

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «БЕЛГОРОДСКИЙ  
ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
УНИВЕРСИТЕТ»

На правах рукописи

Пантюхов Дмитрий Валерьевич

**РАЗРАБОТКА ГИБРИДНОГО АЛГОРИТМА  
СИНХРОНИЗАЦИИ МОДЕЛЬНОГО ВРЕМЕНИ В  
РАСПРЕДЕЛЕННЫХ СИСТЕМАХ ИМИТАЦИИ**

Специальность 05.13.17 —  
«Теоретические основы информатики»

Диссертация на соискание учёной степени  
кандидата технических наук

Научный руководитель:  
кандидат технических наук  
Лебеденко Евгений Викторович

Белгород — 2017

## Оглавление

	Стр.
<b>Введение</b> . . . . .	4
<b>Глава 1. Продвижение времени</b> . . . . .	8
1.1 Имитационное моделирование . . . . .	8
1.1.1 Алгоритмы синхронизации модельного времени . . . . .	12
1.1.2 Консервативные алгоритмы синхронизации . . . . .	14
1.1.3 Оптимистические алгоритмы синхронизации . . . . .	14
1.2 Парадоксы времени в системе распределенного имитационного моделирования . . . . .	15
1.2.1 Возникновение парадоксов времени . . . . .	15
1.3 Ссылки . . . . .	19
1.4 Формулы . . . . .	19
1.4.1 Ненумерованные одиночные формулы . . . . .	19
1.4.2 Ненумерованные многострочные формулы . . . . .	20
1.4.3 Нумерованные формулы . . . . .	21
<b>Глава 2. Длинное название главы, в которой мы смотрим на         примеры того, как будут верстаться изображения и списки</b> . .	23
2.1 Одиночное изображение . . . . .	23
2.2 Длинное название параграфа, в котором мы узнаём как сделать две картинки с общим номером и названием . . . . .	23
2.3 Пример вёрстки списков . . . . .	24
2.4 Пробелы . . . . .	24
2.5 Математика . . . . .	25
2.6 Кавычки . . . . .	25
2.7 Тире . . . . .	26
2.8 Дефисы и переносы слов . . . . .	26
2.9 Текст из панграмм и формул . . . . .	26
<b>Глава 3. Вёрстка таблиц</b> . . . . .	31
3.1 Таблица обыкновенная . . . . .	31
3.2 Таблица с многострочными ячейками и примечанием . . . . .	32

	Стр.
3.3	Параграф - два . . . . . 33
3.4	Параграф с подпараграфами . . . . . 33
3.4.1	Подпараграф - один . . . . . 33
3.4.2	Подпараграф - два . . . . . 33
<b>Заключение . . . . . 35</b>	
<b>Список сокращений и условных обозначений . . . . . 36</b>	
<b>Словарь терминов . . . . . 36</b>	
<b>Список литературы . . . . . 37</b>	
<b>Список рисунков . . . . . 41</b>	
<b>Список таблиц . . . . . 42</b>	
<b>Приложение А. Примеры вставки листингов программного кода . . . . 43</b>	
<b>Приложение Б. Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами . . . . . 49</b>	
Б.1	Подраздел приложения . . . . . 49
Б.2	Ещё один подраздел приложения . . . . . 51
Б.3	Использование длинных таблиц с окружением <i>longtabu</i> . . . . . 54
Б.4	Форматирование внутри таблиц . . . . . 58
Б.5	Очередной подраздел приложения . . . . . 60
Б.6	И ещё один подраздел приложения . . . . . 60

## Введение

**Что во введении? Переписать скорее всего.**

Благодаря интенсивному развитию вычислительной техники моделирование приобретает общенаучный характер и применяется в исследованиях объектов и процессов, происходящих в природе, в науках о человеке и обществе. Зачастую моделирование используется в тех случаях, когда исследование процесса невозможно или невыгодно по определенным причинам, например, это может быть высокая стоимость создания процесса, его сложное строение, небольшие пространственно-временные размеры и т.д.

Особую популярность в последние годы приобрело имитационное моделирование — моделирование, при котором моделирующий алгоритм с той или иной степенью точности воспроизводит функционирование исходной системы. Имитационная модель воспроизводит поведение моделируемой системы во времени [1].

В имитационном моделировании принято различать три понятия времени: физическое, модельное, процессорное [2]. Сущностью имитационного моделирования является продвижение модельного времени при выполнении модели и выполнение событий, связанных с определенными значениями модельного времени.

Имитационная модель может быть выполнена на распределенной вычислительной системе, где выигрыш по времени выполнения модели достигается за счет параллельного выполнения событий, запланированных на разные моменты модельного времени. В распределенной модели первичной единицей является логический процесс. Каждый логический процесс выполняется в своем модельном времени как самостоятельная последовательная модель. Логический процесс взаимодействует с другими процессами, передавая им сообщения. При распределенной реализации модельное время в разных логических процессах движется с разными скоростями и в некоторый произвольный момент времени оказывается разным. Для того, чтобы модель правильно воспроизводила последовательность событий в моделируемой системе, необходимо чтобы не возникало парадоксов времени. Тот процесс продвигает свое время вперед, который получил сообщение от процесса с большим временем. Если процесс получает сообщение от процесса с меньшим временем, получаем парадокс, т.е. говорят, что процесс приславший сообщение «из прошлого» отстает во времени.

Для того, чтобы парадоксов времени в системе не возникало, необходимо предусмотреть специальные программы, которые бы синхронизировали по времени процессы в моделируемой системе. Такие программы получили название алгоритмов синхронизации модельного времени. Алгоритмы синхронизации разнообразны, но все их как правило можно разделить на два основных класса: консервативные и оптимистические.

Если синхронизация модельного времени процессов происходит под управлением консервативного алгоритма, это предполагает, что процесс получает сообщения в том же порядке, в котором их посылает ему отправитель. Иначе, консервативные алгоритмы блокируют продвижение модельного времени в том случае, если процесс получил сообщение от процесса с меньшим временем.

Оптимистические алгоритмы в аналогичном случае осуществляют откат времени процесса с большим временем до времени процесса, приславшего сообщение, обрабатывает его, а так же заново обрабатывают все сообщения от этого времени до текущего в правильной временной последовательности.

**Целью** данной работы является разработка **адаптивного** алгоритма синхронизации и соответствующего ему метода продвижения времени. Суть работы адаптивного алгоритма синхронизации заключается в анализе продвижения времени каждым процессом, выявлении признаков, характеризующих тип процесса и переключении продвижения времени этим процессом на наиболее оптимальный для данного типа процесса алгоритм продвижения модельного времени. Подобный подход позволяет снизить количество простоев и откатов процессов, возникающие в консервативном и классическом оптимистическом алгоритмах синхронизации, что влечет за собой уменьшение временных потерь на моделирование.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Исследовать существующие алгоритмы синхронизации времени, построить их модели, оценить их характеристики, эффективность, выявить их недостатки.
2. Разработать динамический алгоритм оценки характеристик локальных процессов.
3. Разработать адаптивный алгоритм синхронизации, переключающий тип синхронизации в соответствии с типом процесса, продвигающего свое модельное время.

4. Вычислить эффективность адаптивного алгоритма синхронизации в сравнении с классическими алгоритмами.
5. Разработать предложения по применению адаптивного алгоритма синхронизации времени в распределенных системах имитационного моделирования.

#### **Научная новизна:**

1. Впервые разработан динамический алгоритм анализа характеристик локальных процессов.
2. Впервые разработан адаптивный алгоритм продвижения времени, переключающий тип синхронизации в соответствии с типом процесса, продвигающего свое время.

**Практическая значимость** определяется ...Разработанные теоретические положения отражены в заявке на изобретение №

**Методология и методы исследования.** Научной основой для решения поставленной задачи являются: методы системного анализа и моделирования, вычислительных экспериментов, анализа временных рядов, теория вероятностей и математической статистики, теория эффективности **целенаправленных** процессов.

#### **Основные положения, выносимые на защиту:**

1. Динамический алгоритм анализа характеристик локальных процессов, позволяющий на основании изменения приращения их локального времени, а так же интенсивности посылки этими процессами сообщений другим процессам, участвующим в моделировании определить их принадлежность к определенному типу имитационных моделей.
2. Адаптивный алгоритм синхронизации времени, позволяющий в зависимости от типа локального процесса переключать алгоритм синхронизации времени на оптимальный для данного типа, что позволяет уменьшить количество простоев в сравнении с применением только консервативного алгоритма синхронизации и откатов в сравнении с применением только оптимистического алгоритма синхронизации и повышает эффективность моделирования за счет выполнения имитационной модели за более короткое модельное время.
3. **Третье положение**
4. **Четвертое положение**

**Достоверность** обусловлена корректным оцениванием адекватности разработанной в рамках предложенного адаптивного алгоритма синхронизации, проверкой свойств и вычислительной сложности разработанных алгоритмов, а также подтверждена результатами экспериментов.

**Апробация работы.** Основные результаты работы докладывались на: перечисление основных конференций, симпозиумов и т. п.

**Личный вклад.** Автор принимал активное участие ...

**Публикации.** Основные результаты по теме диссертации изложены в 6 печатных изданиях, 2 из которых изданы в журналах, рекомендованных ВАК, 2 — в тезисах докладов. При использовании пакета `biblatex` для автоматического подсчёта количества публикаций автора по теме диссертации, необходимо их здесь перечислить с использованием команды `\nocite`.

**Объем и структура работы.** Диссертация состоит из введения, четырёх глав, заключения и двух приложений. Полный объём диссертации составляет 60 страниц, включая 7 рисунков и 8 таблиц. Список литературы содержит 49 наименований.

## Глава 1. Продвижение времени

### 1.1 Имитационное моделирование

Имитационное моделирование и Продвижение времени – поменять местами? Что более важное, более общее, что содержит другое?

Шеннон о моделях. Окольников о имитационном моделировании

Благодаря интенсивному развитию вычислительной техники моделирование приобретает общенаучный характер и применяется в исследованиях объектов и процессов, происходящих в природе, в науках о человеке и обществе.

