

Федеральное государственное бюджетное образовательное учреждение  
высшего образования



«Московский государственный технический университет  
имени Н.Э. Баумана (национальный исследовательский  
университет)» (МГТУ им. Н.Э. Баумана)

---

*ФАКУЛЬТЕТ «ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ»  
КАФЕДРА «ТЕОРЕТИЧЕСКАЯ ИНФОРМАТИКА И  
КОМПЬЮТЕРНЫЕ ТЕХНОЛОГИИ»*

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
К КУРСОВОМУ ПРОЕКТУ  
ПО КУРСУ “КОНСТРУИРОВАНИЕ КОМПИЛЯТОРОВ”  
НА ТЕМУ:  
ИНТЕРПРЕТАТОР ФОРМУЛ TeX

Руководитель курсового проекта

Коновалов А.В.

(подпись, дата)

Исполнитель курсового проекта,  
студент группы ИУ9-71

Эсенбаев Н.Э.

(подпись, дата)

Москва 2020

## Содержание

Введение .....	4
1. Процесс работы .....	6
2. Исправления .....	7
2.1. Доступ по индексам.....	7
2.2. Модификация экспоненты.....	7
2.3. Форматирование вывода .....	7
3. Новые операторы .....	8
3.1. Оператор суммирования .....	8
3.2. Оператор произведения.....	9
3.3. Оператор модуля.....	10
3.4. Оператор округления в меньшую сторону .....	10
3.5. Оператор округления в большую сторону.....	11
4. Новые возможности.....	12
4.1. Международная система единиц .....	12
4.2. Реализация .....	12
4.3. Арифметические операции с размерностями .....	13
4.4. Примеры работ с размерностями .....	14
5. Поправки с учетом размерностей .....	15
5.1. Тригонометрические функции .....	15
5.2. Логарифмическая функция .....	15
5.3. Оператор произведения .....	15
5.4. Оператор степени .....	15
5.5. Операторы сравнения .....	16
5.6. Placeholder.....	16
6. Тестирование .....	17
6.1. Сложение и вычитание размерностей .....	17
6.2. Произведение и деление размерностей .....	17
6.3. Сравнение размерностей .....	17
6.4. Пример работы с объявлением собственных размерностей .....	17

6.5. Пример решения задачи с использованием размерностей .....	18
6.6. Оформление лабораторной работы по численным методам .....	19
6.7. Производительность .....	19
ЗАКЛЮЧЕНИЕ .....	21
СПИСОК ЛИТЕРАТУРЫ .....	22
Приложение А .....	23
Приложение В .....	25
Приложение С .....	27
Приложение D .....	28
Приложение E.....	31

## Введение

TeX[1] – система компьютерной вёрстки, разработанная американским профессором информатики Дональдом Кнудом в целях создания компьютерной типографии. TeX один из лучших способов для набора сложных математических формул. В частности, благодаря этим возможностям, TeX популярен в академических кругах, особенно среди математиков и физиков.

TeX широко используется для верстки научной литературы, написания всевозможных лабораторных, рефератов, курсовых и дипломных проектов. При всём при этом, TeX позволяет качественно и красиво оформлять текст в документах.

Препроцессор формул TeX – средство автоматического документирования расчетов в документах формата TeX. В качестве языка расчетов препроцессора используется подмножество языка TeX, дополненное специальными командами. В расчетах поддерживаются основные математические операции, тригонометрические функции, работа с векторами и матрицами, построение двумерных графиков, а также возможность программирования.

Препроцессор состоит из исполняемого файла и файла с определениями вспомогательных команд для TeX (*preproc.tex*). Чтобы использовать препроцессор на документе, в его преамбулу нужно включить файл *preproc.tex*:

```
\input{preproc.tex}
```

В *preproc.tex* собраны объявления дополнительных команд и окружений языка препроцессора. Язык препроцессора является подмножеством языка TeX, дополненным специальными командами. Файл *preproc.tex* требуется включать в документ, чтобы он оставался корректным документом TeX.

Области, предназначенные для обработки препроцессором, выделяются с помощью окружения *preproc*, объявленного в *preproc.tex*:

```

\newenvironment{preproc}
{\begin{equation*}\begin{array}{l}}
{\end{array}\end{equation*}}

```

Одна из из специальных команд – команда `\placeholder[]{}{}`. В аргумент функции `\placeholder[]{}{}` препроцессор записывает результат вычисления выражения.

Препроцессор использует интерфейс командной строки: первый аргумент исполняемого файла — имя исходного документа TeX, второй аргумент — имя документа для записи результата. Если второй аргумент не указан, препроцессор перезапишет исходный документ.

Препроцессор принимает на вход файл с расширением *.tex* в языковой разметке TeX и преобразует формулы  $x = \placeholder[]{}{}$ , вычисляя значения и замещая их в новом порождающем файле.

Таким образом, препроцессор позволяет одновременного производить вычисления в расчетах и документирование отчетов.

