Пример ВКР «SPbPU-student-thesis-template». Версия от 3 июля 2019 г.

Минобрнауки России
Санкт-Петербургский политехнический университет Петра Великого
Название института

Работа допущена к защите
Старший М
_____ К.А. Туральчук
«_____» _____ 20XX г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

НАЗВАНИЕ ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ

по направлению XX.XX.XX Наименование направления/специальности
по образовательной программе
XX.XX.XX_YY Наименование образовательной программы

Выполнил
студент гр.N Д.М. Момот

Руководитель
должность, степень[1] И.О. Фамилия

Консультант
по . . . [2]
должность, степень И.О. Фамилия

Консультант
по нормоконтролю[3] И.О. Фамилия

Санкт-Петербург
20XX

[1]Должность указывают сокращенно, подразделения — аббревиатурами. «СПбПУ» и аббревиатуры институтов не добавляют.

[2]Оформляется по решению руководителя ОП или подразделения. Поясняющие 1-3 слова помещаются на титул и в задание. «Научный консультант» должен иметь степень. Без печати и заверения подписи.

[3]Обязателен, из числа ППС по решению руководителя ОП или подразделения. Должность и степень не указываются. Сведения помещаются в последнюю строчку по порядку. Рецензенты не указываются.

САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО

Название института

УТВЕРЖДАЮ
Старший М
_____ К.А. Туральчук
«_____» _____ 20XXг.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы

студенту Момоту Даниэлю Михайловичу гр.N

1. Тема работы: Название выпускной квалификационной работы.
2. Срок сдачи студентом законченной работы[4]: дд.мм.гггг.
3. Исходные данные по работе[5]: статистические данные с сайта [3.1], а также из репозитория [3.4]; основным источником литературы является монография [3.3] и статья [3.2].
   - 3.1. Сайт Федеральной службы государственной статистики. — URL: http://www.gks.ru/ (дата обращения: 06.03.2019).
   - 3.2. *Adams P.* The title of the work // The name of the journal. — 1993. — Vol. 4, no. 2. — P. 201–213.
   - 3.3. *Babington P.* The title of the work. Vol. 4. — 3rd ed. — The address: The name of the publisher, 1993. — 255 p. — (Ser.: 10).
   - 3.4. The UC Irvine Machine Learning Repository. — URL: http://archive.ics.uci.edu/ml (visited on 06.03.2019).
4. Содержание работы (перечень подлежащих разработке вопросов):
   - 4.1. Обзор литературы по теме ВКР.
   - 4.2. Исследование программных продуктов.

[4]Определяется руководителем ОП, но не позднее последнего числа преддипломной практики и/или не позднее, чем за 20 дней до защиты в силу п. 6.1. «Порядка обеспечения самостоятельности выполнения письменных работ и проверки письменных работ на объем заимствований».

[5]Текст, который подчёркнут и/или выделен в отдельные элементы нумерационного списка, приведён в качестве примера.

4.3. Разработка метода/алгоритма/программы.

4.4. Апробация разработанного метода/алгоритма/программы.

5. Перечень графического материала (с указанием обязательных чертежей):

5.1. Схема работы метода/алгоритма.

5.2. Архитектура разработанной программы/библиотеки.

6. Консультанты по работе:

6.1. Нормоконтроль: должность, степень, И.О. Фамилия.

6.2. Разработка программы (алгоритма, метода)[6]: должность, степень, И.О. Фамилия.

7. Дата выдачи задания[7]: дд.мм.гггг.

Руководитель ВКР _____ И.О. Фамилия

Задание принял к исполнению дд.мм.гггг

Студент _____ Д.М. Момот

[6]Возможны также формулировки: «Вопросы программирования/анализа. . .», «Оценка эффективности/быстродействия. . .» и другие.

[7]Не позднее 3 месяцев до защиты (утверждение тем ВКР по университету) или первого числа преддипломной практики или по решению руководителя ОП или подразделения (открытый вопрос).

РЕФЕРАТ

23 с., 1 рисунок, 3 таблицы, 2 приложения.

СТИЛЕВОЕ ОФОРМЛЕНИЕ САЙТА, УПРАВЛЕНИЕ КОНТЕНТОМ, PHP, MYSQL, АРХИТЕКТУРА СИСТЕМЫ[8]

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики.[9]

В данной работе изложена сущность подхода к созданию динамического информационного портала на основе использования открытых технологий Apache, MySQL и PHP. Даны общие понятия и классификация IT-систем такого класса. Проведен анализ систем-прототипов. Изучена технология создания указанного класса информационных систем. Разработана конкретная программная реализация динамического информационного портала на примере портала выбранной тематики.

ABSTRACT

23 p., 1 figures, 3 tables, 2 appendices.

STYLE REGISTRATION, CONTENT MANAGEMENT, PHP, MYSQL, SYSTEM ARCHITECTURE

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class

[8]Всего **слов**: от 3 до 15. Всего **слов и словосочетаний**: от 3 до 5. Оформляются в именительном падеже множественного числа (или в единственном числе, если нет другой формы), оформленных по правилам русского языка.

