat` и формирует сжатые списки для общих, суточных и внутрисуточных параметров (типа `TParamCompressed`).

Процедура `SortParams` для серии `TSeries` и параметров `TParamCompressed` формирует матрицу частных производных.

Наконец, процедура `SeriesConstraints` накладывает ограничения на нормальную систему суточных параметров.

При проектировании вышеперечисленных классов использовались шаблоны стандартной библиотеки C++.

Система печати находится в файлах `printing.h` и `printing.cpp` и представлена процедурами `PrintCommon` для печати общих и `PrintSeries` — суточных и внутрисуточных параметров.

6. Дополнительные программы

6.1. Программа `covarmod`

Эта программа используется в процессе оценивания методом СКК. Программа читает из получаемых файлов невязок (*.sig) оценки сигналов, вычисляет для них автоковариационную функцию и аппроксимирует ее, получая новые значения дисперсии и параметров корреляционной функции. Априорные значения этих величин читаются из управляющего файла `signals.dat`. Для каждого сигнала программа может работать в одном из трех вариантов: а) оставить априорные значения; б) вычислить новую дисперсию, оставив априорные значения корреляционной функции; в) вычислить и дисперсию, и корреляционную функцию. Информация о режиме обработки сигналов задается в управляющем файле `covarmod.dat`. Программа написана на языке C++. Ее головная часть находится в файле `covarmod.cpp`.

6.2. Программа `svrt`

Программа формирует два итоговых файла обработки набора серий данных: файл с оценками глобальных параметров и файл с оценками дисперсий сигналов. Информация, выводимая в файлы, перечисляется в разделе "формат выходных файлов".

6.3. Программа `output`

Программа формирует выходные файлы для отправки в международную службу вращения Земли IERS. Она считывает выходные файлы оценивания суточных параметров, выбирает из них поправки ПВЗ и их

СКО, осуществляет редукцию поправок нутации к модели IAU(1980), интерполирует на средний момент наблюдений значения ПВЗ из одного опорного ряда (IERS(EOP)C04 или NEOS), прибавляет поправки к этим опорным значениям, создает выходной файл в формате IERS в выбранном директории. Имя файла `eop_sig.txt` — для односерийной обработки, `eop_all.txt` — для многосерийной.

Программа находится в файле `output.for`.

7. Графическая программа чистки

Данная программа может запускаться как после выполнения редукций, так и после оценивания. Она читает файлы редукций, а во втором случае — файлы невязок. Вводимые исправления записываются в файл коррекций базы исходных данных.

Физически программа является частью управляющей программы. Она выполнена в системе Borland C++ Builder под операционную систему Windows-95. Конструктивно программа состоит из трех модулей.

Модуль `redacc.h`, `redacc.cpp` предоставляет класс `TReduct`, читающий данные из файлов редукций и невязок, производящий всю математическую обработку и записывающий результат в файл коррекций.

Модуль `scchart.h`, `scchart.cpp` осуществляет работу с графиками. Необходимость создания такой системы вызвана отсутствием другой доступной системы, отрисовывающей тысячи точек (имеющиеся бизнес-системы быстро работают только на сотнях точек).

Модуль `unit14.h`, `unit14.cpp` осуществляет пользовательский интерфейс.

8. Управляющая программа

Управляющая программа выполнена в системе Borland C++ Builder под операционную систему Windows-95. Для хранения имен сессий и серий данных используется база данных DBase формата с SQL-доступом через механизм BDE Engine. Запуск вычислительных программ осуществляется функциями WinAPI.

Управляющая программа позволяет пользователю в удобном виде выбирать серии для обработки, назначать варианты редукций и оценивания. Она также содержит в качестве своих составляющих систему

добавления исходных данных из файлов форматов Mark3-DBH и NGS и графическую программу удаления скачков часов и плохих наблюдений.

Управляющая программа формирует управляющие файлы для вычислительных программ, запускает их и дожидается окончания их работы. Для просмотра выходных файлов она снабжена простейшим многооконным текстовым редактором.

Помимо генерируемых C++ Builder модулей, управляющая программа включает модули, перечисленные в таблице 6.

Таблица 6. Модули управляющей программы.

Названия файлов.	Назначение.	Используется.
directory.h, directory.cpp	Хранение путей к различным директориям пакета.	Большинством модулей пакета.
fortran.h, fortran.cpp	Процедуры формирования управляющих файлов и запуска вычислительных блоков.	
matrix.h, matrix.cpp	Библиотека матричной алгебры	Системой графической чистки
redacc.h, redacc.cpp	Чтение файлов редуций и невязок, запись результатов в файлы коррекций, математические алгоритмы чистки наблюдений.	
scchart.h, scchart.cpp	Система построение графиков.	
qdb.h, qdb.cpp	Система доступа к файлам QDB формата.	Модулем read.
mk3proc.h, mk3proc.cpp	Система декодировки файлов Mark3-DBH формата.	
read.h, read.cpp	Система доступа к базе исходных данных, декодировка файлов формата NGS, дополнение базы исходных данных новыми сериями и внесение изменений в файлы коррекций.	Модулем главного окна и модулем redacc

Для хранения собственных данных управляющая программа имеет файл `quasar.dat` с набором директориев, находящийся в директории `spp`. Этот файл читается перед началом работы программы и сохраняется после внесения изменений в систему директориев. Если при запуске программы файл не обнаруживается (это происходит при первом запуске после инсталляции), он генерируется автоматически, с установкой имен директориев по умолчанию. Инструкция пользователя для управляющей программы и всех ее компонентов (фактически, инструкция эксплуатации пакета) приводится ниже.