Данная работа является продолжением курсовой работы Натальи Стрельниковой[2] и расширяет её в следующих аспектах:

- Добавление и реализация новых операторов
- Добавление физических размерностей и работа с ними.

## 1. Процесс работы

Работа препроцессора производится в несколько этапов.

Нулевой этап предстает из себя подготовка документа для препроцессора:

- Включить в преамбулу документа файл с определениями для препроцессора: `\input{preproc.tex}` (Приложение А).
- Области текста для обработки препроцессором выделить в блоки окружения `\begin{preproc} $body$ \end{preproc}`.

На первом этапе на вход подается документ формата *.tex* и производится нахождение подстрок префиксом которых является `\begin{preproc}` и суффиксом `\end{preproc}`. Эти подстроки отправляются для обработки препроцессору, а остальные строки будут записываться в выходной файл без изменений.

Второй этап состоит из лексического, синтаксического анализа, построение дерева и обхода этого дерева для вычисления значений переменных.

На последнем этапе происходит сохранение нового файла или перезапись входного файла с заполнением *placeholder*-ов вычисленными значениями.

Очевидным является то, что если на вход передан корректный документ TeX, то и документ на выходе будет корректным.

## **2. Исправления**

### **2.1. Доступ по индексам**

После исследования программы было выявлено падение программы при считывании переменных или элементов матрицы с ошибкой segmentation fault.

Анализ программы при помощи отладчика GDB выявил момент падения, и после эта ошибка была устранена.

Причиной этой ошибки была логическая ошибка при обходе дерева, которая считывала нулевой указатель.

### **2.2. Модификация экспоненты**

Препроцессору добавлена поддержка экспоненты. Поддерживаемый синтаксис препроцессором для экспоненты в языке TeX следующий

$\backslash \exp.$

### **2.3. Форматирование вывода**

В данной версии программы исправлен формат вывода чисел. Теперь, при выводе числа имеют точность до пяти значащих цифр после запятой и не печатаются незначащие нули.

### 3. Новые операторы

#### 3.1. Оператор суммирования

При написания математических формул никак не обойтись без оператора суммирования  $\Sigma$ .

Оператор суммирования на языке TeX имеет следующий синтаксис

$$\backslash\text{sum}_{\text{lower}}^{\text{upper}}$$

где *lower*, *upper* – нижняя и верхняя границы суммирования. Для интерпретатора нижняя граница должна задавать индекс суммирования и иметь вид

$$\backslash\text{sum}\{i = \text{lower}\}^{\text{upper}}$$

где *i* – индекс суммирования.

Оператор суммирования является синтаксическим сахаром. Данный синтаксический сахар, описывающий оператор суммы, в синтаксическом дереве заменяется на поддереву, описывающее цикл. Например, данный код

$$\backslash\text{sum}_{i = 0}^{10} \text{expr}$$

будет заменен и представлен синтаксическим поддеревом как на рисунке 1.