[9]До 600 печатных знаков (ГОСТ Р 7.0.99-2018 СИБИД) на русский или английский текст. Текст реферата повторён дважды на русском и английском языке для демонстрации подхода к нумерации страниц. *Внимание! Эта сноска размещена после точки. Это пример как не нужно оформлять сноску.*

of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed.

In the given work the essence of the approach to creation of a dynamic information portal on the basis of use of open technologies Apache, MySQL and PHP is stated. The general concepts and classification of IT-systems of such class are given. The analysis of systems-prototypes is lead. The technology of creation of the specified class of information systems is investigated. Concrete program realization of a dynamic information portal on an example of a portal of the chosen subjects is developed.

СОДЕРЖАНИЕ

Введение 6
Глава 1. Изучение основ общей алгебры 7
Глава 2. Разработка приложения для генерации нормализованных систем уравнений 9
2.1. Общие сведения 9
2.2. Детали реализации 10
2.2.1. Служебные модули 10
2.2.2. Матрицы 10
2.2.3. Полиномы 11
2.2.4. Генерация псевдослучайных объектов 12
2.2.5. Преобразования 12
2.2.6. Ввод-вывод 13
2.2.7. Высокоуровневый алгоритм и взаимодействие с пользователем 13
2.3. Алгоритм генерации случайных систем уравнений 14
2.4. Алгоритм решения систем уравнений и тестирования 14
2.5. Алгоритм нормализации систем уравнений 15
Глава 3. Название третьей главы: разработка программного обеспечения 17
3.1. Введение 17
3.2. Название параграфа 17
3.3. Название параграфа 17
3.4. Выводы 17
Глава 4. Название четвёртой главы. Апробация результатов исследования, а именно: метода, алгоритма, модели исследования 17
4.1. Введение 17
4.2. Название параграфа 17
4.3. Название параграфа 17
4.4. Выводы 17
Заключение 18
Список сокращений и условных обозначений 19
Словарь терминов 20
Список использованных источников 21
Приложение 1. Краткие инструкции по настройке издательской системы LaTeX 24
Приложение 2. Некоторые дополнительные примеры 28

ВВЕДЕНИЕ

Криптография — это наука о методах обеспечения конфиденциальности и целостности данных. В настоящее время она является одной из важнейших областей дискретной математики. Методы криптографии применяются практически во всех отраслях, требующих обеспечения безопасности данных: электронная коммерция, технологии криптовалюты, электронный документооборот, телекоммуникации.

Одним из наиболее популярных направлений криптографии является криптография с открытым ключом. Этот принцип предусматривает наличие двух ключей: публичного (открытого), используемого для шифрования данных, и секретного (закрытого), используемого для расшифровки. При этом к секретному ключу предъявляется требование невозможности его вычисления за разумный срок.

В рамках данной практики поставлена цель разработать приложение, реализующее часть криптографической системы с открытым ключом. В ней в качестве открытого ключа выступает система уравнений, вычисляемая на основе случайно сгенерированных исходных данных. Процесс шифрования состоит в подстановке вектора переменных в неё, а процесс расшифровки – в решении системы. Расшифровка не может быть быстро произведена без знания исходных данных для системы уравнений, что и обеспечивает криптостойкость разрабатываемой системы.

Для достижения выбранной цели поставлены следующие задачи:

A. Изучить основы общей алгебры (и других математических инструментов, необходимых для понимания и реализации используемых алгоритмов).
B. Разработать модуль генерации систем уравнений на основе случайно генерируемых входных данных (матриц и векторов).
C. Разработать модуль решения систем уравнений.
D. Разработать модуль нормализации системы уравнений (представления в виде большего количества более простых уравнений).

Разработанное приложение имеет консольный интерфейс и написано на языке C++, использовался стандарт ISO C++14. Разработка велась в среде Microsoft Visual Studio 2017. В процессе разработки использовалась система контроля версий Git.

ГЛАВА 1. ИЗУЧЕНИЕ ОСНОВ ОБЩЕЙ АЛГЕБРЫ

Для понимания алгоритма и последующей его реализации, необходимо было подготовить математическую базу: освежить часть понятий, а часть узнать впервые. Далее представлены некоторые из таких понятий.

Группа - множество, на котором задана ассоциативная бинарная операция, для которой имеется нейтральный элемент, и для каждого элемента определён обратный к нему. Если операция является коммутативной, кольцо тоже называют коммутативным.

Примеры групп: целые числа и четные числа - группы по сложению, рациональные числа без нуля - группа по умножению.

Кольцо - множество, на котором заданы две бинарные операции + и *, называемые сложением и умножением. При этом сложение коммутативно, ассоциативно и имеет нейтральный элемент, и для каждого элемента есть противоположный элемент. Умножение ассоциативно; также должна присутствовать двусторонняя дистрибутивность умножения относительно сложения. Кольцо может обладать нейтральным элементом по умножению (в этом случае оно называется кольцом с единицей) и коммутативностью умножения (в этом случае оно называется коммутативным).