9. Утилита `SetSeriesPath`

В процессе работы с пакетом может возникнуть необходимость перенести его в другой директорий (на другой диск), перенести готовые файлы коррекций на другой компьютер с другим экземпляром пакета. Сам пакет переносится просто. После перенесения необходимо лишь указать в диалоговом окне директориев новые директории. Несколько сложнее происходит работа при изменении путей к файлам наблюдений, в том числе при перенесении файлов коррекций на другой компьютер, где файлы наблюдений тех же серий данных находятся в других директориях. Пути к этим файлам хранятся в файлах коррекций и должны быть соответствующим образом изменены. Это и делает данная утилита. Утилита просматривает все имеющиеся файлы коррекций и находит среди них экземпляры с неправильными путями, выдает их список на экран. Пользователь указывает директории, в которых могут находиться файлы наблюдений, утилита проверяет директории и подправляет те файлы коррекций, файлы наблюдений которых найдены, и т.д. до нахождения всех файлов наблюдений. Утилита выполнена в системе Borland C++ Builder под операционную систему Windows 95 и находится в директории `spp`.

10. Формат управляющих файлов

Файлы, управляющие вычислительными программами пакета, перечислены в таблице 7.

Для новых методов оценивания возможно появление новых управляющих файлов.

Таблица 7. Управляющие файлы пакета.

Имя файла.	Читающие файл вычислительные программы.	Содержание файла.
path.dat	Все программы.	Пути к поддиректориям пакета.
task.dat		Перечень обрабатываемых серий, дополнительные данные.
reduct.dat	Редукции — Average, reduct.	Параметры редукций.
statsour.dat		Перечень обрабатываемых станций и источников.
der.dat	Все программы оценивания.	Перечень оцениваемых параметров.
signals.dat	Программы оценивания методами СКК и фильтрации Калмана — FormS, lsc, lsc1, lsc2, okf, okf1, okf2; программа covarmod .	Априорные автоковариационные функции сигналов.
regular.dat	Программы оценивания общих и суточных параметров.	Параметры регуляризации.
parunits.dat	Программы формирования выходных файлов.	Единицы измерения параметров.
extract.dat	Программа all .	Набор исключаемых параметров.
covarmod.dat	Программа covarmod .	Режимы оценивания автоковариационных функций сигналов.

Все управляющие файлы являются текстовыми и хранятся в директории `for` (директории вычислительных программ). Файлы `signals.dat`, `regular.dat` и `parunits.dat` содержат априорную информацию и не меняются в процессе работы пакета. Они могут быть изменены вручную. Файлы `der.dat` и `reduct.dat` читаются управляющей программой. Из первого извлекаются имена и типы параметров, режимы оценивания перезаписываются. Во втором меняются только значения параметров редукиций. Остальные файлы формируются управляющей программой. Ниже приводятся форматы файлов и их примеры; если файлы велики и имеют однообразную структуру, приводятся только их фрагменты.

10.1. Файл `path.dat`

Файл хранит имена путей к корневому директории пакета, содержащему поддиректории `ephem`, `catalog` и `correct`, директорию для записи результатов и директорию для хранения промежуточных файлов.

Пример файла:

```
c:\quasar
c:\quasar\results
c:\quasar\temp
```

10.2. Файл `task.dat`

Файл хранит количество и имена обрабатываемых серий данных и Флаги: производить/не производить оценки сигналов (0 или 1), производить/не производить перевзвешивание (0 или 1), записывать/не записывать матрицы взаимных корреляций (0 или 1), производить/не производить регуляризацию (0 или 1), а также что выбирать при оценивании: вектор задержек (0), вектор суммы сигналов и невязок (1) либо вектор невязок (2).

Пример файла:

```
1 Signals
1 Reweighting
0 Corr\_matrix
1 Regularization
0 Loc data variant

2 series
```

951009xs
960205xs

10.3. Файл `reduct.dat`

Файл содержит параметры редукций. В разделе редукций приводятся все параметры и пример файла. Ниже приводится лишь его фрагмент. Собственно параметр хранится в строке, начинающейся на I (целочисленный параметр), L (булевский) либо R (вещественный). Далее после пробела идет значение параметра. Остальные, начинающиеся на \$ строки рассматриваются как комментарии. Они игнорируются вычислительными программами, лишь выводятся на экран управляющей программой как неотредактируемый текст (в объект типа TLabel).