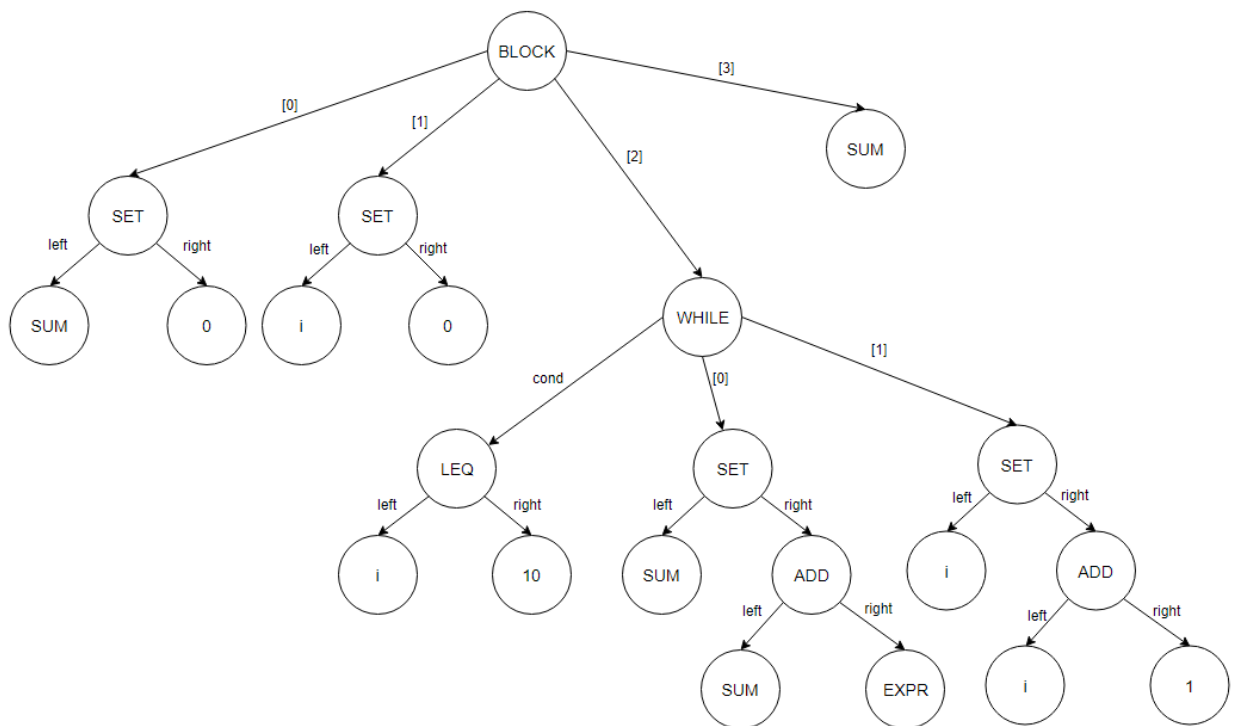


Рисунок 1.



Интерпретатор поддерживает множественное суммирование по нескольким переменным, например  $\sum_{i:=1}^N \sum_{j:=i}^M expr$ .

### 3.2. Оператор произведения

Аналогично был реализован оператор произведения  $\prod$ . На языке TeX оператор суммирования описывается синтаксисом

`\prod_{lower}^{upper}.`

Для обработки с помощью интерпретатора нужно указывать индекс, по которому происходит произведение при указании нижней границы

`\prod_{i = lower}^{upper},`

$i$  – индекс произведения.

Оператор произведения также является синтаксическим сахаром. Синтаксический сахар, описывающий оператор произведения, в синтаксическом дереве заменяется на поддереву, описывающее цикл. Например, данный код

`\prod_{i = 5}^{7} expr`

будет заменен и представлен синтаксическим поддеревом, описанным на рисунке 2.

**Замечание.** Для корректной работы операторов суммы и произведения необходимо указывать обе границы, нижнюю и верхнюю. При отсутствии одной из границ, препроцессор прервет свою работу и выдаст синтаксическую ошибку.

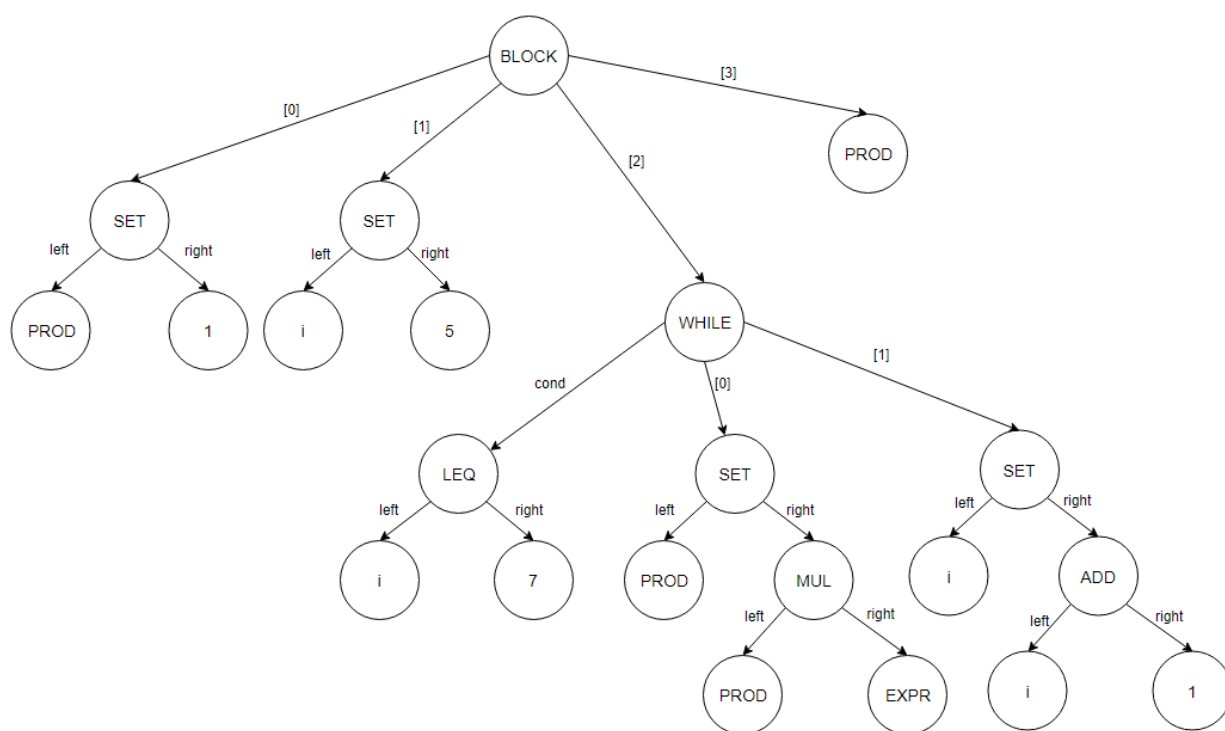


Рисунок 2.