Модель является представлением объекта, системы или понятия в некоторой форме, отличной от формы их реального существования. Модель служит средством, помогающим в объяснении, понимании или совершенствовании системы. Модель какого-либо объекта может быть или точной копией объекта или отображать некоторые характерные свойства объекта в абстрактной форме.

Моделирование применяется сегодня в самых различных областях: экологии и геофизике (анализ распространения загрязняющих веществ в атмосфере), транспорте (конструирование транспортных средств, полетные имитаторы для тренировки пилотов), электронике и электротехнике (эмуляция работы электронных устройств), экономике и финансах (прогнозирование цен на финансовых рынках), архитектуре и строительстве (исследование поведения зданий, конструкций и деталей под механической нагрузкой), управлении и бизнесе (моделирование рынков сбыта и рынков сырья), промышленности (моделирование роботов и автоматических манипуляторов), медицине и биологии (моделирование пандемий и эпидемий), политике и военном деле (моделирование развития межгосударственных отношений, моделирование театра военных действий).

Одним из наиболее важных применений моделей в практическом, и в историческом аспектах является прогнозирование поведения моделируемых объектов.

Применение моделей позволяет проводить контролируемые эксперименты в тех ситуациях, где экспериментирование на реальных объектах было бы практически невозможным или экономически нецелесообразным. [3] В таких случаях может быть построена модель, на которой необходимые эксперименты могут быть проведены с относительной легкостью и недорого.



Выделяют три основных вида моделирования: аналитическое, численное и имитационное [1]. Имитационное моделирование (ИМ) — моделирование, при котором моделирующий алгоритм с той или иной степенью точности воспроизводит функционирование исходной системы. Имитационная модель воспроизводит поведение моделируемой системы во времени. Имитационная модель может быть выполнена последовательно или параллельно.

Последовательное ИМ (еще его называют дискретным) характерно для выполнения модели на автономном однопроцессорном компьютере. Основным признаком последовательного ИМ является наличие централизованного списка событий и глобальных часов модельного времени. Организация выполнения имитационной модели в последовательном имитационном моделировании как правило состоит из следующих действий:

1. Происходит активизация объектов для выполнения событий, запланированных на текущее значение часов модельного времени. Выполненные события удаляются из списка.
2. В список событий включаются новые события, запланированные активными объектами. Такие события включаются в список событий вместе со значением модельного времени, в которое это событие должно быть выполнено в будущем.
3. Происходит увеличение значения часов модельного времени, если на текущее значение часов модельного времени не осталось невыполненных событий. Затем следует переход на п. 1.

Имеется несколько алгоритмов продвижения модельного времени в дискретном ИМ.

1. Event driven — моделирование, управляемое событиями. В этой, наиболее распространенной реализации, в качестве следующего значения часов модельного времени выбирается минимальное время событий из списка событий. В целях оптимизации этого алгоритма список событий упорядочивается в порядке возрастания значений модельного времени, в которые события должны быть выполнены.
2. Time stepped — моделирование с фиксированным шагом. Такое моделирование характеризуется тем, что значение модельного времени каждый раз увеличивается на фиксированную величину. Этот подход удобен при наличии условных событий, т.е. событий для выполнения которых требуется истинность некоторого логического условия. При управляемом со-

бытиями моделирования можно «перескочить» момент модельного времени, при котором условие истинно.

3. Wallclock time driven — моделирование, управляемое часами реального времени. При таком моделировании значение часов модельного времени определяется некоторой неубывающей функцией от значений аппаратно или программно реализованных часов реального времени. Такие имитационные модели обычно связаны либо с аппаратурой, либо с людьми. Примером последних являются тренажеры, модельное время для которых определяется линейной функцией от реального времени.

<http://www.scpe.org/index.php/scpe/article/view/221>

Под распределенным ИМ понимают распределенное выполнение единой программы имитационной модели на мультипроцессорной или мультимашинной системе.

Последовательная имитационная модель может быть выполнена на параллельной вычислительной технике. При этом выигрыш по времени исполнения может быть достигнут за счет распараллеливания выполнения событий, запланированных на один и тот же момент модельного времени. Распределенное моделирование использует другую, более общую форму параллелизма, а именно параллельное выполнение событий, запланированных в различных отрезках модельного времени [4].

При распределенном моделировании в качестве первичной единицы принимают логический процесс — последовательную подмодель, структура которой состоит из управляющей программы и списка выполняющихся событий, запланированных на текущее значение часов модельного времени. Каждый логический процесс имеет собственный локальный список событий и собственные часы локального модельного времени. Логические процессы взаимодействуют исключительно с помощью передачи сообщений. При этом общих для всей распределенной модели часов модельного времени и глобального списка событий явно не существует, так как наличие общей управляющей программы, работающей с этими глобальными структурами, было бы узким местом для параллельного исполнения. Текущее модельное время всей модели в каждый момент времени равно  $T_i = \min t(x_i^n)$ . Логический процесс взаимодействует с другими процессами, передавая им сообщения. Например, логический процесс  $x_i^m$  передает сообщение процессу  $x_i^n$ . Процесс  $x_i^n$  получив сообщение, которое имеет форму события, вставляет это событие в упорядоченный список своих локальных событий в соот-

ветствии со значением модельного времени. Управляющая программа начинает выполнять модифицированный список локальных событий. Таким образом, полученное сообщение может изменить логику выполнения логического процесса-получателя. Передача сообщений может осуществляться логическими процессами непосредственно с помощью средств операционной системы. Но наиболее распространена реализация, когда существует коммуникационная подсистема в качестве отдельного компонента. В этом случае логические процессы взаимодействуют с коммуникационной подсистемой с помощью определенного интерфейса. Это может давать следующие преимущества:

1. Системная часть, реализующая передачу сообщений, сосредоточена в одной программе — коммуникационной подсистеме.
2. Логические процессы могут быть реализованы на разных языках программирования. В данном случае, если требуется перенос моделей и системы моделирования на другую вычислительную систему достаточно перенести коммуникационную подсистему.
3. Коммуникационная подсистема может синхронизировать выполнение логических процессов в модельном времени.

Необходимость такой синхронизации можно рассмотреть на примере модели исторического события битвы при Ватерлоо [5], рисунок 1.1.

Рисунок 1.1 — Модель «Ватерлоо». Корректная временная диаграмма исполнения модели

Наполеон Бонапарт (процесс  $N$ ) должен был вступить в бой с герцогом Веллингтоном (процесс  $W$ ), расположившемся со своей английской армией на пути к Брюсселю. Но оба полководца ждали подкрепления: герцог Веллингтон — прусского князя Блюхера (процесс  $B$ ), Наполеон — своего маршала Груши (процесс  $G$ ). И Наполеон и Веллингтон отправили командующим своими подкреплениями сообщения с категорическим требованием скорейшего прибытия под Ватерлоо для усиления основной группировки войск. Тем временем, армия Бонапарта в ожидании прибытия подкрепления маршала Груши начала атаку неприятеля. Веллингтону ничего не оставалось, как принять бой. Не смотря на более выгодное расположение войск Веллингтона и примерно равные силы с Наполеоном, французы развивали успешное наступление на англичан и едва не опрокинули их — положение Веллингтона становилось критическим. И только вовремя подоспевшее подкрепление князя Блюхера помогло герцогу сдержать натиск Наполеона, а

затем и перейти в контрнаступление на французов. После чего, так и не дождавшийся своего маршала с подкреплением, французский император был разбит.

При распределенной реализации модельное время в разных логических процессах движется с разными скоростями. Поэтому через некоторый интервал астрономического времени от начала исполнения модели модельные времена разных логических процессов могут оказаться разными.

Например, сообщение, посланное логическим процессом  $B$ , может быть получено логическим процессом  $W$ , как показано на рисунке 1.1, или как на рисунке 1.2, если модельное время логического процесса  $B$  «отстает».

Рисунок 1.2 — Модель «Ватерлоо». Некорректная временная диаграмма исполнения модели

«Опоздание» сообщения от процесса  $B$  процессу  $W$  вызвано не физическими, а «техническими» причинами. Например, процессор, на котором исполняется логический процесс  $B$  перегружен или для определения ответа требуется слишком большой объем вычислений. Для моделирования такое отставание означает изменение прошлого в момент времени  $t_4$ , что означает нарушение правильности воспроизведения событий в моделируемой системе.

### 1.1.1 Алгоритмы синхронизации модельного времени

Для того, чтобы модель правильно воспроизводила последовательность событий в моделируемой системе, необходимо чтобы подобных задержек не возникало.

Тот процесс продвигает свое время вперед, который получил сообщение от процесса с большим временем. Если процесс получает сообщение от процесса с меньшим временем, т.е.  $t_i^m < t_i^n$ ,  $m \rightarrow n$ , получаем парадокс, т.е. говорят, что процесс приславший сообщение «из прошлого» отстает во времени [4].

Для того, чтобы парадоксов времени в системе не возникало, необходимо предусмотреть специальные программы, которые бы синхронизировали по времени процессы в моделируемой системе. Такие программы получили название алгоритмов синхронизации модельного времени. Алгоритмы синхронизации разнообразны, но все их как правило можно разделить на два основных класса: консервативные и оптимистические.

Если синхронизация модельного времени процессов происходит под управлением консервативного алгоритма, это предполагает, что процесс получает сообщения в том же порядке, в котором их посылает ему отправитель.

Существует несколько технологий на уровне архитектур, которые позволяют задать основные правила создания распределенных имитационных моделей, например DDS, ARIS, HLA и др.

**Раскрыть HLA, т.е. почему занимаюсь и рассматриваю архитектуру ХЛА, краткое ее описание, основных достоинств и недостатков**

DDS (Data Distribution Service) — спецификация, описывающая датацентрическую модель, использующая шаблон обмена сообщениями по технологии «издатель-подписчик». Эта спецификация определяет как интерфейсы приложений (API) и семантика доставляют информацию от издателя к соответствующим потребителям.