Количество целочисленных, булевских и вещественных параметров редукций, а также порядок их расположения не меняется без изменения программ `reduct` и `average`. Пример файла:

```
I 2
$   iLagr - Lagrange interpolation
$       D = 2
L 1
$   bCon - Coplanar constraints:
$       D = .TRUE. - they are used,
$       = .FALSE. - they are not used.
R 1.000000
$   TimInt - time interval for delay-rate calculation (sec):
$       D = 1.d0.
```

10.4. Файл `statsour.dat`

Файл содержит список обрабатываемых станций и источников, обрабатываемых программой `reduct`.

Пример файла:

```
2 stations
FORTLEZA
KOKEE

2 sources
0048-097
0059+581
```

10.5. Файл der.dat

Все строки файла имеют одинаковый формат. Ниже приводится пример строки:

```
3 CLCK-off 0 0 1 2 1
```

Первое число обозначает тип параметра: 1 — глобальный, 2 — источника, 3 — станции. Далее идет 8-символьный идентификатор. Следующие 5 чисел определяют режим его оценивания. Первое из них может быть 1 или 0, 1 означает, что будет оцениваться общий компонент параметра; следующее число означает степень полиномиального тренда параметра. Аналогично 3-е и 4-е числа определяют оценивание суточного компонента. Последнее число определяет оценивание сигнала (1 или 0).

10.6. Файл signals.dat

Ниже приводится пример файла:

```
5 signals total
```

	UT1-UTC UT1	POLE-x Xpole	POLE-y Ypole	ZEN-wet Wet	CLCK-off Clock
Dispersion	4.102549	0.130315	0.092268	115.331276	775.916234
Terms numb.	2	2	2	2	2
a	0.316032	0.106881	0.111100	0.813924	0.611438
alpha	3.610135	1.598910	0.111774	3.715716	3.545348
beta	6.093829	8.347187	7.172491	6.423095	5.017424
phi	0.534831	0.189258	0.015583	0.524455	0.615146
gamma	-0.592425	-0.191551	-0.015584	-0.578493	-0.706607
a	0.683968	0.893119	0.888900	0.186076	0.388562
alpha	5.725795	4.032097	2.710487	15.333129	4.107233
beta	13.057626	13.306094	11.086930	9.424778	6.789797
phi	0.327142	0.291363	0.239773	1.019662	0.544023
gamma	-0.339335	-0.299898	-0.244476	-1.626896	-0.604912

Каждый сигнал представлен одним столбцом, включающим (сверху вниз): название сигнала длинное и короткое 5-символьное, дисперсия в ², количество слагаемых модели корреляционной функции и

по 4 числа для каждого слагаемого — последовательно a , α в $^{-1}$, β в /, ϕ в радианах (см. [3]).

10.7. Файл `regular.dat`

Файл содержит величины СКО для ПВЗ по данным IERS, которые используются для вычисления параметра регуляризации. Значения СКО различны в разные периоды, которые и указываются в этом файле. Параметры, для которых нет данных в этом файле или они не полные, не регуляризируются.

Пример файла:

```
Apriori RMS of VLBI parameters for regularization
Periods of time: 1984 1995    1996 1999    2000 2001
POLE-x    mas        0.700    0.200    0.200
POLE-y    mas        0.700    0.200    0.200
```

Величины даются в единицах измерения, типичных для данного параметра и совпадающих с единицами файла `parunits.dat`. Количество располагающихся по столбцам периодов действия не ограничено.

10.8. Файл `parunits.dat`

Все строки файла имеют одинаковый формат. Ниже приводится пример строки:

```
POLE-x    mas    32.339352814180000
```

После идентификатора параметра идут его общепринятая размерность и коэффициент пересчета в эту размерность из метров. Отсутствие параметра означает, что его единицы измерения — метры.

10.9. Файл `extract.dat`

В этом файле присутствуют четыре секции, разделенные знаком @. В первой секции идут исключаемые параметры, во второй — станции, в третьей — источники, в четвертой — скорости станций. Например:

```
GLOBL-h
GLOBL-l
@
```

```

FORTLEZA
KOKEE
@
0048-097
0059+581
@
GILCREEK

```

10.10. Файл covarmod.dat

В этом файле для каждого сигнала каждой серии указывается режим обработки: 0 — берутся априорные данные, 1 — оцениваются только СКО, 2 — оцениваются и СКО и автокорреляционные функции. В файле приводится полный список станций, набор глобальных сигналов и сигналов для каждой станции. Сигналы источников в настоящее время программа covarmod не обрабатывает.

Пример файла:

```

7 stations
ALGOPARK FORTLEZA GILCREEK KOKEE NRA020 NYALES20 WETTZELL
3 global signals
UT1-UTC POLE-x POLE-y
2 station signals
ZEN-wet CLCK-off
3 series
970114xe 2 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1
970121xe 2 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1
970128xe 2 2 2 2 1 2 1 2 1 2 1 2 1 2 1 2 1

```

11. Формат файлов эфемерид

Эфемериды хранятся в текстовых файлах в поддиректории `ephem`. Доступ к файлу `tai-utc.dat`, хранящему разности TAI-UTC, осуществляется обычным последовательным способом. К остальным файлам идет прямой доступ. Поэтому все они должны состоять из строк постоянной длины. В этих строках должны содержаться время и характеристика для данного файла информация, приведенная в таблице 8. Формат файла и длина его строк прописываются в дополнительном "заголовочном" файле `table.dat`, пример которого приводится:

```

eop-iers 78 (11X,F5.0,2F9.5,F10.6,F12.6,F12.5,F9.5)

```

```

eop-neos 188 (7X,F8.2,3X,2(F9.6,10X),2X,F10.7,11X,F7.4,1X,2(10X,F9.3))
earth    80 (F9.1,3F12.0,3F11.6)
moon     47 (F9.1,3F12.3)
sun      47 (F9.1,3F12.0)
jupiter  47 (F9.1,3F12.0)
saturn   50 (F9.1,3F13.0)

```

Таблица 8. Файлы эфемерид.