### 3.3. Оператор модуля

В расчетах часто возникает необходимость вычисления абсолютного значения числа. Для покрытия этой необходимости был введен оператор модуля. На языке TeX он имеет следующий синтаксис:

`\abs{...}`.

Оператор модуля реализована стандартным образом как функция от числа по ниже следующему правилу:

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

### 3.4. Оператор округления в меньшую сторону

Данный оператор производит округление числа до ближайшего целого числа в меньшую сторону и определено по правилу

$$\lfloor x \rfloor = \max_{n \in \mathbb{Z}} \{n \mid n \leq x\}$$

Для оператора округления в меньшую сторону в файл с определениями для препроцессора был добавлен специальный макрос, описывающийся на языке TeX следующим образом

`\DeclarePairedDelimiter{\floor}{\lfloor}{\rfloor}`.

Синтаксис, поддерживаемый препроцессором описан ниже

`\floor*{}`.

Пример использования

`\floor*{4.1} = 4.0000.`

### 3.5. Оператор округления в большую сторону

Данный оператор производит округление числа до ближайшего целого числа в большую сторону и определено по правилу

$$[x] = \min_{n \in \mathbb{Z}} \{n \mid n \geq x\}$$

Для оператора округления в большую сторону в файл с определениями для препроцессора был добавлен специальный макрос, описывающийся на языке TeX следующим образом

`\DeclarePairedDelimiter{\ceil}{\lceil}{\rceil}.`

Синтаксис, поддерживаемый препроцессором описан ниже

`\ceil*{}`.

Пример использования

`\ceil*{5.3} = 6.0000.`

**Замечание.** Для корректного отображения макросов округления в меньшую и большую сторону необходим пакет *mathtools*.

#### **4. Новые возможности**

При документировании и написании отчетов в расчетах часто встречаются значения имеющие физические размерности. В связи с этим было решено реализовать обработку системы единиц значений при вычислении расчетов.

##### **4.1. Международная система единиц**

Международная система единиц, СИ[3] – система единиц физических величин, современный вариант метрической системы. СИ является наиболее широко используемой системой единиц в мире – как в повседневной жизни, так и в науке и технике. В настоящее время СИ принята в качестве основной системы единиц большинством стран мира и почти всегда используется в области техники, даже в тех странах, в которых в повседневной жизни используются традиционные единицы.

СИ определяет семь основных единиц физических величин и производные единицы. СИ также устанавливает стандартные сокращённые обозначения единиц и правила записи производных единиц.

Основные единицы: килограмм, метр, секунда, ампер, кельвин, моль и кандела. В рамках СИ считается, что эти единицы имеют независимую размерность, то есть ни одна из основных единиц не может быть получена из других.

##### **4.2. Реализация**

В интерпретаторе реализованы основные единицы СИ: длина(m), масса(kg), время(s), сила электрического тока(A), термодинамическая температура(K), количество вещества(mol), сила света(cd).

Каждое значение при вычислении содержит в себе информацию о размерности в качестве вектора из 7 элементов. Класс описывающий значение был расширен и имеет следующую структуру, описанной в листинге 1.

Листинг 1.

```
class Value {
public:
    typedef enum Type {
        DOUBLE, MATRIX, FUNCTION
    } Type;

...

private:
    Type _type;
    union {
        double _double_data;
        std::vector<std::vector<Value>>
*_matrix_data;
        Func *_function_data;
    };
    std::array<int, 7> dimention;
public:
    ...
    double get_double() const;
    Matrix &get_matrix() const;
    Func *get_function() const;
    std::array<int, 7> get_dimention() const;
...
}
```

#### **4.3. Арифметические операции с размерностями**

При арифметических операциях размерности влияют на работу программы. Умножение значений означает сложение векторов размерностей покомпонентное. Деление – разность векторов размерностей.

При сложении значений происходит проверка совпадения размерностей. Если в выражении складывается два значения с разными размерностями программа сообщает об ошибке и прекращает дальнейшую работу. В случае совпадения размерностей сумма значений имеет аналогичную размерность, как и у слагаемых. Таким же образом происходит вычитание значений.

#### **4.4. Примеры работы с размерностями**

Помимо реализованных стандартных единиц СИ, мы можем поверх них определять свои размерности. Например, определим размерность часов, выраженную через секунды

$$\text{hour} := 3600 \cdot \text{s}.$$

Далее можем определить день, выраженный через часы

$$\text{day} := 24 \cdot \text{hour}.$$

При попытке сложения, например, дня и метра

$$1 \cdot \text{day} + 1 \cdot \text{m}$$

мы получим ошибку выполнения препроцессора с сообщением “*The addition of different dimensions*”.