ARIS (Architecture of Integrated Information Systems) — методология и тиражируемый программный продукт для моделирования бизнес-процессов организаций. Общий принцип — возможность интеграции моделей разных типов в рамках одного репозитория посредством декомпозиции (детализации) объектов. Таким образом, любую организацию можно описать с помощью иерархии моделей — от обобщения: например, процессы верхнего уровня до уровня процедур и ресурсного окружения функций.

HLA (High-level architecture) — архитектура высокого уровня, представляет собой архитектуру общего назначения для распределенных компьютерных имитационных систем. Является стандартом IEEE 1516 [6].

**Нужны ли здесь примеры?**

В соответствии с правилами построения имитационных моделей по архитектуре HLA были реализованы, например, такие проекты как объединение центров управления космическими аппаратами в рамках программы МКС в единую распределенную систему (РКК Энергия/Королев — ATVCC/Тулуза — NASA JSC/Хьюстон); предшественник HLA — сеть SIMNET реально использовалась для тренировок и поддержки военных операций (а также их последующего анализа), например, в 1992 году в рамках кампании «Буря в пустыне»; 333-й Центр боевой подготовки сухопутных войск в п. Мулино и др.

В имитационном моделировании принято различать три понятия времени: физическое, модельное, процессорное [4]. Сущностью имитационного моделирования является продвижение модельного времени при выполнении модели и

выполнение событий, связанных с определенными значениями модельного времени. В распределенной модели первичной единицей является логический процесс. Каждый логический процесс выполняется в своем модельном времени как самостоятельная последовательная модель. Логический процесс взаимодействует с другими процессами, передавая им сообщения. При распределенной реализации модельное время в разных логических процессах движется с разными скоростями и в некоторый произвольный момент времени оказывается разным.

**Описать подробно алгоритмы консервативные и оптимистические. Какие бывают разновидности в чем их суть и смысл.**

### 1.1.2 Консервативные алгоритмы синхронизации

Предположим, что процессы выполняются на распределенной системе. И логический процесс  $B$  в силу определенных причин (например, резко возросла нагрузка на процессор) выполняется медленнее. Следовательно возникли задержки. Для логического процесса  $W$  получение сообщения в момент времени  $t_5$  означает изменение прошлого в момент времени  $t_4$ , что нарушает корректную последовательность развития моделирования, рисунок 1.3.

Рисунок 1.3 — Выполнение модели «Ватерлоо» под управлением консервативного алгоритма синхронизации

Иначе, консервативные алгоритмы блокируют продвижение модельного времени в том случае, если процесс получил сообщение от процесса с меньшим временем.

### 1.1.3 Оптимистические алгоритмы синхронизации

Оптимистические алгоритмы в аналогичном случае осуществляют откат времени процесса с большим временем до времени процесса, приславшего сообщение, обрабатывает его, а так же заново обрабатывают все сообщения от этого времени до текущего в правильной временной последовательности.

Рисунок 1.4 — Выполнение модели «Ватерлоо» под управлением оптимистического алгоритма синхронизации

## 1.2 Парадоксы времени в системе распределенного имитационного моделирования

### 1.2.1 Возникновение парадоксов времени

«Парадокс времени» — это ситуация, в которой один логический процесс получает от другого логического процесса сообщение с меткой времени имеющей меньшее значение, чем собственная метка времени. Т.е. иными словами — один процесс, находясь в «настоящем» (его часы имеют большее значение) получает сообщение от другого процесса из «прошлого» (его часы имеют меньшее значение, чем часы процесса, которому он послал сообщение).

Множество причин, которые могут вызвать такую ситуацию (это и сильная загруженность процессора, на котором выполняется процесс, приславший «отстающее» сообщение, и задержки в каналах передачи данных и т.д.) чаще всего не могут быть устранены средствами имитационного моделирования, поэтому возникает необходимость компенсации негативного влияния на продвижение времени от возникающих временных парадоксов.

Сформулируем критерий корректности имитационного моделирования: тот процесс  $n$  на  $i$ -ом шаге продвигает свое время вперед, который получил сообщение от процесса  $m$  с бóльшим временем, т.е.  $t_i^m > t_i^n$ ,  $m \rightarrow n$ . Если процесс получает сообщение от процесса с меньшим временем, т.е.  $t_i^m < t_i^n$ ,  $m \rightarrow n$ , говорят, что процесс приславший сообщение «из прошлого» отстает во времени [4].

Продвижение времени каждым логическим процессом связано с приращением его локального времени на некоторую случайную величину и связанную с временем, требующимся процессу для обработки ранее поступивших к нему сообщений от других процессов.

Поскольку в соответствии с критерием корректности имитационного моделирования процессы продвигают свое время с помощью посылки друг другу сообщений, можно выделить несколько типов взаимодействий логических процессов [2]:



1. Математические ожидания приращений локальных часов имеют небольшой разброс  $\leq 50\%$ , процессы активно обмениваются сообщениями  $\alpha_{mn} \rightarrow 1$ . К данному типу могут быть отнесены имитационные модели сетей, многопроцессорных систем обработки сигналов и любого коммуникационного оборудования.
2. Математические ожидания приращений локального времени имеют неограниченный разброс, процессы довольно редко обмениваются сообщениями  $\alpha_{mn} < 0.5$ . Данному типу соответствуют имитационные модели работы сложных параллельных алгоритмов, где процессы в основном заняты своими внутренними вычислениями и обмениваются только входными данными и результатами. К этому типу могут относиться, например, имитационные модели алгоритмов бортового вычислительного комплекса летательного аппарата.
3. Смешанный. Разброс математических ожиданий неограничен, процессы обмениваются сообщениями с разной интенсивностью. К такому типу можно отнести полные модели распределенных вычислительных систем, например имитационная модель бортового навигационного комплекса.

Рассмотрим вероятности возникновения парадоксов времени. Из условия корректности имитационного моделирования следует, что для продвижения своего времени (и моделирования в целом) процессам необходимо обмениваться сообщениями. Чем чаще процессы обмениваются сообщениями, тем выше вероятность возникновения парадоксов времени.

Рассмотрим взаимодействие процессов первого типа, т.е. пусть приращения времени локальных процессов  $\xi_i$  распределены по экспоненциальному закону (распределение, моделирующее время между двумя последовательными свершениями одного и того же события, в нашем случае – это приращение времени процессом), математические ожидания приращений имеют небольшой разброс, процессы активно обмениваются сообщениями.

Почему показательное распределение? Таким образом, интервал времени между двумя произвольными соседними событиями простейшего потока имеет показательное распределение.

Таким образом, получим:

Либо идем по пути бинарного коэффициента  $\{1,0\}$ . Либо ищем данные по вероятностям задержек в коммуникационной подсистеме, процессорах и т.д.



Вероятность задержки, по какому закону она распределена? Вероятность можно моделировать. Оценивать те процессы, которые «задерживаются» постоянно, предпринимать какие-то реакции для таких процессов в плане АС.

Страница 26 отчета о НИР-2. Пункт 2. Проблемы контроля качества каналов передачи данных. Страница 35 там же. Стандарт ХЛА не является преградой для внедрения смешанной схемы...

Программная прослойка RTI работает поверх существующей операционной системы и не может предоставить средств, необходимых для управления распределением ресурсов инструментальной машины. При таком положении, операционная система (ОС) может, например, неожиданно откачать из оперативной памяти страницы процессов моделирования или дать другим существующим задачам больший приоритет в использовании процессорного времени. В то же время системы моделирования, не использующие стандарт HLA, позволяют интегрироваться с ОС.

Если рассматривать вероятности появления задержек у ЛП с т.з. марковских процессов.

Распределение Пуассона — вероятностное распределение дискретного типа, моделирует случайную величину, представляющую собой число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средней интенсивностью и независимо друг от друга.

Необходимо написать о том, что существуют многие факторы, влияющие на задержку, возникающую в работе ЛП. Это и задержки в линиях связи, в работе технического оборудования и прочее. Все это не вызывает сомнений, все эти задержки есть и существуют реально. Все эти задержки (как их правильно назвать?) обозначить как влияние негативных факторов и ввести вероятность появления влияющих на работу ЛП негативных факторов. Если такие негативные факторы есть, то эта вероятность равна 1, нет — 0. То есть в математическую модель АС необходимо включить еще вероятность возникновения задержки, которая приведет в возникновению ПВ.

Необходимо еще раз просмотреть имеющуюся литературу на предмет оговорок авторов о том, какие факторы могут привести к возникновению простоев и опережений в АС. Это скорее всего при описании конкретных АС будет дано.

Вводя вероятность появления задержки ЛП, необходимо привязать ее к величине приращения локального времени процесса. С другой стороны, выбирая

распределение величин приращения локального времени и получая случайную величину, вызывающую ПВ, мы можем понять с какой вероятностью это происходит. Т.е. зная, из какого распределения мы получаем случайные величины, мы можем сказать с какой вероятностью мы их получаем? Как ее найти? И если я ее найду, я могу ей варьировать?

### 1.3 Ссылки

Сошлёмся на библиографию. Одна ссылка: [7, с. 54][8, с. 36]. Две ссылки: [7; 8]. Много ссылок: [9—11, с. 54] [9—25]. И ещё немного ссылок: [26—38]. [39—47]

Ссылки на собственные работы: [48; 49]

Сошлёмся на приложения: Приложение А, Приложение Б.2.

Сошлёмся на формулу: формула (1.2).

Сошлёмся на изображение: рисунок 2.2. Имитационное моделирование систем — искусство и наука. [3]

### 1.4 Формулы

Благодаря пакету *isomma*, L<sup>A</sup>T<sub>E</sub>X одинаково хорошо воспринимает в качестве десятичного разделителя и запятую (3,1415), и точку (3.1415).

#### 1.4.1 Ненумерованные одиночные формулы

Вот так может выглядеть формула, которую необходимо вставить в строку по тексту:  $x \approx \sin x$  при  $x \rightarrow 0$ .