Название файла.	Назначение.
eop-iers.dat	ПВЗ — опорный ряд IERS
eop-neos.dat	ПВЗ — опорный ряд NEOS
earth.dat	Эфемериды Земли
Moon.dat	Эфемериды Луны
Sun.dat	Эфемериды Солнца
Jupiter.dat	Эфемериды Юпитера
Saturn.dat	Эфемериды Сатурна

12. Формат файлов каталогов

Каталоги координат станций, источников и некоторая другая априорная информация хранятся в поддиректории `catalog` в файлах, перечисленных в таблице 9.

Для многокомпонентной модели структурной задержки радиосточника данные хранятся в отдельном для каждого радиосточника файле в поддиректории `struct` директория `catalog`. Все файлы являются текстовыми. Формат всех файлов кроме `sta_voc.dat` и `sour_voc.dat` разработан IERS и не менялся, за исключением файла `thermal.dat`. Там для обозначения конца данных (начала комментариев) введена строка `@END`. Файлы `sta_voc.dat` и `sour_voc.dat` имеют оригинальный формат. Они имеют шапку (3 строки файла `sour_voc.dat`, 4 — файла `sta_voc.dat`), столбцы данных. Файл `sta_voc.dat` имеет в конце комментарии, отделенные от строк данных строкой `@END`.

Таблица 9. Файлы каталогов.

Название файла.	Назначение, хранящиеся данные
sour_voc.dat	Псевдонимы имен радиоисточников (VLBI des.)
source.dat	Координаты радиоисточников.
Struct.dat \Struct*.mod	Структурные задержки радиоисточников Структурные задержки радиоисточников для многокомпонентной модели
sta_voc.dat	9-символьные идентификаторы станций (DOMES NB.), коэффициенты атмосферных нагрузок
Station.dat	Координаты и скорости станций.
Ocload.dat	Океанические нагрузки станций.
Rebound.dat	Постледниковые поднятия станций.
Thermal.dat	Температурные деформации станций.

Пример файла sour_voc.dat:

```
-----
ICRF Designation   IERS Des.  VLBI Des.
-----
J165352.2+394536   1652+398   DA426
J164029.6+394646   1638+398   NRA0512
J033930.9-014635   0336-019   STA26
```

Пример файла sta_voc.dat:

```
-----
DOMES NB.  ITRF NAME          NGS NAME  CorOff CoATL  Ref Fix/
           [mm]      [mm/mb]    mob
-----
14201S004  WETTZELL             WETTZELL  -04.20 -0.372 *   f
40408S002  FAIRBANKS           GILCREEK  +21.00 -0.393 *
31906M001  SAO MIGUEL          AZORES    +00.00 +0.000
10503M002  METSAHOVI          METSHOVI  +00.00 -0.303 *
@END
```

13. Формат выходных файлов

Все выходные файлы, кроме файлов матрицы взаимных корреляций, текстовые. Директорий, куда они попадают, задает пользователь; по умолчанию это поддиректорий **results** корневого директория паке-та. В таблицах 10 и 11 перечислены все виды выходных файлов.

Под датой и моментом времени понимаются модифицированная юлианская дата и время в долях суток. Векторы **W**, **S**, **R** содержат суммы оценок невязок и сигнальной части, только сигнальной части и только невязок, соответственно. Поправки общих и суточных параметров печатаются в метрах и специфических единицах измерения, сигналы, невязки, RMS сигналов, SIG, RMS, WRMS невязок — в миллиметрах, СНІ (значение χ^2 невязок) — безразмерная величина. Имена станций, источников и сигналов в начале файла оценок невязок записаны по восемь в строке. Ниже приводятся примеры файлов в виде фрагментов, т.к. эти файлы достаточно велики. Пропущенная часть обозначается строкой точек.

Пример файла оценок общих параметров.

```
50455.75226852 beginning
51906.74777778 ending
      208 series
----- Parameters -----

Global parameters

Source parameters
  1 0119+115 SOUR-ras      .0001  .0007 m      .002  .014 0.1ms
  2 0119+115 SOUR-dec     .0018  .0009 m      .057  .028 mas
  3 0804+499 SOUR-ras     .0018  .0005 m      .040  .012 0.1ms
  4 0804+499 SOUR-dec    -.0004  .0006 m     -.012  .021 mas
. . . . .
468 1045-188 SOUR-dec    -.0063  .0085 m     -.204  .274 mas
469 2021+614 SOUR-ras     .0104  .0075 m      .223  .162 0.1ms
470 2021+614 SOUR-dec     .0236  .0066 m      .763  .212 mas

Station parameters
471 FORTLEZA ANT-off     -.0092  .0006 m
472 FORTLEZA STAT-x     -.0002  .0002 m
473 FORTLEZA STAT-y     -.0006  .0001 m
474 FORTLEZA STAT-z     -.0010  .0001 m
```

Таблица 10. Выходные файлы обработки одной серии данных.