Примеры колец: вещественные числа, комплексные числа, множество функций, стремящихся к нулю в единице.

Поле - множество, на котором заданы две бинарные операции + и *, называемые сложением и умножением. При этом сложение коммутативно, ассоциативно и имеет нейтральный элемент, и для каждого элемента есть противоположный элемент. Умножение ассоциативно, коммутативно, имеет нейтральный элемент, и для каждого элемента есть противоположный элемент; также должна присутствовать двусторонняя дистрибутивность умножения относительно сложения.

Примеры полей: рациональные числа, комплексные числа.

Другими словами, кольцо является коммутативной группой по сложению, а поле является коммутативным кольцом с единицей. В группе можно складывать и вычитать элементы, кольцо добавляет операцию умножения, а в поле можно еще и делить (делением называют взятие элемента, обратного по умножению).

Конечным полем, или полем Галуа, называют поле, состоящее из конечного числа элементов. Можно показать, что количество элементов конечного поля

является степенью некоторого простого числа. Это простое число называется характеристикой поля, а количество элементов поля называют порядком.

Рассмотрим пример конечного поля из двух элементов. Оно обозначается F_2 или $GF(2)$. Элементы можно определить как 0 и 1, в этом случае операции + и * определяются как сложение по модулю 2 и умножение соответственно. Также элементы этого конечного поля можно определить как «Ложь» и «Истина», тогда + и * определяются как «исключающее или» и «и» соответственно. Тем не менее, это всего лишь два представления одного и того же поля:

Таблица 1.1

Операции над элементами поля $GF(2)$, представленными в виде чисел

+	0	1
0	0	1
1	1	0

*	0	1
0	0	0
1	0	1

Таблица 1.2

Операции над элементами поля $GF(2)$, представленными в виде логических объектов

+	F	T
F	F	T
T	T	F

*	F	T
F	F	F
T	F	T

Многочленом над полем называется многочлен, коэффициенты которого принадлежат заданному полю. Так как многочлены можно складывать, вычитать и умножать (и эти операции коммутативны), множество всех многочленов над данным полем является кольцом.

Аффинное преобразование – преобразование вида $f(x) = Mx + v$, где M – обратимая матрица, v - вектор.

ГЛАВА 2. РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ НОРМАЛИЗОВАННЫХ СИСТЕМ УРАВНЕНИЙ

2.1. Общие сведения

Приложение имеет консольный интерфейс на английском языке. Сгенерированные данные записываются в файлы (подробнее описано ниже). При запуске приложения можно ввести ключ $/h$ для вызова справки или задать аргументы. Аргументы должны вводиться в следующем порядке:

A. Первый – количество уравнений в генерируемой системе (натуральное число). Обязателен.

B. Второй аргумент, если задан – ключ запуска. Допускаются ключи:
   1. $/s$ — тихий запуск, в этом режиме в папку записываются только файлы с системой, её решением и нормализованной системой;
   2. $/r$ — стандартный запуск (по умолчанию), в этом режиме в папку записываются те же файлы, что в тихом режиме, а также исходные данные и все промежуточные результаты;
   3. $/t$ — запуск в режиме тестирования, в этом случае осуществляется стандартный запуск, после чего происходит тестирование для различных векторов. Процесс тестирования описан ниже.

C. Третий аргумент, если задан — имя папки, куда будут записаны сгенерированные файлы. Если директория с таким именем не существует, она будет предварительно создана. По умолчанию используется имя “results”.

В папке, имя которой передается третьим аргументом, создается папка с именем вида YYYY.MM.DD_HH.MM.SS — определяется системной датой и временем. В эту папку записываются сгенерированные данные в следующих файлах:

A. Случайно сгенерированные исходные данные:
   1. pre_rand/M1.txt – матрица M_1;
   2. pre_rand/M2.txt – матрица M_2;
   3. pre_rand/v1.txt – вектор v_1;
   4. pre_rand/v2.txt – вектор v_2.

B. Промежуточные данные:
   1. pre_gen/S.txt – преобразование S;
   2. pre_gen/T.txt – преобразование T;

3. pre_gen/F.txt – преобразование F;
4. inv/invM1.txt – матрица, обратная к M_1;
5. inv/invM2.txt – матрица, обратная к M_2;
6. inv/invF.txt – преобразование, обратное к F;

C. Результат работы программы:

1. P.txt – ненормализованная система уравнений;
2. P_sol.txt – решение ненормализованной системы уравнений;
3. P_norm.txt – нормализованная система уравнений;

2.2. Детали реализации

Наиболее важные элементы реализации приложения описаны далее в этом разделе.

2.2.1. Служебные модули

Реализация базируется на следующих служебных модулях:

A. $Utility$ – содержит используемые в других модулях строковые функции.

B. $File\_system$ – содержит функции, используемые при работе с файловой системой.

C. $BOOL$ – псевдоним для типа данных int, также определены константы $FALSE = 0$ и $TRUE = 1$. Он используется в качестве замены логического типа данных, что позволяет достичь увеличения скорости работы с данными на 30-50%.