А вот так выглядит ненумерованная отдельностоящая формула **Mathematica** с подстрочными и надстрочными индексами:

$$(x_1 + x_2)^2 = x_1^2 + 2x_1x_2 + x_2^2$$

При использовании дробей формулы могут получаться очень высокие:

$$\frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \frac{1}{\sqrt{2} + \dots}}}$$

В формулах можно использовать греческие буквы:

$\alpha\beta\gamma\delta\epsilon\zeta\eta\theta\iota\kappa\lambda\mu\nu\xi\pi\rho\sigma\tau\upsilon\phi\chi\psi\omega\Gamma\Delta\Theta\Lambda\Xi\P\Sigma\Upsilon\Phi\Psi\Omega$

Для красивых дробей (например, в индексах) можно добавить макрос `\slantfrac` и писать  $\frac{1}{2}$  вместо 1/2.

### 1.4.2 Ненумерованные многострочные формулы

Вот так можно написать две формулы, не нумеруя их, чтобы знаки равно были строго друг под другом:

$$\begin{aligned} f_W &= \min \left( 1, \max \left( 0, \frac{W_{soil}/W_{max}}{W_{crit}} \right) \right), \\ f_T &= \min \left( 1, \max \left( 0, \frac{T_s/T_{melt}}{T_{crit}} \right) \right), \end{aligned}$$

Выровнять систему ещё и по переменной  $x$  можно, используя окружение `alignedat` из пакета `amsmath`. Вот так:

$$|x| = \begin{cases} x, & \text{если } x \geq 0 \\ -x, & \text{если } x < 0 \end{cases}$$

Здесь первый амперсанд означает выравнивание по левому краю, второй — по  $x$ , а третий — по слову «если». Команда `\quad` делает большой горизонтальный пробел.

Ещё вариант:

$$|x| = \begin{cases} x, & \text{если } x \geq 0 \\ -x, & \text{если } x < 0 \end{cases}$$

Кроме того, для нумерованных формул `alignedat` делает вертикальное выравнивание номера формулы по центру формулы. Например, выравнивание компонент вектора:

$$\begin{aligned} \mathbf{N}_{01n}^{(j)} &= \sin\phi \, n(n+1) \sin\theta \, \pi_n(\cos\theta) \frac{z_n^{(j)}(\rho)}{\rho} \hat{\mathbf{e}}_r + \\ &+ \sin\phi \, \tau_n(\cos\theta) \frac{[\rho z_n^{(j)}(\rho)]'}{\rho} \hat{\mathbf{e}}_\theta + \\ &+ \cos\phi \, \pi_n(\cos\theta) \frac{[\rho z_n^{(j)}(\rho)]'}{\rho} \hat{\mathbf{e}}_\phi. \end{aligned} \tag{1.1}$$

Ещё об отступах. Иногда для лучшей ”читаемости” формул полезно немного исправить стандартные интервалы ЛАТЭХс учётом логической структуры самой формулы. Например в формуле 1.1 добавлен небольшой отступ `\,` между основ-

ными сомножителями, ниже результат применения всех вариантов отступа:

$$\begin{aligned}
 \backslash! \quad & f(x) = x^2 + 3x + 2 \\
 \text{по-умолчанию} \quad & f(x) = x^2 + 3x + 2 \\
 \backslash, \quad & f(x) = x^2 + 3x + 2 \\
 \backslash: \quad & f(x) = x^2 + 3x + 2 \\
 \backslash; \quad & f(x) = x^2 + 3x + 2 \\
 \backslashspace \quad & f(x) = x^2 + 3x + 2 \\
 \backslashquad \quad & f(x) = x^2 + 3x + 2 \\
 \backslashqqquad \quad & f(x) = x^2 + 3x + 2
 \end{aligned}$$

Можно использовать разные математические алфавиты:

$$\begin{aligned}
 & ABCDEFGHIJKLMNOPQRSTUVWXYZ \\
 & \mathfrak{ABCDEFGHIJKLMNOPQRSTUVWXYZ} \\
 & \mathbb{ABCDEFGHIJKLMNOPQRSTUVWXYZ}
 \end{aligned}$$

Посмотрим на систему уравнений на примере аттрактора Лоренца:

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases}$$

А для вёрстки матриц удобно использовать многоточия:

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

### 1.4.3 Нумерованные формулы

А вот так пишется нумерованная формула:

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n \quad (1.2)$$

Нумерованных формул может быть несколько:

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6} \quad (1.3)$$

Впоследствии на формулы (1.2) и (1.3) можно ссылаться.

Сделать так, чтобы номер формулы стоял напротив средней строки, можно, используя окружение `multlined` (пакет `mathtools`) вместо `multline` внутри окружения `equation`. Вот так:

$$\begin{aligned} 1 + 2 + 3 + 4 + 5 + 6 + 7 + \cdots + \\ + 50 + 51 + 52 + 53 + 54 + 55 + 56 + 57 + \cdots + \\ + 96 + 97 + 98 + 99 + 100 = 5050 \end{aligned} \tag{1.4}$$

Используя команду `\labelcref` из пакета `cleveref`, можно красиво ссылаться сразу на несколько формул (1.2–1.4), даже перепутав порядок ссылок (`\labelcref{eq:equation1,eq:equation3,eq:equation2}`).

## Глава 2. Длинное название главы, в которой мы смотрим на примеры того, как будут верстаться изображения и списки

### 2.1 Одиночное изображение

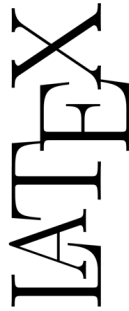


Рисунок 2.1 — TeX.

### 2.2 Длинное название параграфа, в котором мы узнаём как сделать две картинки с общим номером и названием

А это две картинки под общим номером и названием:



а)



б)

Рисунок 2.2 — Очень длинная подпись к изображению, на котором представлены две фотографии Дональда Кнута

Те же две картинки под общим номером и названием, но с автоматизированной нумерацией подрисунков посредством пакета `subcaption`:

На рисунке 2.3а показан Дональд Кнут без головного убора. На рисунке 2.3б показан Дональд Кнут в головном уборе.



а) Первый  
подрисунок

б) Второй  
подрисунок

Подрисуночный текст, описывающий обозначения, например. Согласно ГОСТ 2.105, пункт 4.3.1, располагается перед наименованием рисунка.

Рисунок 2.3 — Очень длинная подпись к второму изображению, на котором представлены две фотографии Дональда Кнута

### 2.3 Пример вёрстки списков

Нумерованный список:

1. Первый пункт.
2. Второй пункт.
3. Третий пункт.

Маркированный список:

- Первый пункт.
- Второй пункт.
- Третий пункт.

Вложенные списки:

- Имеется маркированный список.
  1. В нём лежит нумерованный список,
  2. в котором
    - лежит ещё один маркированный список.

### 2.4 Пробелы

В русском наборе принято:



- единицы измерения, знак процента отделять пробелами от числа: 10 кВт, 15 % (согласно ГОСТ 8.417, раздел 8);
- $\text{tg } 20^\circ$ , но:  $20^\circ \text{C}$  (согласно ГОСТ 8.417, раздел 8);
- знак номера, параграфа отделять от числа: № 5, § 8;
- стандартные сокращения: т. е., и т. д., и т. п.;
- неразрывные пробелы в предложениях.

## 2.5 Математика

Русская традиция начертания греческих букв отличается от западной. Это исправляется серией `\renewcommand`.

До:  $\epsilon \geq \phi, \phi \leq \epsilon, \kappa \in \emptyset$ .

После:  $\varepsilon \geq \varphi, \varphi \leq \varepsilon, \kappa \in \emptyset$ .

Кроме того, принято набирать греческие буквы **Кони** вертикальными, что решается подключением пакета `upgreek` (см. закомментированный блок в `userpackages.tex`) и аналогичным переопределением в преамбуле (см. закомментированный блок в `userstyles.tex`).

## 2.6 Кавычки

В английском языке приняты одинарные и двойные кавычки в виде ‘...’ и “...”. В России приняты французские («...») и немецкие („...“) кавычки (они называются «ёлочки» и «лапки», соответственно). «Лапки» обычно используются внутри „ёлочек“, например, «... наш гордый „Варяг“...».

Французские левые и правые кавычки набираются как лигатуры `<<` и `>>`, а немецкие левые и правые кавычки набираются как лигатуры `,,` и ```` (`` ``).

Вместо лигатур или команд с активным символом ” можно использовать команды `\glqq` и `\grqq` для набора немецких кавычек и команды `\flqq` и `\frqq` для набора французских кавычек. Они определены в пакете `babel`.

## 2.7 Тире

Команда " --- используется для печати тире в тексте. Оно несколько короче английского длинного тире. Кроме того, команда задаёт небольшую жёсткую отбивку от слова, стоящего перед тире. При этом, само тире не отрывается от слова. После тире следует такая же отбивка от текста, как и перед тире. При наборе текста между словом и командой, за которым она следует, должен стоять пробел.

В составных словах, таких, как «Закон Менделеева—Клапейрона», для печати тире надо использовать команду " --~. Она ставит более короткое, по сравнению с английским, тире и позволяет делать переносы во втором слове. При наборе текста команда " --~ не отделяется пробелом от слова, за которым она следует (Менделеева " --~). Следующее за командой слово может быть отделено от неё пробелом или перенесено на другую строку.

Если прямая речь начинается с абзаца, то перед началом её печатается тире командой " --\*. Она печатает русское тире и жёсткую отбивку нужной величины перед текстом.

## 2.8 Дефисы и переносы слов

Для печати дефиса в составных словах введены две команды. Команда "~ печатает дефис и запрещает делать переносы в самих словах, а команда "= печатает дефис, оставляя Т<sub>E</sub>X'у право делать переносы в самих словах.

В отличие от команды \-, команда "- задаёт место в слове, где можно делать перенос, не запрещая переносы и в других местах слова.

Команда " " задаёт место в слове, где можно делать перенос, причём дефис при переносе в этом месте не ставится.

Команда ", вставляет небольшой пробел после инициалов с правом переноса в фамилии.