Название и назначение.	Содержание.	Чем создается.
Файл оценок суточных параметров *.sng.	Номера версии и итерации; оценки суточных параметров; дата и время первого и последнего наблюдений и момента, на который приводятся величины трендов; количество наблюдений, станций, источников, оцениваемых суточных параметров; RMS, WRMS, CHI, MSIG невязок наблюдений; RMS невязок отдельно по каждой базе; средние арифметические и RMS сигналов.	Различными программами оценивания.
Файл оценок невязок *.sig.	Номера версии и итерации; количество простых и сложных наблюдений, суточных параметров; количество и имена сигналов глобальных, для каждой станции, для каждого источника; количество и имена станций и источников; для каждого наблюдения — имена базы станций, источника, момента времени, значения векторов W , S , R (в мм.); для каждого сложного наблюдения — момент времени и значения сигналов (в мм.); средние арифметические и RMS сигналов;	
Файл матрицы взаимных корреляций.	Матрица взаимных корреляций суточных параметров.	

Таблица 11. Выходные файлы обработки группы серий данных.

Название и назначение.	Содержание.	Чем создается.
Файл оценок общих параметров <code>all.txt</code> .	Оценки общих параметров; количество обработанных серий; дата и время первого и последнего наблюдений.	Программой <code>all</code> .
Файл сводных значений суточных параметров <code>erp_sng.txt</code> .	Для каждой серии: имя серии и ее юлианская дата; количество станций, наблюдений и суточных параметров; SIG RMS WRMS CHI невязок. Для каждого глобального параметра — значение его оценки и ошибки по каждой серии, итоговые значения по всем сериям — средние арифметические и СКО глобальных параметров.	Программой <code>cvrt</code> .
Файл сводных значений сигналов <code>rms_sng.txt</code> .	Для каждой серии: имя серии и ее юлианская дата; номер итерации; RMS всех сигналов.	

475	NRA020	ANT-off	-.0049	.0004	m
476	NRA020	STAT-x	.0001	.0001	m
477	NRA020	STAT-y	-.0005	.0001	m
478	NRA020	STAT-z	.0002	.0001	m

Пример файла оценок суточных параметров.

```

-----
0 version
0 iteration
1590 observations
30 parameters
5 stations
38 sources
49367.7711806 first observation
49368.7631944 last observation
49368.2671875 mean observation

----- Parameters -----

Common parameters
1 NUT-psi -.0051 .0056 m -.165 .180 mas
2 NUT-eps -.0075 .0024 m -.242 .076 mas
3 UT1-UTC -.0837 .0013 m -1.800 .028 0.1ms
4 POLE-x -.0033 .0023 m -.108 .075 mas
5 POLE-y -.0055 .0023 m -.178 .074 mas

Source parameters

Station parameters
6 GILCREEK ZEN-wet .0003 .0009 m
7 GILCREEK ZEN-wet *t -.0007 .0083 m
8 GILCREEK CLCK-off 504.4701 .0021 m
9 GILCREEK CLCK-off*t -3.2064 .0073 m
10 GILCREEK CLCK-off*t**2 -.0168 .0206 m
. . . . .
26 FD-VLBA ZEN-wet .0511 .0009 m
27 FD-VLBA ZEN-wet *t -.0086 .0084 m
28 FD-VLBA CLCK-off 487.5720 .0024 m
29 FD-VLBA CLCK-off*t 2.4549 .0077 m
30 FD-VLBA CLCK-off*t**2 -.0465 .0219 m

```

----- RMS -----

Observations	W	S	R	Stat1	Stat2
1590	20.9	20.1	3.4		
108	17.5	17.0	2.7	GILCREEK	ONSALA60
170	14.7	13.6	4.8	GILCREEK	WESTFORD
.					
203	24.2	23.7	3.2	FD-VLBA	WESTFORD
119	31.5	30.3	4.2	FD-VLBA	WETTZELL
.					
WRMS			1.8		
CHI			.7		
MSIG			5.1		

----- Sk -----

	--- Global -----			GILCREEK		ONSALA60		WESTFORD	
	UT1	Xpole	Ypole	Wet	Clock	Wet	Clock	Wet	Clock
MO	-.3	-.1	.1	.6	-.4	-.4	.0	.4	-.1
SKO	1.3	1.4	2.4	3.8	4.3	2.8	6.1	3.4	4.5