2.2.2. Матрицы

В пространстве имен $matrixes$ описаны следующие классы:

A. Row – описывает строку матрицы или вектор-столбец. Агрегирует объект типа $std :: vector < BOOL >$.

B. $Matrix$ – описывает квадратную матрицу, агрегирует объект типа $std :: vector < Row < BOOL > * >$. Использование указателей позволяет заметно снизить накладные расходы. Реализован метод $init\_zeros()$, который задает размерность матрицы и инициализирует ее нулями, и метод $initInverse()$, инициализирующий матрицу как матрицу, обратную по отношению к переданной по ссылке в качестве параметра.

C. *MatrixBuilder* – реализует паттерн Строитель, предоставляет удобный интерфейс для определения объектов *Matrix*.

2.2.3. Полиномы

В пространстве имен *polynomials* описаны следующие классы:

A. *Monomial* – представляет моном, или же терм. Так как все конструкции находятся в поле $GF(2)$, степени всех переменных не превышают первую, поэтому хранится только список переменных, представленных в терме (в виде вектора). Так как полиномы генерируются самим приложением, становится возможной дальнейшая оптимизация: хранить только номера переменных, а их названия определять простым добавлением номера к букве x. Так, переменная x5 имеет номер 5. Все коэффициенты также равны единице, поэтому тоже не хранятся. Основные методы — *simplify*(), вызываемый после каждого изменения структуры терма, в том числе в конце работы конструктора, он гарантирует упорядоченность переменных по возрастанию; и метод *substitute*(), подставляющий набор значений переменных в моном и возвращающий результат типа *BOOL*.

B. *Polynomial* – описывает полином, представляет собой вектор мономов. Определены операторы + = и ∗ =, позволяющие прибавлять моном и домножать на моном, соответственно. Определен метод *substitute*(), рекурсивно вызывающий метод *substitute*() каждого монома. Мономы в каждый момент отсортированы, за что отвечает метод *simplify*(). Порядок сортировки следующий:
    1. Мономы большей степени расположены раньше, чем мономы меньшей степени;
    2. Мономы равной степени сортируются лексикографически.

C. Также в пространстве имен *polynomials* определены классы, реализующие паттерн Строитель: *MonomialBuilder*, упрощающий создание объектов *Monomial*, и *PolynomialBuilder* и *DNFBuiler* – они оба упрощают создание объектов *Polynomial*, но по-разному: *PolynomialBuilder* собирает полином из мономов, а *DNFBuilder* собирает полином как сумму других полиномов.

2.2.4. Генерация псевдослучайных объектов

Генерация псевдослучайных объектов (ПСО) не может обеспечиваться независимым вызовом функции-генератора псевдослучайных чисел (ГПСЧ), так как вызовы могут поступать с интервалом менее одной секунды, в результате чего различные объекты будут инициализироваться одинаковыми значениями. В связи с этим создано пространство имен *random*, содержащее несколько классов. Они генерируют случайные объекты, используя один и тот же объект ГПСЧ:

A. *RandomEngine* – хранит ГПСЧ *std* :: *mt*19937 (вихрь Мерсенна). Имеет метод *getRandomEngine*(), возвращающий константную ссылку на этот объект. Используется для инициализации ГПСЧ других классов этого пространства имен.
B. $RandomMatrixFactory$ – реализует паттерн Фабрика, генерирует ПСО типа $Matrix$ и Row.
C. $RandomPolynomialFactory$ – реализует паттерн Фабрика, генерирует ПСО типа $Polynomial$ (полином второй степени).

2.2.5. Преобразования

В пространстве имен $transformations$ определены следующие классы:

A. $Transformation$ — определяет преобразование. Инкапсулирует *std* :: $vector < Polynomial >$, каждый полином соответствует преобразованию одной координаты. В нем определен метод *initComposition*(), инициализирующий преобразование как композицию двух преобразований, передаваемых по константной ссылке. Также определен метод *substitute*(), вызывающий метод *substitute*() для полиномов всех координат, и возвращающий *std* :: $vector < BOOL >$ (по ссылке, чтобы избежать лишнего копирования), и метод *normalize*(), нормализующий систему (алгоритм его работы представлен в соответствующем разделе ниже).
B. $AffineTransformation$ — наследуется от класса $Transformation$ и позволяет определять аффинное преобразование по матрице M и вектору v как $F(x) = Mx + v$.
C. $TransformationBuilder$ — реализует паттерн строитель для объектов $Transformation$.

2.2.6. Ввод-вывод

Для организации единообразной структуры ввода-вывода в проекте в пространстве имен *IO* определены следующие классы:

A. *Writer* — производит запись в файл, инкапсулирует объект типа *ofstream*. Может записывать в файл объекты типов *Row*, *Matrix*, *Polynomial*.
B. *Reader* — производит чтение из файла, инкапсулирует объект типа *ifstream*. Может принимать из файла объекты типов *Row*, *Matrix*, *Transformation*.
C. *Parser* — используется при чтении *Transformation* из файла, разбирает строку в объект *Polynomial*.
D. *ParserBackground* — используется классом *Parser*, предоставляет низкоуровневый функционал для разбора строк. В том числе, содержит машину состояний.