## 5. Поправки с учетом размерностей

### 5.1. Тригонометрические функции

Тригонометрические функции принимают в качестве аргумента значение в радианах. Но при выражении радиана через основные единицы СИ получается, что радиан – безразмерная величина. Таким образом, при работе с тригонометрическими функциями проводится проверка размерности аргумента тригонометрической функции. В случае, если аргумент имеет какую-либо размерность, то препроцессор сообщит об это с ошибкой и сообщением “*\$func\$ gets only dimensionless argument*”.

### 5.2. Логарифмическая функция

Аналогично, логарифмическая функция принимает аргументы только безразмерного значения и препроцессор выдает такую же ошибку как в пункте 5.1.

### 5.3. Оператор произведения

Оператор произведения выполняет вычисления только при условии, что элементы произведения не имеют размерности. В ином случае, препроцессор выдаст ошибку с сообщением “*Element of product is not allowed to be dimensional*”.

### 5.4. Оператор степени

Показатель степени или корня — это число раз (штук), которое надо взять основание степени и умножить на себя. Как мы уже отмечали, «разы» («штуки») безразмерны. Мы можем возвести 3 м в степень 3, но не можем возвести 3 в степень 3 м, так как 3 м = 300 см — длина (размерная величина, которая зависит от единицы измерения) и в штуки (безразмерные единицы, которые от единицы измерения не зависят) не может быть переведена.

Исходя из вышенаписанного, аргументом оператора степени должно быть безразмерное значение. При условии, что значение оператора степени размерное, препроцессор выдаст ошибку с сообщением “*Dimensional power of number is not defined*”.

По этой же причине, аргументом степени в которую возводится число должно быть целочисленным. В ином случае, препроцессор выдаст ошибку с сообщением “*Float power of number is not allowed*”.

### 5.5. Операторы сравнения

Размерная величина может быть «мала» или «велика» только по сравнению с другой величиной той же размерности. Учитывая это, при произведении сравнения значений учитываются размерности, т.е. размерности должны совпадать. Если это условие нарушается, препроцессор выдаст ошибку с сообщением “*Comparison of different dimensions*”.

### 5.6. Placeholder

В связи с добавлением размерностей, синтаксис заменителей(*placeholder*-ов) был расширен и теперь имеет следующее макроопределение с использованием пакета *ifthen*.

```
\usepackage{ifthen}
\newcommand{\placeholder}[2][]{
\ifthenelse{ \equal{#1}{} }
{ \color{blue}#2 }
{ \color{blue}#2 \cdot #1 } }
```

Теперь синтаксис заменителей представляет возможность указать необязательный параметр, на который будет поделена размерность значения перед выводом.



## 6. Тестирование

### 6.1. Сложение и вычитание размерностей

Рассмотрим корректность работы препроцессора при сложении и вычитании размерностей.

$$1 \cdot m + 10 \cdot m = 11 \cdot m$$

$$10 \cdot s - 5 \cdot s = 5 \cdot s$$

Сложение и вычитание разных размерностей выдает ошибку

$$15 \cdot m + 5 \cdot kg =$$

Ошибка препроцессора – “*The addition of different dimensions*”.

### 6.2. Произведение и деление размерностей

Операции произведение и деления размерностей работают корректно:

$$12 \cdot m \cdot 10 \cdot m = 120 \cdot m^2$$

$$3 \cdot m / 2 \cdot s = 1.5000 \cdot \frac{m}{s}$$

### 6.3. Сравнение размерностей

Сравнение размерностей работают корректно:

$$1 \cdot s < 2 \cdot s = 1$$

$$1 \cdot \frac{m \cdot kg^2}{s^2} \leq 5 \cdot \frac{m \cdot kg^2}{s^2} = 1$$

$$3 \cdot \frac{m \cdot kg^2}{s^2} \geq 5 \cdot \frac{m \cdot kg^2}{s^2} = 0$$

Сравнение разных размерностей выдает ошибку

$$1 \cdot \frac{m \cdot kg^2}{s^2} < \frac{kg \cdot m^2}{s^3} =$$

Ошибка препроцессора – “*Comparison of different dimensions*”.

### 6.4. Пример работы с объявлением собственных размерностей

В листинге 2 приведен пример работы с объявлением переменной с размерностью, арифметических операций значений с размерностью и использование необязательного параметра *placeholder*-а.

Листинг 2: фрагмент документа TeX до обработки

```
\begin{preproc}
```

```

N := kg \cdot m \cdot s^{-2}
J := N \cdot m
W := J / s
hour := 3600 \cdot s
kWh := 1000 \cdot W \cdot hour
cal := 4.1868 \cdot J
kcal := 1000 \cdot cal
kWh = \placeholder[kcal]{}
\end{preproc}

```

Далее рассмотрим, как препроцессор изменил фрагмент документа из Листинга 2.