Пример файла оценок невязок.

```
0 version
0 iteration
1590 observations
467 complex observations
30 parameters
3 global signals
UT1-UTC UT1 POLE-x Xpole POLE-y Ypole
0 source signals

2 station signals
ZEN-wet Wet CLCK-off Clock
38 sources
1357+769 0823+033 1606+106 0229+131 1749+096 0048-097 1741-038 0119+041
1308+326 1219+044 2145+067 0J287 NRA0512 0059+581 0820+560 0552+398
```

1921-293 4C39.25 2255-282 2234+282 1739+522 1622-253 0804+499 0454-234
 1128+385 0727-115 0528+134 1053+815 0735+178 0955+476 OK290 0954+658
 1726+455 0458-020 0919-260 1334-127 1034-293 1044+719

5 stations

GILCREEK ONSALA60 WESTFORD WETTZELL FD-VLBA

----- 0-C -----

	Time	W	S	R	Stat1	Stat2	Sour	
1	49367.7711806	-17.0	-16.7	-.3	GILCREEK	ONSALA60	1357+769	1
2	49367.7711806	4.0	-3.0	7.0	GILCREEK	WESTFORD	1357+769	2
3	49367.7711806	9.2	13.8	-4.6	GILCREEK	WETTZELL	1357+769	3
.								
1589	49368.7631944	-26.3	-26.2	-.1	FD-VLBA	WETTZELL	1726+455	3448
1590	49368.7631944	-24.6	-22.7	-2.0	FD-VLBA	GILCREEK	1726+455	3453

----- Sk -----

--- Global ----- GILCREEK ONSALA60 WESTFORD										
	Time	UT1	Xpole	Ypole	Wet	Clock	Wet	Clock	Wet	Clock
1	49367.7711806	.0	.6	-.5	1.8	6.8	-.4	-6.7	-.8	7.2
2	49367.7743287	.0	.6	-.4	3.0	7.2	-1.0	-6.9	-.5	7.3
3	49367.7753935	-.1	.6	-.4	3.5	7.3	-1.7	-7.0	-.3	7.2
.										
466	49368.7610417	1.8	-.5	-4.9	4.3	-6.1	-.5	6.2	-3.5	-6.0
467	49368.7631944	1.8	-.5	-4.9	3.7	-6.2	-.5	6.1	-3.5	-6.1

--- Global ----- GILCREEK ONSALA60 WESTFORD										
	MO	UT1	Xpole	Ypole	Wet	Clock	Wet	Clock	Wet	Clock
	MO	-.3	-.1	.1	.6	-.4	-.4	.0	.4	-.1
	SKO	1.3	1.4	2.4	3.8	4.3	2.8	6.1	3.4	4.5

Пример файла сводных значений оценок суточных параметров.

DATE	MJD	I	S	NOBS	PAR	SIG	RMS	WRMS	CHI	NUT-psi mas	NUT-eps mas		
970114	50463	0	5	1256	30	14.1	9.0	5.2	.56	-1.656	.075		
970121	50470	0	6	1247	35	12.4	8.9	5.0	.59	-1.535	.081		
970128	50477	0	5	936	30	16.2	12.6	6.5	.66	-1.674	.089		
MEAN WEIGHTED										-1.620	.061	.429	.067

Пример файла сводных значений оценок невязок.

DATE	MJD	I	UT1	KOKEE				NRA020		NYALES20	
				Xpole	Ypole	Wet	Clock	Wet	Clock	Wet	Clock
970114	50463	0	1.3	1.7	1.2	9.1	7.5	6.9	4.6	3.4	14.1
970121	50470	0	1.4	1.5	1.1	22.8	8.0	5.2	6.1	.0	.0
970128	50477	0	1.1	1.2	2.2	9.7	5.5	5.1	5.1	2.8	9.1

Файл матрицы взаимных корреляций — двоичный и имеет структуру, показанную в таблице 12. Используется общая процедура чтения/записи матриц, и поэтому, хотя $N = M$, хранятся обе эти величины.

Таблица 12. Структура файла матрицы взаимных корреляций.

Тип данного.	Число байт.	Назначение.
Integer*4	4	Число строк N .
Integer*4	4	Число столбцов M .
Real*8	$8 * N * M$	Собственно матрица.

14. Формат временных файлов

Временные файлы хранятся в директории временных файлов (при установке — поддиректорий temp корневого каталога пакета) и обеспечивают передачу данных от одних вычислительных программ к другим. Типы временных файлов приведены в таблице 13.

Файлы `average.txt` и `*.cov` текстовый, остальные - двоичные.

Файлы, для которых указано только расширение, в качестве имени имеют имя серии. Процедуры чтения/записи двоичных файлов на языке FORTRAN находятся в файле `accdata.for`. На C++ из двоичных временных файлов читается лишь файл редукций — конструктором класса TSeries файла `readred.cpp`. Файл `average.txt` формируется головной процедурой программы `average` и читается процедурой `SetAverage` файла `loadcat.for`. Файлы `*.cov` формируются процедурой `ProcSeries` файла `procseries.cpp` и читаются процедурой `LoadSignals` файла `sortpar.for`. Все временные файлы имеют жесткую структуру.

Таблица 13. Типы временных файлов.