2.2.7. Высокоуровневый алгоритм и взаимодействие с пользователем

Ввиду несложности взаимодействия с пользователем, оно определено прямо в файле *Main*, содержащем точку входа приложения. Метод *main*() принимает параметры (что позволяет передавать аргументы прямо из командной строки), если же они не поступили, то аргументы запрашиваются, если же они снова не поступают, то показывается окно справки. Разбор аргументов осуществляется также в файле *Main*.

После приема параметров создается объект класса *Environment* и запускается его метод *run*() с параметрами (в режиме тестирования или нет, удалить лишние файлы в конце работы приложения или нет, печатать аргументы в консоль или нет – последний аргумент во всех случаях *true*, но легко может быть изменен при добавлении новых режимов запуска). В методе *run*(), в зависимости от переданных параметров, запускаются методы *generateSystem*(), *solveSystem*(), *normalizeSystem*(), *testYourself*(), реализующими, соответственно, генерацию системы, решение системы, нормализацию систему и тестирования решения системы. Код этих методов приведен в соответствующем приложении.

2.3. Алгоритм генерации случайных систем уравнений

При генерации систем уравнений использовался следующий алгоритм:

A. Принимается число n.

B. Генерируются случайные обратимые матрицы M_1 и M_2 над полем $GF(2)$ и случайные векторы v_1 и v_2 над тем же полем. Размерность матриц и векторов n.

C. Строятся аффинные преобразования $S = M_1X + v_1$, $T = M_2X + v_2$, где X – вектор переменных. Благодаря обратимости матриц, к ним будут существовать обратные матрицы и, следовательно, будут существовать также преобразования, обратные к S и T.

D. Строится преобразование F как $F[i] = x_i + g_i(x_0,x_1,\ldots,x_{i-1})$, где $i = 0..n-1, g_i(x_0,x_1,\ldots,x_{i-1}))$ – случайный квадратичный полином.

E. Вводим преобразование $P = SoFoT$.

Нетрудно понять, что решить полученную систему квадратичных уравнений без перебора и без знания преобразований S, T и F практически невозможно. Однако, используя знание о промежуточных преобразованиях, можно решить систему за полиномиальное время (подробнее описано в следующем разделе).

2.4. Алгоритм решения систем уравнений и тестирования

Итак, преобразование построено как композиция преобразований S, F, T:

$$P = SoFoT. \tag{2.1}$$

Следовательно, обратное преобразование может быть построено как

$$P^{-1} = T^{-1}oF^{-1}oS^{-1}. \tag{2.2}$$

А, имея обратное преобразование, несложно решить систему, подставив в него нуль-вектор:

$$P(X) = 0 \Rightarrow P^{-1}(0) = X. \tag{2.3}$$

Аналогично, для проверки нахождения обратного преобразования, достаточно проверить, что для любого вектора переменных X выполняется

$$P^{-1}(P(X)) = X. \tag{2.4}$$

Искомое обратное преобразование легко может быть построено, зная v_1,v_2,M_1,M_2,F:

$$P^{-1} = T^{-1}oF^{-1}oS^{-1} = M_2^{-1} * [F^{-1}(M_1^{-1} * (X + v_1)) + v_1]. \quad (2.5)$$

В свою очередь, преобразование F^{-1} получается из F последовательной подстановкой выражений для x_i. Это становится возможным благодаря тому, что $F[i]$ гарантированно содержит x_i и не содержит переменных с бОльшими номерами:

$$\begin{aligned} F[X] &= x_0 + g_0() \\ &= x_1 + g_1(x_0) \\ &\dots \\ &= x_{n-1} + g_{n-1}(x_0,x_1,\dots,x_{n-2}) \end{aligned} \quad \Rightarrow \quad (2.6)$$

$$\begin{aligned} F^{-1}[X] &= x_0 + g_0() \\ &= x_1 + g_1(x_0 + g_0()) \\ &\dots \\ &= x_{n-1} + g_{n-1}(x_0 + g_0(),x_1 + g_1(x_0),\dots,g_{n-2}(x_0,x_1,\dots,x_{n-3})) \end{aligned}$$

2.5. Алгоритм нормализации систем уравнений

При генерации систем уравнений на выход подаются квадратичные уравнения от n переменных. Следовательно, количество слагаемых пропорционально n^2, что определяет большие расходы по памяти и времени на обработку одного уравнения. Это обуславливает необходимость декомпозиции каждого уравнения на более простые уравнения. Кроме того, увеличение количества (и, следовательно, количества переменных) затрудняет попытки решения системы уравнений подбором: увеличение количества переменных на единицу удваивает требуемое время.