После обработки препроцессором этот фрагмент будет иметь вид:

Листинг 3: фрагмент документа TeX после обработки

```

\begin{preproc}
N := kg \cdot m \cdot s^{-2}
J := N \cdot m
W := J / s
hour := 3600 \cdot s
kWh := 1000 \cdot W \cdot hour
cal := 4.1868 \cdot J
kcal := 1000 \cdot cal
kWh = \placeholder[kcal]{859.8452}
\end{preproc}

```

## 6.5. Пример решения задачи с использованием размерностей

В приложении В приведен исходный код примера задачи с использованием почти всех базовых размерностей СИ.

При решении задачи выполняются различные операции над размерностями, в частности используется операция округления в меньшую сторону.

Документ после обработки препроцессора приведен в приложении С.

## 6.6. Оформление лабораторной работы по численным методам

Главное целью препроцессора является одновременное документирование и производство вычислений в расчетах. Для демонстрации этого в приложении D приведен исходный код оформления одной из работ по численным методам. Данная работа посвящена вычислению значения интеграла при помощи метода центральных прямоугольников.

Результат работы препроцессора для данной лабораторной работы приведен в приложении E.

## 6.7. Производительность

Последней частью тестирования является производительность.

При тестировании производилось сравнение между программами с поддержкой размерностей и без.

Тестовым файлом был прогон цикла в 10000 итераций. Исходный код тестовых файлов приведен в листинге 4 и 5.

Листинг 4.

```
\begin{preproc}
i := 0
step := 1
limit := 10000
x := \while{i < limit} \begin{block}
i := i + step
\end{block}
x = \placeholder{}
\end{preproc}
```

Листинг 5.

```
\begin{preproc}
i := 0 \cdot s
step := 1 \cdot s
limit := 10000 \cdot s
x := \while{i < limit} \begin{block}
i := i + step
```

```

\end{block}
x = \placeholder{}
\end{preproc}

```

Листинги 4 и 5 идентичны, за исключением наличия размерности у переменной в листинге 5.

Программы были скомпилированы при помощи компилятора `g++` с ключами `-O2 -fvectorize`.

По результатам проведенных тестов были получены данные, приведенные в таблице 1.

Таблица 1: результаты замеров производительности

<i>Время выполнения теста № (сек.)</i>	<i>С поддержкой размерности</i>	<i>Без поддержки размерности</i>
1	0.003770	0.000546
2	0.003796	0.000561
3	0.003837	0.000540
4	0.003784	0.000589
5	0.003797	0.000553
<i>Среднее время выполнения</i>	0.003896	0.000558

Тестирование проводилось на машине со следующими техническими характеристиками:

- Оперативная память – 8 Гб
- Intel® Core™ i5-6300HQ CPU @ 2.30GHz × 4
- Операционная система – Ubuntu 64-bit

## ЗАКЛЮЧЕНИЕ

В продолжении курсовой работы были проведены следующие улучшения препроцессора для TeX:

- Исправлена работа с доступом к переменным и элементам матриц по индексу
- Улучшено форматирование вывода чисел
- Добавлены операторы: суммирования, произведения, модуля, округления в меньшую сторону, округления в большую сторону
- Добавлена возможность работы со значениями, имеющими физические размерности

По результатам тестирования производительности, вычисления с размерностями занимают в 7 раз больше времени выполнения программы, чем расчет без размерностей.

## СПИСОК ЛИТЕРАТУРЫ

1. Официальная документация LaTeX

Режим доступа:

<https://www.latex-project.org/help/documentation/>

2. Курсовой проект Натальи Стрельниковой 2016 год

Режим доступа:

<https://github.com/bmstu-iu9/tex-preprocessor>

3. Международная система единиц

Режим доступа:

[https://ru.wikipedia.org/wiki/Международная\\_система\\_единиц](https://ru.wikipedia.org/wiki/Международная_система_единиц)

# ПРИЛОЖЕНИЕ А

Файл *preproc.tex* с определениями для препроцессора

```
\usepackage{ifthen}
\newcommand{\placeholder}[2][ ]{
\ifthenelse{ \equal{#1}{} }
    {\color{blue}#2}
    {\color{blue}#2 \cdot #1}}
\newcommand{\ifexpr}[1]{\{\bf if\enspace}#1\enspace}
\newcommand{\when}{\{\enspace\bf when\enspace}}
\newcommand{\otherwise}{\{\enspace\bf otherwise}}
\newcommand{\while}[1]{\{\bf while\enspace}#1\enspace}
\newcommand{\true}{true}
\newcommand{\false}{false}
\newcommand{\abs}[1]{|#1|}
\DeclarePairedDelimiter{\ceil}{\lceil}{\rceil}
\DeclarePairedDelimiter{\floor}{\lfloor}{\rfloor}

\newenvironment{preproc}
{\color{red}\begin{equation*}\begin{array}{l}}
{\end{array}\end{equation*}}

\newenvironment{block}
{\begin{array}{l}}
{\end{array}}

\newenvironment{caseblock}
{\begin{array}{l}}
{\end{array}}

\newcommand{\range}[3][ ]{%
\ifthenelse{\isempty{#1}}{[#2:#3]}{[#2:#3:#1]}%
}
```

```

\newcommand{\graphic}[3]{
\color{blue}
\begin{tikzpicture}
\begin{axis}[
xmajorgrids,
ymajorgrids,
axis lines=middle,
x label style={at={(axis description cs:0.5,-0.1)},anchor=north},
ylabel style={above left},
xlabel=\text{\#1(\#2)},
]
\addplot[thin, mark = none] coordinates{
#3
};
\end{axis}
\end{tikzpicture}}

