

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
имени М.В. ЛОМОНОСОВА  
**Факультет вычислительной математики и кибернетики**

Кафедра нелинейных динамических систем и процессов управления

Препроцессор для среды LaTeX с использованием символьных  
вычислений

Курсовая работа

Королев Александр Михайлович  
Научный руководитель Роговский Александр Игоревич

# Содержание

1	Введение	3
2	Цель	4
3	Исходный документ	5
4	Парсер	6
5	Заключение	7

# 1 Введение

Среда LaTeX активно используется для создания отчетов в разных математических дисциплинах. Зачастую, приходится описывать какие-либо системы с помощью длинных формул, объемных матриц и т.д. Стандартные методы языка LaTeX подразумевают ручное переписывание всех математических объектов, что сильно увеличивает код программы. Так же, невозможно повторное использование ранее введенных объектов без их полного копирования.

В данной работе будет рассмотрен разрабатываемый препроцессор языка LaTeX, позволяющий использовать в коде программы код других языков программирования, таких как Python, Maxima.

## 2 Цель

**Цель** - разработка программы, состоящую из двух частей:

- 1) “Парсер” - в исходном документе выделяет код на языке python или maxima, выполняет его, и результат выполнения подставляет в документ на языке LaTeX. В рамках курсовой работы рассматривается только "парсер".
- 2) “Анализатор” - некоторый фреймворк для удобного использования систем компьютерной математики при наборе математических текстов.

**Требования** к “парсеру”:

- 1) Документ должен корректно собираться даже если парсер не отработал;
- 2) Корректная обработка ошибок: в сообщении об ошибке должно указываться место в исходном документе;
- 3) Возможность добавление как однострочного кода, так и многострочного; как с выводом, так и без него;
- 4) Возможность использования в коде любых символов, допустимых для языка программирования;
- 5) Возможность добавлять код “inline”, то есть так, чтобы до и/или после кода в строке следовал текст документа.
- 6) Для работы парсера не требуется подключать никаких дополнительных пакетов latex.
- 7) Незначительное время работы парсера: не более 2 x время сборки документа.
- 8) Желательно иметь возможность перейти на другой язык программирования.

### 3 Исходный документ

Предлагаются следующие правила оформления встраиваемого кода:

- 1) `code{Text}` - в фигурных скобках пишется код, который будет выполнен, но результат не будет подставлен в документ;
- 2) `codep{Text}` - код будет выполнен и результат последней строки данной команды подставится вместо нее.

Так же эти команды можно использовать `inline`, т.е. делать вставку прямо в тексте.

Например, следующий код:

```
\code{
z = abs(-1)
}
Absolute of -1 is \codep{z}
```

Преобразуется в:

```
Absolute of -1 is 1
```

Если есть необходимость в сборке исходного документа до отработки парсера, то для корректной сборки PDF файла нужно вставить в начало документа следующие команды:

```
\long\def\code#1{}
\long\def\codep#1{}
```

- так получится игнорировать любой код внутри введенных команд.

## 4 Парсер

Чтобы сгенерировать из исходного документа готовый документ на языке LaTeX, необходимо ввести следующую команду в терминал:

`python3 parser.py file_name`, где `file_name` - путь к исходному файлу. Будет сгенерирован новый файл `file_name_new`, готовый к сборке.

Пока “парсер” обрабатывает только код на языке python, в дальнейшем будет реализована возможность перехода на язык maxima.

## 5 Заключение

В данной работе был рассмотрен реализованная программа "парсер выполнены основные требования, поставленные в начале. В дальнейшем будет реализована часть программы "анализатор", а так же будут доделаны все недочеты по части "парсера".