Наименование.	Имя или расширение.	Создающая программа.	Используемые программы.
Файл опорных значений температур, давлений, влажностей станций.	average.txt.	average	reduct
Файл редукций.	*.red	reduct	Программы оценивания и графической чистки
Файл матрицы стохастических параметров СКК-оценивания.	*.s	FormS FormS	Программы СКК-оценивания
Файл оценок ковариационных функций.	*.cov	covarmod	
Файл списка параметров СКК-оценивания.	param.bin	Программами первого шага	Программы второго шага, addall, add
Файл нормальных систем общих и суточных параметров.	*.c11		
Файл оценок общих параметров	allmatr.bin	addall	all
Файл оценок общих параметров	all.bin	all	Программы второго шага

14.1. Двоичные файлы

Двоичные файлы имеют содержание, показанное в таблицах 14-16, (типы данных приведены в FORTRAN-стиле). В файлах редукций хранящиеся номера баз вычисляются по формуле: $i_{base} = (i_1 - 1)iStat - \frac{(i_1 - 1)i_1}{2} - i_1 + i_2$, где $iStat$ — количество станций в серии, i_1, i_2 — номера станций базы; если $i_1 > i_2$, то в формуле они меняются местами. Оригинальные номера наблюдений — это номера соответствующих наблюдений в базе исходных наблюдений (как правило для редукций выбирается лишь часть наблюдений, их непосредственные номера уже не совпадают с оригинальными).

Все двоичные промежуточные файлы, кроме файла редукций, неполные — важные для чтения/записи переменные в них не хранятся и определяются до операций чтения/записи:

- в файле `*.s` величины N — количество наблюдений — и iU — количество столбцов в сжатой матрице частных производных сигналов;
- в файле `param.bin` величины $iXcm$ — количество общих, $iXsr$ — суточных параметров.
- в файле `*.cls` величины N , iXc и iYc в файле не хранятся и должны быть определены до его чтения/записи.

В файле `param.bin` списки параметров хранятся в сжатом виде — наборы параметров глобальных, каждого источника, каждой станции. Если оценивается тренд m -й степени — присутствуют все члены полинома от 0-вой степени до m -й.

Файл `all.bin` содержит полные списки станций, источников и параметров.

Во всех файлах матрица частных производных хранится в сжатом виде по строкам.

14.2. Текстовые промежуточные файлы

Файл `average.txt` содержит три столбца данных — средние значения температуры, давления и влажности. Ниже приведен пример файла:

1	DSS45	18.897	941.305	60.314
2	FORTLEZA	29.200	1007.158	86.330
3	HARTRAO	22.170	863.343	65.721

Таблица 14. Содержание файла редукций.

Тип данного.	Число байт.	Назначение.
Integer*2	2	Номер версии.
Integer*4	4	Модифицированная юлианская дата.
Real*8	8	Момент времени (в долях суток) первого наблюдения.
Real*8	8	Момент времени (в долях суток) последнего наблюдения.
Integer*4	4	Количество станций $iStat$.
Character*8	$8 * iStat$	Имена станций.
Integer*4	4	Количество источников $iSou$.
Character*8	$8 * iSou$	Имена источников.
Integer*4	4	Количество параметров iX .
Character*8	$8 * iX$	Имена параметров.
Integer*4	4	Количество глобальных параметров.
Integer*4	4	Количество параметров источников.
Integer*4	4	Количество параметров каждой станции.
Integer*4	4	Количество наблюдений N .
Integer*2	$2 * 4 * N$	Массив, для каждого наблюдения хранящий номера наблюдавших станций (первые два числа), наблюдавшегося источника (третье число) и номер базы (четвертое число).
Integer*2	$2 * N$	Массив оригинальных номеров наблюдений.
Real*8	$8 * N$	Моменты времени наблюдений
Real*8	$8 * N$	Ошибки наблюдений.
Real*8	$8 * N$	Вектор О-С
Real*8	$8 * iX * N$	Матрица частных производных.

Таблица 15. Содержание файлов *.s, param.bin и allmatr.bin.

Тип данного.	Число байт.	Назначение.
Файлы *.s.		
Real*8	$8 * N * N/2$	Матрица Q .
Real*8	$8 * N * iU$	Матрица частных производных сигналов.
Файл param.bin.		
Character*8	$8 * iX_{cm}$	Список общих параметров.
Character*8	$8 * iX_{sr}$	Список суточных параметров.
Integer*4	$4 * iX_{cm}$	Степени трендов общих параметров.
Integer*4	$4 * iX_{sr}$	Степени трендов суточных параметров.
Файл allmatr.bin.		
Integer*4	4	Количество источников $iSour$.
Character*8	$8 * iSour$	Имена источников.
Integer*4	4	Количество станций $iStat$.
Character*8	$8 * iStat$	Имена станций.
Integer*4	4	Количество общих параметров iX .
Character*8	$8 * iX$	Идентификаторы параметров.
Integer*4	$4 * iX$	Тип параметров: 1-глобальные, 2-источников, 3-станций.
Integer*4	$4 * iX$	Степень тренда.
Integer*4	$4 * iX$	Номер станции или источника.
Real*8	$iX * iX$	Матрица нормальной системы.
Real*8	iX	Вектор нормальной системы.