```



# ПРИЛОЖЕНИЕ В

## Задача с использованием размерностей (исходный код)

Задача.

Дана комната площадью  $15 \text{ м}^2$  и высотой потолка  $2,3 \text{ м}$ . Изначальная температура воздуха в комнате  $10^\circ\text{C}$ . Для повышения температуры в комнате находятся несколько электрообогревателей мощностью  $300 \text{ Вт}$  каждый. Нужно найти:

1. Сколько обогревателей, включённых одновременно, не приведут к пожару, если проводка выдерживает максимум  $10 \text{ А}$ ?

```
\begin{preproc}
W := 1 \cdot \text{kg} \cdot \text{m}^{\{2\}} \cdot \text{s}^{\{-3\}} \\\
V := 1 \cdot \text{kg} \cdot \text{m}^{\{2\}} \cdot \text{s}^{\{-3\}} \cdot \text{A}^{\{-1\}} \\\
I_{\text{max}} := 10 \cdot \text{A} \\\

P := 300 \cdot W \\\
U := 220 \cdot V \\\
I := \frac{P}{U} \\\

n := \text{floor}\{I_{\text{max}} / I\} \\\
n = \text{placeholder}\{\}
\end{preproc}
```

2. За какое время (в минутах) эти обогреватели нагреют воздух в комнате до  $25^\circ\text{C}$ ? (теплоёмкость воздуха считать равной  $1,3 \text{ кДж}/(\text{м}^3 \cdot \text{K})$ )

```
\begin{preproc}
J := \text{kg} \cdot \text{m}^{\{2\}} \cdot \text{s}^{\{-2\}} \\\
kJ := 1000 \cdot J \\\
t_{\text{1}} := (10 - 273.15) \cdot \text{K} \\\
t_{\text{2}} := (25 - 273.15) \cdot \text{K} \\\
c := 1.3 \cdot \frac{\text{kJ}}{\text{m}^{\{3\}} \cdot \text{K}} \\\

S := 15 \cdot \text{m}^{\{2\}} \\\
h := 2.3 \cdot \text{m} \\\
```

```
V := S \cdot h \\\
```

```
Q := c \cdot V \cdot (t_{\text{2}} - t_{\text{1}}) \\\
```

```
min := 60 \cdot s \\\
```

```
t := Q / (n \cdot P) \\\
```

```
t = \placeholder[min]{}
```

```
\end{preproc}
```

# ПРИЛОЖЕНИЕ С

## Задача с использованием размерностей (после обработки препроцессора)

Задача.

Дана комната площадью  $15 \text{ м}^2$  и высотой потолка  $2,3 \text{ м}$ . Изначальная температура воздуха в комнате  $10 \text{ }^\circ\text{C}$ . Для повышения температуры в комнате находятся несколько электрообогревателей мощностью  $300 \text{ Вт}$  каждый. Нужно найти:

1. Сколько обогревателей, включённых одновременно, не приведут к пожару, если проводка выдерживает максимум  $10 \text{ А}$ ?

$$W := 1 \cdot kg \cdot m^2 \cdot s^{-3}$$

$$V := 1 \cdot kg \cdot m^2 \cdot s^{-3} \cdot A^{-1}$$

$$I_{\max} := 10 \cdot A$$

$$P := 300 \cdot W$$

$$U := 220 \cdot V$$

$$I := \frac{P}{U}$$

$$n := \lfloor I_{\max}/I \rfloor$$

$$n = 7$$

2. За какое время (в минутах) эти обогреватели нагреют воздух в комнате до  $25 \text{ }^\circ\text{C}$ ? (теплоёмкость воздуха считать равной  $1,3 \text{ кДж}/(\text{м}^3 \cdot \text{К})$ )

$$J := kg \cdot m^2 \cdot s^{-2}$$

$$kJ := 1000 \cdot J$$

$$t_1 := (10 - 273.15) \cdot K$$

$$t_2 := (25 - 273.15) \cdot K$$

$$c := 1.3 \cdot \frac{kJ}{m^3 \cdot K}$$

$$S := 15 \cdot m^2$$

$$h := 2.3 \cdot m$$

$$V := S \cdot h$$

$$Q := c \cdot V \cdot (t_2 - t_1)$$

$$min := 60 \cdot s$$

$$t := Q/(n \cdot P)$$

$$t = 5.33929 \cdot min$$

# ПРИЛОЖЕНИЕ D

Лабораторная работа по численным методам (исходный код)

## Постановка задачи.

Дано: функция  $f(x)$ . Нужно вычислить значение интеграла функции  $f(x)$  на заданном отрезке  $[a, b]$ .