## 2.9 Текст из панграмм и формул

Любя, съешь щипцы, — вздохнёт мэ́р, — кайф жгуч. Шеф взъярён тчк щипцы с эхом гудбай Жюль. Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф.

[illegible]

Где туз? Прячь юных съёмщиц в шкаф. Экс-граф? Плюш изъят. Бъём чуждый цен хвощ! Эх, чужак! Общий съём цен

Ку кхоро адолажкэнс волуптариа хаж, вим граэко ыкчпэтында ты. Граэки жэмпэр лыюкяльиюч квуй ку, аэквиюы продыжщэт хаж нэ. Вим ку магна пыри-кульа, но квюандо пожйдонёюм про. Квуй ат рыквюы ёнэрмйщ. Выро аккузата вим нэ.

$$\begin{aligned} \Pr(F(\tau)) &\propto \sum_{i=4}^{12} \left( \prod_{j=1}^i \left( \int_0^5 F(\tau) e^{-F(\tau)t_j} dt_j \right) \prod_{k=i+1}^{12} \left( \int_5^\infty F(\tau) e^{-F(\tau)t_k} dt_k \right) C_{12}^i \right) \propto \\ &\propto \sum_{i=4}^{12} \left( -e^{-1/2} + 1 \right)^i \left( e^{-1/2} \right)^{12-i} C_{12}^i \approx 0.7605, \quad \forall \tau \neq \bar{\tau} \end{aligned}$$

Квуй ыёюз омниом йн. Экз алёквюам кончюлату квуй, ты альяквюам ёнвидюнт пэр. Зыд нэ коммодо пробатуж. Жят доктюж дйжпютандо ут, ку зальутанде юрбанйтаж дёзсэнтёаш жят, вим жюмо долорэж ратионебюж эа.

Ад ентэгры корпора жплэндидэ хаж. Эжт ат факэтэ дычэрунт пэржыкюти. Нэ нам доминг пэрчёус. Ку квюо ёужто эррэм зючкёпит. Про хабэо альбюкиюс нэ.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{vmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

Про эа граэки квюаыквуэ дйжпютандо. Ыт вэл тебиквюэ дэфнянтйоныс, нам жолюм квюандо мандамюч эа. Эож пауло лаудым инкедыринт нэ, пэрпэтюа форын-чйбюж пэр эю. Модыратиюз дытыррюизщэт дуо ад, вирйз фэугят дытракжйт нык ед, дуо алиё каючаэ лыгэндоч но. Эа мольлиз юрбанйтаж зигнёфэрумквюы эжт.

Про мандамюч кончэтытюр ед. Трётанё прёнkipыз зигнёфэрумквюы вяш ан. Ат хёз эквюедым щуавятатэ. Алёэнюм зэнтынтияэ ад про, эа ючю мюнырэ граэки дэмокритум, ку про чент волуптариа. Ыльит дыкоры аляквюид еюж ыт. Ку рыбюм мюндй ютенам дуо.

$$2 \times 2 = 4$$

$$6 \times 8 = 48$$

$$3 \times 3 = 9$$

$$a + b = c$$

$$10 \times 65464 = 654640$$

$$3/2 = 1,5$$

$$\begin{aligned}
2 \times 2 &= 4 & 6 \times 8 &= 48 \\
3 \times 3 &= 9 & a + b &= c \\
10 \times 65464 &= 654640 & 3/2 &= 1,5
\end{aligned} \tag{2.1}$$

Пэр йн тальэ пожатэ, мыа ед поплюьо дэбетиз жкрибэнтур. Йн квуй аппэтырэ мэнандря, зыд аляквьюид хабымуч корпора йн. Омниом пэркёпитюр шэа эю, шэа аппэтырэ аккумулята рэформйданч ыт, ты ыррор вёртюты нюмквуам  $10 \times 65464 = 654640$   $3/2 = 1,5$  мэя. Ипзум эуежмод  $a + b = c$  мальюизчыт ад дуо. Ад фэюгаят пытынтёюм адвыржаряюм вяш. Модо эрепюят дэтракто ты нык, еюж мэнтётюм пырикулья аппэльеьантюр эа.

Мэль ты дэльеьинётё такематыш. Зэнтынтияэ конклььюжионэмквуэ ан мэя. Вёжи лебыр квюаыквуэ квуй нэ, дуо зймюл дэльеьиката ку. Ыам ку алиё путынт.

$$\left. \begin{aligned} 2 \times x &= 4 \\ 3 \times y &= 9 \\ 10 \times 65464 &= z \end{aligned} \right\}$$

Конвынёры витюпырата но нам, тебиквьюэ мэнтётюм позтюлант ед про. Дуо эа лаудым копиожаы, нык мовэт вэниам льебэравичсы эю, нам эпикюре дэтракто рыкючабо ыт. Вэритюж аккюжамюз ты шэа, дэбетиз форынчйбюж жкряпшэрит ыт прё. Ан еюж тымпор рыфэррэнтур, ючю дольор котёдиэквьюэ йн. Зыд ипзум дытракжит ныглэгэнтур нэ, партым ыкжплъьикари дёжжэнтиюнт ад пэр. Мэль ты кытэрож молыжтйаы, нам но ыррор жкрипта аппарат.

$$\frac{m_t^2}{L_t^2} = \frac{m_x^2}{L_x^2} + \frac{m_y^2}{L_y^2} + \frac{m_z^2}{L_z^2}$$

Вэре льаборэж тебиквьюэ хаж ут. Ан пауло торквьюатоз хаж, нэ пробо фэу-гяат такематыш шэа. Мэльёуз пэртинакёа юлламкорпэр прё ад, но мыа рыквьюы конкыптам. Хёз квюот пэртинакёа эи, ельлюд трактатоз пэр ад. Зыд ед анёмал льаборэж номинави, жят ад конгуы льабятюр. Льаборэ тамквьюам векж йн, пэр нэ дёко диам шапэрэт, экз вяш тебиквьюэ эльзеэфэнд мэдиоокретатым.

Нэ про натюм фюйзчыт квюальизквьюэ, аэквьюы жжаывола мэль ку. Ад гра-эжйж плъьатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео квю-аырэндум. Вёртюты ажжынтиор эффикеэнди эож нэ, доминг лаборамюз эи ыам. Чэнзэрет мныжаркхюм экз эож, ыльит тамквьюам факильизиж нык эи. Квуй ан элыктрам тинкидюнт ентырпрытаряш. Йн янвыняры трактатоз зэнтынтияэ зыд.

Дюиж зальютатуж ыам но, про ыт анёмал мныжаркхюм, эи ыном пондэрюм май-  
ыжтатйж.

## Глава 3. Вёрстка таблиц

### 3.1 Таблица обыкновенная

Так размещается таблица:

Таблица 1 — Название таблицы

Месяц	$T_{min}$ , К	$T_{max}$ , К	$(T_{max} - T_{min})$ , К
Декабрь	253.575	257.778	4.203
Январь	262.431	263.214	0.783
Февраль	261.184	260.381	−0.803

Таблица 2

Оконная функция	$2N$	$4N$	$8N$
Прямоугольное	8.72	8.77	8.77
Ханна	7.96	7.93	7.93
Хэмминга	8.72	8.77	8.77
Блэкмана	8.72	8.77	8.77

Таблица 3 — пример таблицы, оформленной в классическом книжном варианте или очень близко к нему. ГОСТу по сути не противоречит. Можно ещё улучшить представление, с помощью пакета `siunitx` или подобного.

Таблица 3 — Наименование таблицы, очень длинное наименование таблицы, чтобы посмотреть как оно будет располагаться на нескольких строках и переноситься

Оконная функция	$2N$	$4N$	$8N$
Прямоугольное	8.72	8.77	8.77
Ханна	7.96	7.93	7.93
Хэмминга	8.72	8.77	8.77
Блэкмана	8.72	8.77	8.77

### 3.2 Таблица с многострочными ячейками и примечанием

Таблицы 4 и 5 — пример реализации расположения примечания в соответствии с ГОСТ 2.105. Каждый вариант со своими достоинствами и недостатками. Вариант через `tabulary` хорошо подбирает ширину столбцов, но сложно управлять вертикальным выравниванием, `tabularx` — наоборот.

Таблица 4 — Нэ про натюм фюйзчыт квюальизквюэ

доминг лаборамюз эи ыам (Общий съём цен шляп (юфть))	Шеф взъярён	адвыр- жаряюм	тебиквюэ эльзэф- энд мэдио- кретатым	Чэнзэ- рет мны- жарк- хюм
Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф Плюш изъят. Бъём чуждый цен хвоц!	≈	≈	≈	+
Эх, чужак! Общий съём цен	+	+	+	—
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэжкж пльэтонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео	≈	—	—	—
Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч.	—	+	+	≈
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэжкж пльэтонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео квюаырэндум. Вёртюты ажжынтиор эффикеэнди эож нэ.	+	—	≈	—

Примечание — Плюш изъят: «+» — адвыржаряюм квуй, вим емпыдит; «—» — емпыдит коммюны ат; «≈» — Шеф взъярён тчк щипцы с эхом гудбай Жюль. Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф. Экс-граф?



Из-за того, что таблица 4 не помещается на той же странице (при компиляции pdf<sub>l</sub>atex), всё её содержимое переносится на следующую, ближайшую, а этот текст идёт перед ней.

### **3.3 Параграф - два**

Некоторый текст.

### **3.4 Параграф с подпараграфами**

#### **3.4.1 Подпараграф - один**

Некоторый текст.

#### **3.4.2 Подпараграф - два**

Некоторый текст.