Таблица 16. Содержание файлов *.cls и all.bin.

Тип данного.	Число байт.	Назначение.
Файлы *.cls.		
Integer*4	4	Несжатое количество общих параметров iX .
Integer*4	4	Несжатое количество суточных параметров iY .
Integer*4	4	Несжатое количество суточных параметров с учетом ограничений $iY1$.
Real*8	$8 * 2$	Числа $T0, T1$ пересчета моментов времени для расчета трендов.
Real*8	$8 * iXc * N$	Матрица частных производных общих параметров A_k .
Real*8	$8 * iYc * N$	Матрица частных производных суточных параметров B_k .
Real*8	$8 * iX * iX$	Матрица нормальной системы общих параметров P_k .
Real*8	$8 * iX$	Вектор нормальной системы общих параметров p_k .
Real*8	$8 * iY1 * iY1$	Матрица нормальной системы суточных параметров G_k .
Real*8	$8 * iX * iY1$	Матрица связи общих и суточных параметров H_k .
Real*8	$8 * iY1$	Вектор нормальной системы суточных параметров g_k .
Файл all.bin.		
Integer*4	4	Количество станций $iStat$.
Character*8	$8 * iStat$	Имена станций.
Integer*4	4	Количество источников $iSour$.
Character*8	$8 * iSour$	Имена источников.
Integer*4	4	Количество параметров iX .
Real*8	$8 * iX$	Значения оценок параметров.

Файл оценок оценок ковариационных функций *.cov содержит следующие данные:

- номер итерации;
- общее количество сигналов;
- количество и имена глобальных сигналов, сигналов источников и станций;
- количество и имена источников и станций;
- RMS сигналов;
- параметры корреляционных функций сигналов;

Пример файла приведен ниже:

```

1 iteration
13 signals total
96 points total
0.020833 interval between points
  3 global signals
UT1-UTC  UT1  POLE-x  Xpole POLE-y  Ypole
  0 source signals

  2 station signals
ZEN-wet  Wet  CLCK-off Clock
  28 sources
0537-441 1144-379 1610-771 0637-752 4C39.25 1954-388 0727-115 1622-253
0208-512 0048-097 STA26    0528+134 0229+131 2145+067 2255-282 1741-038
2128-123 0454-234 1606+106 1034-293 1308+326 0823+033 1749+096 1124-186
2052-474 1219+044 0458-020 1104-445

  2 stations
HOBART26 KOKEE
      8      8      8      4      4      4      4
      0.500  0.500  0.500  1.000  1.000  1.000  1.000
      2.000  2.000  2.000  6.240  2.640  6.240  2.640
      6.283  6.283  6.283  6.283  9.865  6.283  9.865
      0.000  0.000  0.000  0.840  0.260  0.840  0.260
      0.500  0.500  0.500  0.000  0.000  0.000  0.000
      2.000  2.000  2.000  0.000  0.000  0.000  0.000
      12.566 12.566 12.566  0.000  0.000  0.000  0.000
      0.000  0.000  0.000  0.000  0.000  0.000  0.000

```

Параметры корреляционных функций записываются аналогично записи в файле `signals.dat` (см. "Формат управляющих файлов").

В заключение выражаю благодарность проф. В. С. Губанову за постановку задачи, полезные дискуссии о конструкции и функциональной схеме пакета QUASAR, участие в написании некоторых процедур и их отладке, Ю. Л. Русинову за участие в тестировании пакета и подготовке публикации, И. А. Козловой за участие в тестировании выполненных на C++ систем сортировки данных.

Работа выполнена при частичной финансовой поддержке Российского фонда фундаментальных исследований по проекту 00-02-16690.

Список литературы

- [1] Суркис И. Ф. Структура файлов "MARK-3 DBH" и системы их декодировки. Сообщения ИПА РАН, 104, 1997.
- [2] Губанов В. С., Суркис И. Ф. Обработка РСДБ-наблюдений: Программный пакет QUASAR. I. Редукция данных наблюдений. Сообщения ИПА РАН, 141, 2001.
- [3] Губанов В. С., Козлова И. А., Суркис И. Ф. Обработка РСДБ-наблюдений: Программный пакет QUASAR. II. Методы анализа данных. Сообщения ИПА РАН, 142, 2001.
- [4] O.Titov and N. Zarraoa. OCCAM 3.4. User's Guide. Сообщения ИПА РАН, 69, 1997.

И. Ф. Суркис

Обработка РСДБ-наблюдений: Программный пакет QUASAR. III. Структура и схема функционирования.

Оригинал-макет подготовлен с помощью системы **Л^AT_EX**

Подписано к печати 2.04.2002 Формат 60 × 90/16. Офсетная печать. Печ.л. 4.3
Уч.-изд.л. 4.3 Тираж 100 Заказ бесплатно

Отпечатано в типографии ПИЯФ РАН
(188350 Ленинградская обл., г. Гатчина, Орлова роща).

Институт прикладной астрономии РАН, 197110, С.-Петербург, Ждановская ул., 8.