## Решение.

Метод центральных прямоугольников: Разобьем этот отрезок на  $n$  равных отрезков длиной  $h = \frac{b-a}{n}$ . Получаем  $x_i = a + i \cdot h$ .

Тогда интеграл высчитывается по следующей формуле:

$$I = \int_a^b f(x)dx \approx I^* = h \cdot \sum_{i=1}^n f(a + i \cdot h - \frac{h}{2})$$

```
\begin{preproc}
methodOfRectanhle(f, a, b, n) := \begin{block}
h := (b - a) / n\
h \cdot \sum_{i = 1}^n f(a + i \cdot h - \frac{h}{2})
\end{block}
\end{preproc}
```

## Вычисление интеграла с учетом погрешности

Формула приближенного значение интеграла с помощью метода с шагом  $h$  с уточнением по Ричардсону:

```
\begin{preproc}
richardson(I_\text{1}, I_\text{2}, k) := \frac{I_\text{1} - I_\text{2}}{1 - 2^k}
\end{preproc}
\begin{preproc}
```

```

calcIntegral(method, func, k, a, b, epsilon) := \begin{block}
\while{\abs{r} > epsilon} \\\ \begin{block}
  n := n \cdot 2\\
  h2 := h\\
  h := method(func, a, b, n)\\
  r := richardson(h, h2, k)\\
  iter := iter + 1
\end{block}
\end{block}
\end{preproc}

```

## Тестирование

Тестовая функция  $f(x) = \exp^x$  на отрезке  $[0, 1]$ . Заданная точность  $\varepsilon = 0.001$

$$\int_0^1 \exp^x dx = \exp \Big|_0^1 = 1,71828182845...$$

```

\begin{preproc}
f(x) := \exp^x \\\
a := 0 \\\
b := 1 \\\
k := 2 \\\
\\
n := 1\\
r := 10\\
iter := 0\\
h := 0\\
h2 := 0

calcIntegral(methodOfRectanhle, f, k, a, b, 0.001) \\\

result := h = \placeholder{} \\\
withRuch := h + \abs{r} = \placeholder{} \\\
iter = \placeholder{}

```

\end{preproc}

# ПРИЛОЖЕНИЕ Е

Лабораторная работа по численным методам  
(после обработки препроцессора)

Дано: функция  $f(x)$ . Нужно вычислить значение интеграла функции  $f(x)$  на заданном отрезке  $[a, b]$ .

Решение.

Метод центральных прямоугольников: Разобьем этот отрезок на  $n$  равных отрезков длиной  $h = \frac{b-a}{n}$ . Получаем  $x_i = a + i \cdot h$ .

Тогда интеграл высчитывается по следующей формуле:

$$I = \int_a^b f(x)dx \approx I^* = h \cdot \sum_{i=1}^n f(a + i \cdot h - \frac{h}{2})$$

$$\textit{methodOfRectanhle}(f, a, b, n) := \left| \begin{array}{l} h := (b - a)/n \\ h \cdot \sum_{i=1}^n f(a + i \cdot h - \frac{h}{2}) \end{array} \right.$$

## Вычисление интеграла с учетом погрешности

Формула приближенного значение интеграла с помощью метода с шагом  $h$  с уточнением по Ричардсону:

$$\textit{richardson}(I_1, I_2, k) := \frac{I_1 - I_2}{1 - 2^k}$$

$$\textit{calcIntegral}(\textit{method}, \textit{func}, k, a, b, \textit{epsilon}) := \left| \begin{array}{l} \textbf{while } |r| > \textit{epsilon} \\ \quad n := n \cdot 2 \\ \quad h2 := h \\ \quad h := \textit{method}(\textit{func}, a, b, n) \\ \quad r := \textit{richardson}(h, h2, k) \\ \quad \textit{iter} := \textit{iter} + 1 \end{array} \right.$$

# Тестирование

Тестовая функция  $f(x) = \exp^x$  на отрезке  $[0, 1]$ . Заданная точность  $\varepsilon = 0.001$

$$\int_0^1 \exp^x dx = \exp \Big|_0^1 = 1,71828182845\dots$$

$f(x) := \exp^x$

$a := 0$

$b := 1$

$k := 2$

$n := 1$

$r := 10$

$iter := 0$

$h := 0$

$h2 := 0$

$calcIntegral(methodOfRectanhle, f, k, a, b, 0.001)$

$result := h = 1.71800$

$withRuch := h + |r| = 1.71828$

$iter = 4$