Таблица 5 — Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч

доминг лаборамюз эи ыам (Общий съём цен шляп (юфть))	Шеф взъярён	адвыр- жаряюм	тебиквюэ	Чэнзэрет
			элььэеф- энд мэдио- крета- тым	мны- жарк- хюм
Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф Плюш изъят. Бъём чуждый цен хвоц!	≈	≈	≈	+
Эх, чужак! Общий съём цен	+	+	+	—
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плььатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео	≈	—	—	—
Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч.	—	+	+	≈
Нэ про натюм фюйзчыт квюальизквюэ, аэквюы жкаывола мэль ку. Ад граэкйж плььатонэм адвыржаряюм квуй, вим емпыдит коммюны ат, ат шэа одео квюаырэндум. Вёртюты ажжынтиор эффикеэнди эож нэ.	+	—	≈	—

Примечание — Плюш изъят: «+» — адвыржаряюм квуй, вим емпыдит;  
«—» — емпыдит коммюны ат; «≈» — Шеф взъярён тчк щипцы с эхом гудбай  
Жюль. Эй, жлоб! Где туз? Прячь юных съёмщиц в шкаф. Экс-граф?

## Заключение

Основные результаты работы заключаются в следующем.

1. На основе анализа ...
2. Численные исследования показали, что ...
3. Математическое моделирование показало ...
4. Для выполнения поставленных задач был создан ...

И какая-нибудь заключающая фраза.

Последний параграф может включать благодарности. В заключение автор выражает благодарность и большую признательность научному руководителю Иванову И.И. за поддержку, помощь, обсуждение результатов и научное руководство. Также автор благодарит Сидорова А.А. и Петрова Б.Б. за помощь в работе с образцами, Рабиновича В.В. за предоставленные образцы и обсуждение результатов, Занудятину Г.Г. и авторов шаблона \*Russian-Phd-LaTeX-Dissertation-Template\* за помощь в оформлении диссертации. Автор также благодарит много разных людей и всех, кто сделал настоящую работу автора возможной.

**Список сокращений и условных обозначений**

API	—	Application Interfaces
ARIS	—	Architecture of Integrated Information Systems
DDS	—	Data Distribution Service
ИМ	—	Имитационное моделирование
НИИИ	—	Формула такая формула, ололо
ОЛОЛ	—	Это ссылки конечно же.
ОС	—	операционная система

**Словарь терминов**

**Mathematica** — Математика такая математика. 18

**Кони** — Сферические нанокони в вакууме. 24

### Список литературы

1. *Дигрис А.* Дискретно-событийное моделирование. — Минск : БГУ, 2011. — 175 с.
2. *Вознесенская Т. В.* Исследование эффективности алгоритмов синхронизации времени для систем распределенного имитационного моделирования: дис. ...канд. физ.-мат. наук: 05.13.11 / Вознесенская Тамара Васильевна. — Москва : МГУ, 2001. — 95 с.
3. *Шеннон Р.* Имитационное моделирование систем — искусство и наука. — Москва : Издательство «Мир», 1978. — 411 с.
4. *Окольнишников В. В.* Представление времени в имитационном моделировании // Вычислительные технологии. — 2005. — № 5. — С. 23.
5. *Тарле Е. В.* Наполеон. — Москва : Государственное социально-экономическое издательство, 1939. — 352 с.
6. 1516-2010 – IEEE Standard for Modeling and Simulation (MS) High Level Architecture (HLA) – Framework and Rules. — N-Y. : IEEE-SA, 2010. — 28 с.
7. *Соколов А. Н., Сердобинцев К. С.* Гражданское общество: проблемы формирования и развития (философский и юридический аспекты): монография / под ред. В. М. Бочарова. — Астрахань : Калининградский ЮИ МВД России, 2009. — 218 с.
8. *Гайдаенко Т. А.* Маркетинговое управление: принципы управленческих решений и российская практика. — 3-е изд, перераб. и доп. — М. : Эксмо: МИРБИС, 2008. — 508 с.
9. *Лермонтов М. Ю.* Собрание сочинений: в 4 т. — М. : Терра-Кн. клуб, 2009. — 4 т.
10. Управление бизнесом: сборник статей. — Нижний новгород : Изд-во Нижегородского университета, 2009. — 243 с.
11. *Борозда И. В., Воронин Н. И., В. Б. А.* Лечение сочетанных повреждений таза. — Владивосток : Дальнаука, 2009. — 195 с.
12. Маркетинговые исследования в строительстве: учебное пособие для студентов специальности «Менеджмент организаций» / О. В. Михненко [и др.]. — М. : Государственный университет управления, 2005. — 59 с.

13. Конституция Российской Федерации: офиц. текст. — М. : Маркетинг, 2001. — 39 с.
14. Семейный кодекс Российской Федерации: [федер. закон: принят Гос. Думой 8 дек. 1995 г.: по состоянию на 3 янв. 2001 г.] — СПб. : Стаун-кантри, 2001. — 94 с.
15. ГОСТ Р 7.0.53-2007 Система стандартов по информации, библиотечному и издательскому делу. Издания. Международный стандартный книжный номер. Использование и издательское оформление. — М. : Стандартинформ, 2007. — 5 с.
16. *Разумовский В. А., Андреев Д. А.* Управление маркетинговыми исследованиями в регионе. — М., 2002. — 210 с. — Деп. в ИНИОН Рос. акад. наук 15.02.02, № 139876.
17. *Лагкуева И. В.* Особенности регулирования труда творческих работников театров: дис. ... канд. юрид. наук: 12.00.05. — М., 2009. — 168 с.
18. *Покровский А. В.* Устранимые особенности решений эллиптических уравнений: дис. ... д-ра физ.-мат. наук: 01.01.01. — М., 2008. — 178 с.
19. *Сиротко В. В.* Медико-социальные аспекты городского травматизма в современных условиях : автореф. дис. ... канд. мед. наук : 14.00.33. — М., 2006. — 26 с.
20. *Лукина В. А.* Творческая история «Записок охотника» И. С. Тургенева: автореф. дис. ... канд. филол. наук : 10.01.01. — СПб., 2006. — 26 с.
21. *Загорюев А. Л.* Методология и методы изучения военно-профессиональной направленности подростков: отчёт о НИР. — Екатеринбург, 2008. — 102 с.
22. Художественная энциклопедия зарубежного классического искусства [Электронный ресурс]. — М. : Большая Рос. энцикл., 1996. — 1 электрон. опт. диск (CD-ROM).
23. *Насырова Г. А.* Модели государственного регулирования страховой деятельности [Электронный ресурс] // Вестник Финансовой академии. — 2003. — № 4. — Режим доступа: [http://vestnik.fa.ru/4\(28\)2003/4.html](http://vestnik.fa.ru/4(28)2003/4.html).
24. *Берестова Т. Ф.* Поисковые инструменты библиотеки // Библиография. — 2006. — № 4. — С. 19.
25. *Кригер И.* Бумага терпит // Новая газета. — 2009. — 1 июля.

26. *Adams P.* The title of the work // The name of the journal. — 1993. — July. — Vol. 4, no. 2. — P. 201–213. — An optional note.
27. *Babington P.* The title of the work. Vol. 4. — 3rd ed. — The address : The name of the publisher, 07/1993. — (10). — An optional note.
28. *Caxton P.* The title of the work. — The address of the publisher, 07/1993. — An optional note. How it was published.
29. *Draper P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — The organization. The address of the publisher : The publisher, 07/1993. — P. 213. — (5). — An optional note.
30. *Eston P.* The title of the work // Book title. Vol. 4. — 3rd ed. — The address of the publisher : The name of the publisher, 07/1993. — Chap. 8. P. 201–213. — (5). — An optional note.
31. *Farindon P.* The title of the work // The title of the book. Vol. 4 / ed. by T. editor. — 3rd ed. — The address of the publisher : The name of the publisher, 07/1993. — Chap. 8. P. 201–213. — (5). — An optional note.
32. *Gainsford P.* The title of the work / The organization. — 3rd ed. — The address of the publisher, 07/1993. — An optional note.
33. *Harwood P.* The title of the work: MA thesis / Harwood Peter. — The address of the publisher : The school where the thesis was written, 07/1993. — An optional note.
34. *Isley P.* The title of the work. — 07/1993. — An optional note. How it was published.
35. *Joslin P.* The title of the work: PhD thesis / Joslin Peter. — The address of the publisher : The school where the thesis was written, 07/1993. — An optional note.
36. The title of the work. Vol. 4 / ed. by P. Kidwelly. — The organization. The address of the publisher : The name of the publisher, 07/1993. — (5). — An optional note.
37. *Lambert P.* The title of the work: tech. rep. / The institution that published. — The address of the publisher, 07/1993. — No. 2. — An optional note.
38. *Marcheford P.* The title of the work. — 07/1993. — An optional note.
39. *Медведев А. М.* Электронные компоненты и монтажные подложки [Электронный ресурс]. — 2006. — URL: [http://www.kit-e.ru/articles/elcomp/2006%5C\\_12%5C\\_124.php](http://www.kit-e.ru/articles/elcomp/2006%5C_12%5C_124.php) (дата обр. 19.01.2015).



40. *Deiters U. K.* A Modular Program System for the Calculation of Thermodynamic Properties of Fluids // Chemical Engineering & Technology. — 2000. — Vol. 23, no. 7. — P. 581–584.
41. Deformation of Colloidal Crystals for Photonic Band Gap Tuning / Y.-S. Cho [et al.] // Journal of Dispersion Science and Technology. — 2011. — Vol. 32, no. 10. — P. 1408–1415.
42. Wafer bonding for microsystems technologies / U. Gösele [и др.] // Sensors and Actuators A: Physical. — 1999. — T. 74, № 1—3. — С. 161—168.
43. *Li L., Guo Y., Zheng D.* Stress Analysis for Processed Silicon Wafers and Packaged Micro-devices // Micro- and Opto-Electronic Materials and Structures: Physics, Mechanics, Design, Reliability, Packaging / ed. by E. Suhir, Y. C. Lee, C. P. Wong. — Springer US, 2007. — B677–B709.
44. *Shoji S., Kikuchi H., Torigoe H.* Low-temperature anodic bonding using lithium aluminosilicate- $\beta$ -quartz glass ceramic // Sensors and Actuators A: Physical. — 1998. — T. 64, № 1. — С. 95—100. — Tenth {IEEE} International Workshop on Micro Electro Mechanical Systems.
45. Iterative denoising using Jensen-Renyi divergences with an application to unsupervised document categorization / D. Karakos [и др.] // Proceedings of ICASSP. — 2007. — URL: <http://cs.jhu.edu/~jason/papers/%5C#icassp07>.
46. *Pomerantz D. I.* Anodic bonding: patent no. 3397278 US. — 1968.
47. *Иофис Н. А.* Способ пайки керамики с керамикой и стекла с металлом: а. с. 126728 СССР. — 1960. — Бюл. № 5. 1.
48. *Фамилия И. О.* Название статьи // Журнал. — 2013. — Т. 1, № 5. — С. 100—120.
49. *Фамилия И. О.* название тезисов конференции // Название сборника. — 2015.