После нормализации системы все уравнения системы должны приобрести один из следующих форматов (в порядке убывания приоритета):

A. $x_z + x_i x_j = 0$;

B. $x_z + x_i + x_j = 0$;

C. $x_i + x_j + c = 0$, где x_z, x_i, x_j - переменные, c - константа.

Нормализация производится путем постепенного упрощения исходных уравнений (уравнения ядра) и попутного добавления новых уравнений, подчиняющихся шаблонам (уравнения связи).

Для нормализации системы достаточно нормализовать каждое уравнение ядра. Алгоритм нормализации одного уравнения ядра выглядит следующим образом:

A. Заменить каждое слагаемое второй степени $x_i x_j$ на новую переменную x_z, и добавить в конец системы новое уравнение связи $x_z = x_i x_j$.
B. Заменить сумму каждых двух слагаемых $x_i + x_j$ на новую переменную x_z, и добавить в конец системы новое уравнение связи $x_z = x_i + x_j$.
C. После каждой замены пройтись также по остальным уравнениям связи и произвести аналогичную замену, чтобы не появилось синонимичных переменных.
D. Продолжать процесс, пока не достигнуто последнее слагаемое полинома, или пока полином не приобрел шаблонный вид.

Рис.2.1. Новый научно-исследовательский корпус СПбПУ [29] (с помощью окружения minipage)

ГЛАВА 3. НАЗВАНИЕ ТРЕТЬЕЙ ГЛАВЫ: РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1. Введение

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

3.2. Название параграфа

3.3. Название параграфа

3.4. Выводы

Текст выводов по главе 3.

ГЛАВА 4. НАЗВАНИЕ ЧЕТВЁРТОЙ ГЛАВЫ. АПРОБАЦИЯ РЕЗУЛЬТАТОВ ИССЛЕДОВАНИЯ, А ИМЕННО: МЕТОДА, АЛГОРИТМА, МОДЕЛИ ИССЛЕДОВАНИЯ

4.1. Введение

Хорошим стилем является наличие введения к главе. Во введении может быть описана цель написания главы, а также приведена краткая структура главы.

4.2. Название параграфа

4.3. Название параграфа

Пример ссылки на литературу [1; 2; 5; 23].

4.4. Выводы

Текст выводов по главе 4.

ЗАКЛЮЧЕНИЕ

Заключение (2 – 5 страниц) содержит выводы по теме работы, конкретные предложения и рекомендации по исследуемым вопросам. Количество общих выводов должно вытекать из количества задач, сформулированных во введении выпускной квалификационной работы.

Предложения и рекомендации должны быть органически увязаны с выводами и направлены на улучшение функционирования исследуемого объекта. При разработке предложений и рекомендаций обращается внимание на их обоснованность, реальность и практическую приемлемость.

Заключение не должно содержать новой информации, положений, выводов и т. д., которые до этого не рассматривались в выпускной квалификационной работе. Рекомендуется писать заключение в виде тезисов.

Последним абзацем в заключении можно выразить благодарность всем людям, которые помогали автору в написании ВКР.

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

$\left.\begin{matrix} a_n \\ b_n \end{matrix}\right\}$ Коэффициенты разложения Ми в дальнем поле, соответствующие электрическим и магнитным мультиполям.

$\hat{\mathbf{e}}$ Единичный вектор.

E_0 Амплитуда падающего поля.

$\left.\begin{matrix} a_n \\ b_n \end{matrix}\right\}$ Коэффициенты разложения Ми в дальнем поле соответствующие электрическим и магнитным мультиполям ещё раз, но без окружения minipage нет вертикального выравнивания по центру.

j Тип функции Бесселя.

k Волновой вектор падающей волны.

$\left.\begin{matrix} a_n \\ b_n \end{matrix}\right\}$ Коэффициенты разложения Ми в дальнем поле соответствующие электрическим и магнитным мультиполям, теперь окружение minipage есть и добавленно много текста, так что описание группы условных обозначений значительно превысило высоту этой группы... Для отбивки пришлось добавить дополнительные отступы.

L Общее число слоёв.

l Номер слоя внутри стратифицированной сферы.

λ Длина волны электромагнитного излучения в вакууме.

n Порядок мультиполя.

$\left.\begin{matrix} \mathbf{N}^{(j)}_{e1n} & \mathbf{N}^{(j)}_{o1n} \\ \mathbf{M}^{(j)}_{o1n} & \mathbf{M}^{(j)}_{e1n} \end{matrix}\right\}$ Сферические векторные гармоники.

μ Магнитная проницаемость в вакууме.

r,θ,φ Полярные координаты.

ω Частота падающей волны.

BEM Boundary element method, метод граничных элементов.

CST MWS Computer Simulation Technology Microwave Studio.

СЛОВАРЬ ТЕРМИНОВ

TeX — система компьютерной вёрстки, разработанная американским профессором информатики Дональдом Кнутом.

Панграмма — короткий текст, использующий все или почти все буквы алфавита.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Котельников И. А.*, *Чеботаев П. З.* LaTeX по-русски. — 3-е изд. — Новосибирск: Сибирский Хронограф, 2004. — 496 с. — URL: http://www.tex.uniyar.ac.ru/doc/kotelnikovchebotaev2004b.pdf (дата обращения: 06.03.2019).

2. *Песков Н. В.* Поиск информативных фрагментов описаний объектов в задачах распознавания: дис. . . . канд. канд. физ.-мат. наук: 05.13.17 / Песков Николай Владимирович. — М., 2004. — 102 с.

3. Положение о порядке проведения государственной итоговой аттестации по образовательным программам высшего образования — программам бакалавриата, программам специалитета и программам магистратуры (в редакции приказа от 03.05.2018 № 946). — 2018. — URL: https://dep.spbstu.ru/userfiles/files/prev/docs/for_students/gia_03_05_2018.pdf (дата обращения: 06.03.2019).

4. Руководство студента СПбПУ по подготовке выпускной квалификационной работы и сопутствующих документов с помощью LaTeX / В. А. Пархоменко [и др.]. — 2018. — URL: https://github.com/ParkhomenkoV/SPbPU-student-thesis-template/blob/master/Author_guide_SPbPU-student-thesis.pdf (дата обращения: 06.03.2019).

5. *Автономова Н. С.* Философский язык Жака Деррида. — М.: Российская политическая энциклопедия (РОССПЭН), 2011. — 510 с. — (Сер.: Российские Пропилеи).

6. *Adams P.* The title of the work // The name of the journal. — 1993. — Vol. 4, no. 2. — P. 201–213.

7. Author and editor guide to prepare and submit the academic SPbPU editions to Clarivate Analytics: Book Citation Index Web of Science / V. Parkhomenko [et al.]. — 2018. — URL: https://github.com/ParkhomenkoV/SPbPU-BCI-template/blob/master/Author_guide_SPbPU-BCI.pdf (visited on 06.03.2019).

8. *Babington P.* The title of the work. Vol. 4. — 3rd ed. — The address: The name of the publisher, 1993. — 255 p. — (Ser.: 10).

9. *Badiou A.* Briefings on Existence: A Short Treatise on Transitory Ontology / ed. and trans. from the French, with an introd., by N. Madarasz. — NY: SUNY Press, 2006. — 190 p. — URL: https://books.google.ru/books?id=7HNkAT%5C_NFksC (visited on 05.12.2017).

10. *Caxton P.* The title of the work. — The address of the publisher, 1993. — 255 p.

11. *Domanov O.* BibLATEX support for GOST standard bibliographies. — URL: https://ctan.org/pkg/biblatex-gost (visited on 06.03.2019).

12. *Domanov O.* Biblatex-GOST examples. — URL: http://ctan.altspu.ru/macros/latex/contrib/biblatex-contrib/biblatex-gost/doc/biblatex-gost-examples.pdf (visited on 06.03.2019).

13. *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, 1993. — (Ser.: 5).

14. *Eston P.* The title of the work // Book title. Vol. 4. — 3rd ed. — The address of the publisher: The name of the publisher, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).

15. *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, 1993. — Chap. 8 — P. 201–213. — (Ser.: 5).

16. *Feuersanger C.*, *Tantau T.* The TikZ and PGF packages. — URL: https://ctan.org/pkg/pgf (visited on 06.03.2019).

17. *Fiorio C.* The algorithm2e package. — URL: https://ctan.org/pkg/algorithm2e (visited on 06.03.2019).

18. *Gainsford P.* The title of the work / The organization. — 3rd ed. — The address of the publisher, 1993. — 255 p.

19. *Ganter B.*, *Wille R.* Formal concept analysis: mathematical foundations. — Springer, Berlin, 1999. — 284 p.

20. *Harwood P.* The title of the work: Master's thesis / Harwood Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.

21. *Isley P.* The title of the work. — 1993.

22. *Joslin P.* The title of the work: diss. ... PhD in Engineering / Joslin Peter. — The address of the publisher: The school where the thesis was written, 1993. — 255 p.

23. *Kotelnikov I. A.*, *Chebotaev P. Z.* LaTeX in Russian. — 3rd ed. — Novosibirsk: Sibiskiy Hronograph, 2004. — 496 p. — URL: http://www.tex.uniyar.ac.ru/doc/kotelnikovchebotaev2004b.pdf (visited on 06.03.2019); (in Russian).

24. *Lambert P.* The title of the work: tech. rep. / The institution that published. — The address of the publisher, 1993. — 255 p. — No. 2.

25. *Marcheford P.* The title of the work. — 1993.

26. MiKTeX web site. — URL: https://miktex.org/ (visited on 06.03.2019).

27. Notes on relation between symbolic classifiers / X. Naidenova [et al.] // CEUR Workshop Proceedings / ed. by K. S. Watson B.W. — 2017. — Vol. 1921. — P. 88–103. — URL: http://ceur-ws.org/Vol-1921/paper9.pdf (visited on 19.12.2017).

28. *Peskov N. V.* Searching for informative fragments of object descriptions in the recognition tasks: diss. ... cand. phys.-math. sci.: 05.13.17 / Peskov Nickolay Vladimirovich. — M., 2004. — 102 p. — (in Russian).