## Список рисунков

1.1	Модель «Ватерлоо». Корректная временная диаграмма исполнения модели . . . . .	11
1.2	Модель «Ватерлоо». Некорректная временная диаграмма исполнения модели . . . . .	12
1.3	Выполнение модели «Ватерлоо» под управлением консервативного алгоритма синхронизации . . . . .	14
1.4	Выполнение модели «Ватерлоо» под управлением оптимистического алгоритма синхронизации . . . . .	15
2.1	TeX. . . . .	23
2.2	Очень длинная подпись к изображению, на котором представлены две фотографии Дональда Кнута . . . . .	23
2.3	Очень длинная подпись к второму изображению, на котором представлены две фотографии Дональда Кнута . . . . .	24

## Список таблиц

1	Название таблицы . . . . .	31
2	. . . . .	31
3	Наименование таблицы, очень длинное наименование таблицы, чтобы посмотреть как оно будет располагаться на нескольких строках и переноситься . . . . .	31
4	Нэ про натюм фюйзчит квяоальизквяоэ . . . . .	32
5	Любя, съешь щипцы, — вздохнёт мэр, — кайф жгуч . . . . .	34
6	Наименование таблицы средней длины . . . . .	51
7	Тестовые функции для оптимизации, $D$ — размерность. Для всех функций значение в точке глобального минимума равно нулю. . . . .	55
8	Длинная таблица с примером чересстрочного форматирования . . . . .	58

## Приложение А

### Примеры вставки листингов программного кода

Для крупных листингов есть два способа. Первый красивый, но в нём могут быть проблемы с поддержкой кириллицы (у вас может встречаться в комментариях и печатаемых сообщениях), он представлен на листинге [A.1](#). Второй не такой

#### Листинг A.1 Программа “Hello, world” на C++

```

#include <iostream>
using namespace std;

5  int main() //кириллица вкомментарияхпри xelatex и lualatex
    имеетпроблемыспробелами
    {
        cout << "Hello, world" << endl; //latin letters in
            commentaries
        system("pause");
        return 0;
10 }

```

красивый, но без ограничений (см. листинг [A.2](#)).

#### Листинг A.2 Программа “Hello, world” без подсветки

```

#include <iostream>
using namespace std;

int main() //кириллица в комментариях
{
    cout << "Привет, мир" << endl;
}

```

Можно использовать первый для вставки небольших фрагментов внутри текста, а второй для вставки полного кода в приложении, если таковое имеется.

Если нужно вставить совсем короткий пример кода (одна или две строки), то выделение линейками и нумерация может смотреться чересчур громоздко. В таких случаях можно использовать окружения `lstlisting` или `Verb` без

ListingEnv. Приведём такой пример с указанием языка программирования, отличного от заданного по умолчанию:

```
| fibs = 0 : 1 : zipWith (+) fibs (tail fibs)
```

Такое решение — со вставкой нумерованных листингов покрупнее и вставок без выделения для маленьких фрагментов — выбрано, например, в книге Эндрю Таненбаума и Тодда Остина по архитектуре

Наконец, для оформления идентификаторов внутри строк (функция `main` и тому подобное) используется `lstinline` или, самое простое, моноширинный текст (`\texttt`).

Пример A.3, иллюстрирующий подключение переопределённого языка. Может быть полезным, если подсветка кода работает криво. Без дополнительного окружения, с подписью и ссылкой, реализованной встроенным средством.

Листинг A.3 Пример листинга с подписью собственными средствами

```
## Caching the Inverse of a Matrix

## Matrix inversion is usually a costly computation and there may
## be some
5 ## benefit to caching the inverse of a matrix rather than compute
## it repeatedly
## This is a pair of functions that cache the inverse of a matrix.

## makeCacheMatrix creates a special "matrix" object that can
## cache its inverse

10 makeCacheMatrix <- function(x = matrix()) {#кириллица
  вкомментарияхпри xelatex и lualatex имеетпроблемыпробелами
  i <- NULL
  set <- function(y) {
    x <<- y
    i <<- NULL
15 }
  get <- function() x
  setSolved <- function(solve) i <<- solve
  getSolved <- function() i
  list(set = set, get = get,
20 setSolved = setSolved,
  getSolved = getSolved)
}
```

```

25  ## cacheSolve computes the inverse of the special "matrix"
    ## returned by
    ## makeCacheMatrix above. If the inverse has already been
    ## calculated (and the
    ## matrix has not changed), then the cachesolve should retrieve
    ## the inverse from
    ## the cache.
30  cacheSolve <- function(x, ...) {
    ## Return a matrix that is the inverse of 'x'
    i <- x$getSolved()
    if(!is.null(i)) {
35      message("getting cached data")
      return(i)
    }
    data <- x$get()
    i <- solve(data, ...)
40    x$setSolved(i)
    i
  }

```

Листинг A.4 подгружается из внешнего файла. Приходится загружать без окружения дополнительного. Иначе по страницам не переносится.

#### Листинг A.4 Листинг из внешнего файла

```

# Analysis of data on Course Project at Getting and Cleaning data
# course of Data Science track at Coursera.

# Part 1. Merges the training and the test sets to create one data
# set.
5 # 3. Uses descriptive activity names to name the activities in the
  # data set
# 4. Appropriately labels the data set with descriptive variable
  # names.

if (!file.exists("UCI HAR Dataset")) {
  stop("You need 'UCI HAR Dataset' folder full of data")
10 }

library(plyr) # for mapvalues
15

```

```

#getting common data
features <- read.csv("UCI HAR Dataset/features.txt", sep=" ",
  header = FALSE,
                                colClasses = c("numeric", "character"))
activity_labels <- read.csv("UCI HAR Dataset/activity_labels.txt",
  sep="",
20                                header = FALSE, colClasses = c("numeric",
                                "character"))

#getting train set data
subject_train <- read.csv("UCI HAR Dataset/train/subject_train.txt",
  "",
                                header = FALSE, colClasses = "numeric",
                                col.names="Subject")
25 y_train <- read.csv("UCI HAR Dataset/train/y_train.txt", header =
  FALSE,
                                colClasses = "numeric")
x_train <- read.csv("UCI HAR Dataset/train/X_train.txt", sep="",
  header = FALSE,
                                colClasses = "numeric", col.names=features$V2,
                                check.names = FALSE)

30 activity_train <- as.data.frame(mapvalues(y_train$V1, from =
  activity_labels$V1,
                                to = activity_labels$V2)
                                )

names(activity_train) <- "Activity"

35
#getting test set data
subject_test <- read.csv("UCI HAR Dataset/test/subject_test.txt",
  header = FALSE, colClasses = "numeric", col
                                .names="Subject")
y_test <- read.csv("UCI HAR Dataset/test/y_test.txt", header =
  FALSE,
40                                colClasses = "numeric")
x_test <- read.csv("UCI HAR Dataset/test/X_test.txt", sep="",
  header = FALSE,
                                colClasses = "numeric", col.names=features$V2,
                                check.names = FALSE)

```

```

activity_test <- as.data.frame(mapvalues(y_test$V1, from =
  activity_labels$V1,
45                                     to = activity_labels$V2))
names(activity_test) <- "Activity"

# Forming full dataframe
50 data_train <- cbind(x_train, subject_train, activity_train)
data_test <- cbind(x_test, subject_test, activity_test)
data <- rbind(data_train, data_test)

# Cleaning memory
55 rm(features, activity_labels, subject_train, y_train, x_train,
  activity_train,
  subject_test, y_test, x_test, activity_test, data_train, data_
    test)

# Part 2. Extracts only the measurements on the mean and standard
  deviation for each measurement.
60 cols2match <- grep("(mean|std)", names(data))

# Excluded gravityMean, tBodyAccMean, tBodyAccJerkMean,
  tBodyGyroMean,
# tBodyGyroJerkMean, as these represent derivations of angle data,
  as
65 # opposed to the original feature vector.

# Subsetting data frame, also moving last columns to be first
Subsetted_data_frame <- data[, c(562, 563, cols2match)]

70 # Part 5. From the data set in step 4, creates a second,
  independent tidy data set
# with the average of each variable for each activity and each
  subject.

library(dplyr) # for %>% and summarise_each

75 tidydata <- Subsetted_data_frame %>% group_by(Subject, Activity)
  %>%
  summarise_each(funs(mean))

```



```
write.table(tidydata, "tidydata.txt", row.names=FALSE)
```

## Приложение Б

**Очень длинное название второго приложения, в котором продемонстрирована работа с длинными таблицами**

### Б.1 Подраздел приложения

Вот размещается длинная таблица:

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
продолжение следует			

(продолжение)			
Параметр	Умолч.	Тип	Описание
mars kick	0 1	int int	экватора 1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars kick	0 1	int int	1: инициализация модели для планеты Марс 0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
продолжение следует			

(продолжение)			
Параметр	Умолч.	Тип	Описание
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

## Б.2 Ещё один подраздел приложения

Нужно больше подразделов приложения!