29. SPbPU photo gallery. — URL: http://www.spbstu.ru/media/photo-gallery/ (visited on 06.03.2019).

30. SPbPU-student-thesis-template. — URL: https://github.com/ParkhomenkoV/SPbPU-student-thesis-template (visited on 06.03.2019).

31. TeXstudio web site. — URL: https://www.texstudio.org/ (visited on 06.03.2019).

32. The title of the work. Vol. 4 / ed. by P. Kidwelly. — The organization. The address of the publisher: The name of the publisher, 1993. — 255 p. — (Ser.: 5).

Приложение 1

Краткие инструкции по настройке издательской системы LaTeX

В SPbPU-BCI-template *автоматически выставляются необходимые настройки и в исходном тексте шаблона приведены примеры оформления текстово-графических объектов*, поэтому авторам достаточно заполнить имеющийся шаблон текстом главы (статьи), не вдаваясь в детали оформления, описанные далее. Возможный «быстрый старт» оформления главы (статьи) под Windows следующий[П1.1]:

A. Установка полной версии MikTeX [26]. В процессе установки лучше выставить параметр доустановки пакетов «на лету».
B. Установка TexStudio [31].
C. Запуск TexStudio и компиляция `my_chapter.tex` с помощью команды «Build&View» (например, с помощью двойной зелёной стрелки в верхней панели). *Иногда, для достижения нужного результата необходимо несколько раз скомпилировать документ.*
D. В случае, если не отобразилась библиография, можно
   - воспользоваться командой Tools → Commands → Biber, затем запустив Build&View;
   - настроить автоматическое включение библиографии в настройках Options → Configure TexStudio → Build → Build&View (оставить по умолчанию, если сборка происходит слишком долго): `txs:///pdflatex | txs:///biber | txs:///pdflatex | txs:///pdflatex | txs:///view-pdf`.

В случае возникновения ошибок, попробуйте скомпилировать документ до последних действий или внимательно ознакомьтесь с описанием проблемы в log-файле. Бывает полезным переход (по подсказке TexStudio) в нужную строку в pdf-файле или запрос с текстом ошибке в поисковиках. Наиболее вероятной проблемой при первой компиляции может быть отсутствие какого-либо установленного пакета LaTeX.

В случае корректной работы настройки «установка на лету» все дополнительные пакеты будут скачиваться и устанавливаться в автоматическом режиме. Если доустановка пакетов осуществляется медленно (несколько пакетов за один запуск

[П1.1] Внимание! Пример оформления подстрочной ссылки (сноски).

компилятора), то можно попробовать установить их в ручном режиме следующим образом:

1. Запустите программу: меню → все программы → MikTeX → Maintenance (Admin) → MiKTeX Package Manager (Admin).
2. Пользуясь поиском, убедитесь, что нужный пакет присутствует, но не установлен (если пакет отсутствует воспользуйтесь сначала MiKTeX Update (Admin)).
3. Выделив строку с пакетом (возможно выбрать несколько или вообще все неустановленные пакеты), выполните установку Tools → Install или с помощью контекстного меню.
4. После завершения установки запустите программу MiKTeX Settings (Admin).
5. Обновите базу данных имен файлов Refresh FNDB.

Для проверки текста статьи на русском языке полезно также воспользоваться настройками Options → Configure TexStudio → Language Checking → Default Language. Если русский язык «ru_RU» не будет доступен в меню выбора, то необходимо вначале выполнить Import Dictionary, скачав из интернета любой русскоязычный словарь.

Далее приведены формулы (П1.2), (П1.1), рис.П1.2, рис.П1.1, табл.П1.2, табл.П1.1.

$$\pi \approx 3{,}141. \qquad \text{(П1.1)}$$

Рис.П1.1. Вид на гидробашню СПбПУ [29]

Таблица П1.1

Представление данных для сквозного примера по ВКР [28]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

Таблица П1.2

Представление данных для сквозного примера по ВКР [28]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

П1.1. Параграф приложения

П1.1.1. Название подпараграфа

Название параграфа оформляется с помощью команды `\subsection{...}`.

П1.1.1.1. Название подподпараграфа

$$\pi \approx 3{,}141. \tag{П1.2}$$

Рис.П1.2. Вид на гидробашню СПбПУ [29]

Приложение 2

Некоторые дополнительные примеры

В приложении[П2.1] приведены формулы (П2.2), (П2.1), рис.П2.2, рис.П2.1, табл.**??**, табл.П2.1

$$\pi \approx 3{,}141. \tag{П2.1}$$

Рис.П2.1. Вид на гидробашню СПбПУ [29]

Таблица П2.1

Представление данных для сквозного примера по ВКР [28]

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

П2.1. Подраздел приложения

$$\pi \approx 3{,}141. \tag{П2.2}$$

Рис.П2.2. Вид на гидробашню СПбПУ [29]

[П2.1] Внимание! Пример оформления подстрочной ссылки (сноски).