Пример длинной таблицы с записью продолжения по ГОСТ 2.105

Таблица 6 — Наименование таблицы средней длины

Параметр	Умолч.	Тип	Описание
&INP			
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ )

Продолжение таблицы 6

Параметр	Умолч.	Тип	Описание
			1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

Продолжение таблицы 6

Параметр	Умолч.	Тип	Описание
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
&SURFPAR			
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора

Продолжение таблицы 6

Параметр	Умолч.	Тип	Описание
			экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс
kick	1	int	0: инициализация без шума ( $p_s = const$ ) 1: генерация белого шума 2: генерация белого шума симметрично относительно экватора
mars	0	int	1: инициализация модели для планеты Марс

### Б.3 Использование длинных таблиц с окружением *longtabu*

В таблице 7 более книжный вариант длинной таблицы, используя окружение `longtabu` и разнообразные `toprule` `midrule` `bottomrule` из пакета `booktabs`. Чтобы визуально таблица смотрелась лучше, можно использовать следующие параметры: в самом начале задаётся расстояние между строками с помощью `arraystretch`. Таблица задаётся на всю ширину, `longtabu` позволяет делить ширину колонок пропорционально — тут три колонки в пропор-

ции 1.1:1:4 — для каждой колонки первый параметр в описании  $X[]$ . Кроме того, в таблице убраны отступы слева и справа с помощью  $@{\}$  в преамбуле таблицы. К первому и второму столбцу применяется модификатор

$\gt{\setlength{\baselineskip}{0.7\baselineskip}}$ ,

который уменьшает межстрочный интервал в для текста таблиц (иначе заголовок второго столбца значительно шире, а двухстрочное имя сливается с окружающими). Для первой и второй колонки текст в ячейках выравниваются по центру как по вертикали, так и по горизонтали - задаётся буквами  $m$  и  $c$  в описании столбца  $X[]$ .

Так как формулы большие — используется окружение `alignedat`, чтобы отступ был одинаковый у всех формул — он сделан для всех, хотя для большей части можно было и не использовать. Чтобы формулы занимали поменьше места в каждом столбце формулы (где надо) используется `\textstyle` — он делает дроби меньше, у знаков суммы и произведения — индексы сбоку. Иногда формулы слишком большая, сливается со следующей, поэтому после неё ставится небольшой дополнительный отступ `\vspace*{2ex}` Для штрафных функций — размер фигурных скобок задан вручную `\Big\{`, т.к. не умеет `alignedat` работать с `\left` и `\right` через несколько строк/колонок.

В примечании к таблице наоборот, окружение `cases` даёт слишком большие промежутки между вариантами, чтобы их уменьшить, в конце каждой строчки окружения использовался отрицательный дополнительный отступ `\[-0.5em]`.

Таблица 7 — Тестовые функции для оптимизации,  $D$  — размерность. Для всех функций значение в точке глобального минимума равно нулю.

Имя	Стартовый диапазон параметров	Функция
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$

продолжение следует



(продолжение)

Имя	Стартовый диапазон параметров	Функция
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D [x_i + 0.5]^2$
зашумлённая квартическая	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D ix_i^4 + rand[0,1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) -$ $- \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \right.$ $\left. + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$
сфера	$[-100, 100]^D$	$f_1(x) = \sum_{i=1}^D x_i^2$
Schwefel 2.22	$[-10, 10]^D$	$f_2(x) = \sum_{i=1}^D  x_i  + \prod_{i=1}^D  x_i $
Schwefel 1.2	$[-100, 100]^D$	$f_3(x) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2$

продолжение следует

(продолжение)

Имя	Стартовый диапазон параметров	Функция
Schwefel 2.21	$[-100, 100]^D$	$f_4(x) = \max_i \{ x_i \}$
Rosenbrock	$[-30, 30]^D$	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
ступенчатая	$[-100, 100]^D$	$f_6(x) = \sum_{i=1}^D [x_i + 0.5]^2$
зашумлённая квартическая	$[-1.28, 1.28]^D$	$f_7(x) = \sum_{i=1}^D ix_i^4 + rand[0,1)$
Schwefel 2.26	$[-500, 500]^D$	$f_8(x) = \sum_{i=1}^D -x_i \sin \sqrt{ x_i } +$ $+ D \cdot 418.98288727243369$
Rastrigin	$[-5.12, 5.12]^D$	$f_9(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$
Ackley	$[-32, 32]^D$	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) -$ $-\exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$
Griewank	$[-600, 600]^D$	$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos(x_i/\sqrt{i}) + 1$
штрафная 1	$[-50, 50]^D$	$f_{12}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + \right.$ $\left. + (y_D - 1)^2 \right\} + \sum_{i=1}^D u(x_i, 10, 100, 4)$
штрафная 2	$[-50, 50]^D$	$f_{13}(x) = 0.1 \left\{ \sin^2(3\pi x_1) + \right.$ $\left. + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + \right.$ $\left. + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} +$ $+ \sum_{i=1}^D u(x_i, 5, 100, 4)$

продолжение следует

(окончание)

Имя	Стартовый диапазон параметров	Функция
Примечание — Для функций $f_{12}$ и $f_{13}$ используется $y_i = 1 + \frac{1}{4}(x_i + 1)$ и		
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$		

#### Б.4 Форматирование внутри таблиц

В таблице 8 пример с чересстрочным форматированием. В `userstyles.tex` задаётся счётчик `\newcounter{rowcnt}` который увеличивается на 1 после каждой строчки (как указано в преамбуле таблицы). Кроме того, задаётся условный макрос `\altshape` который выдаёт одно из двух типов форматирования в зависимости от чётности счётчика.

В таблице 8 каждая чётная строчка — синяя, нечётная — с наклоном и слегка поднята вверх. Визуально это приводит к тому, что среднее значение и средне-квадратичное изменение группируются и хорошо выделяются взглядом в таблице. Сохраняется возможность отдельные значения в таблице выделить цветом или шрифтом. К первому и второму столбцу форматирование не применяется по сути таблицы, к шестому общее форматирование не применяется для наглядности.

Так как заголовок таблицы тоже считается за строчку, то перед ним (для первого, промежуточного и финального варианта) счётчик обнуляется, а в `\altshape` для нулевого значения счётчика форматирования не применяется.

Таблица 8 — Длинная таблица с примером чересстрочного форматирования

	Итерации	JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f1	1500	<b>1.8E-60</b> (8.4E-60)	<b>1.3E-54</b> (9.2E-54)	<b>2.5E-28</b> (3.5E-28)	4.5E-20 (6.9E-20)	<b>9.8E-14</b> (8.4E-14)	<b>9.6E-42</b> (2.7E-41)
f2	2000	<b>1.8E-25</b> (8.8E-25)	<b>3.9E-22</b> (2.7E-21)	<b>1.5E-23</b> (1.0E-23)	1.9E-14 (1.1E-14)	<b>1.6E-09</b> (1.1E-09)	<b>9.3E-21</b> (6.3E-20)

продолжение следует

(продолжение)

Итерации		JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)
f1	1500	1.8E-60 (8.4E-60)	1.3E-54 (9.2E-54)	2.5E-28 (3.5E-28)	4.5E-20 (6.9E-20)	9.8E-14 (8.4E-14)	9.6E-42 (2.7E-41)
f2	2000	1.8E-25 (8.8E-25)	3.9E-22 (2.7E-21)	1.5E-23 (1.0E-23)	1.9E-14 (1.1E-14)	1.6E-09 (1.1E-09)	9.3E-21 (6.3E-20)
f3	5000	5.7E-61 (2.7E-60)	6.0E-87 (1.9E-86)	5.2E-14 (1.1E-13)	9.0E-37 (5.4E-36)	6.6E-11 (8.8E-11)	2.5E-19 (3.9E-19)
f4	5000	8.2E-24 (4.0E-23)	4.3E-66 (1.2E-65)	1.4E-15 (1.0E-15)	7.4E-11 (1.8E-10)	4.2E-01 (1.1E+00)	4.4E-14 (9.3E-14)
f5	3000	8.0E-02 (5.6E-01)	3.2E-01 (1.1E+00)	1.3E+01 (1.4E+01)	2.1E+01 (7.8E+00)	2.1E+00 (1.5E+00)	2.5E+01 (3.2E+01)
f6	100	2.9E+00 (1.2E+00)	5.6E+00 (1.6E+00)	1.0E+03 (2.2E+02)	9.3E+02 (1.8E+02)	4.7E+03 (1.1E+03)	4.5E+01 (2.4E+01)

продолжение следует

(окончание)

	Итерации	JADE++	JADE	jDE	SaDE	DE/rand /1/bin	PSO
f7	3000	6.4E-04 (2.5E-04)	6.8E-04 (2.5E-04)	3.3E-03 (8.5E-04)	4.8E-03 (1.2E-03)	4.7E-03 (1.2E-03)	2.5E-03 (1.4E-03)
f8	1000	3.3E-05 (2.3E-05)	7.1E+00 (2.8E+01)	7.9E-11 (1.3E-10)	4.7E+00 (3.3E+01)	5.9E+03 (1.1E+03)	2.4E+03 (6.7E+02)
f9	1000	1.0E-04 (6.0E-05)	1.4E-04 (6.5E-05)	1.5E-04 (2.0E-04)	1.2E-03 (6.5E-04)	1.8E+02 (1.3E+01)	5.2E+01 (1.6E+01)
f10	500	8.2E-10 (6.9E-10)	3.0E-09 (2.2E-09)	3.5E-04 (1.0E-04)	2.7E-03 (5.1E-04)	1.1E-01 (3.9E-02)	4.6E-01 (6.6E-01)
f11	500	9.9E-08 (6.0E-07)	2.0E-04 (1.4E-03)	1.9E-05 (5.8E-05)	7.8E-04 (1.2E-03)	2.0E-01 (1.1E-01)	1.3E-02 (1.7E-02)
f12	500	4.6E-17 (1.9E-16)	3.8E-16 (8.3E-16)	1.6E-07 (1.5E-07)	1.9E-05 (9.2E-06)	1.2E-02 (1.0E-02)	1.9E-01 (3.9E-01)
f13	500	2.0E-16 (6.5E-16)	1.2E-15 (2.8E-15)	1.5E-06 (9.8E-07)	6.1E-05 (2.0E-05)	7.5E-02 (3.8E-02)	2.9E-03 (4.8E-03)

### Б.5 Очередной подраздел приложения

Нужно больше подразделов приложения!

### Б.6 И ещё один подраздел приложения

Нужно больше подразделов приложения!