LaTeX – наиболее популярный набор макрорасширений системы компьютерной вёрстки TeX, который облегчает набор сложных документов. Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая набор текста на нескольких языках, нумерацию разделов и формул, перекрестные ссылки, размещение иллюстраций и таблиц на странице, ведение библиографии и др.

Описание проделанной работы.

Задача – кода программы, переводящей текст формата «.txt» в формат «.tex». При этом должна создаваться структура LaTeX-документа, должен появляться соответствующий синтаксис для вставки изображений и формул (например, «sqrt(x)» заменяется на «$\sqrt(x)$») и т. д. Таким образом должен создаваться итоговый текст, который необходимо подгрузить на платформу Overleaf (или любую другую платформу, позволяющую работать с LaTeX) вместе со скриншотами экрана или собрать это все с помощью любой локальной программы (например, Texstudio).

В итоге должен получиться черновой конспект видеолекции. Далее более подробно рассмотрим проделанную работу и полученные результаты.

Что касается создания программы, то было написано много функций как основных, так и вспомогательных. К основным функциям относятся следующие:

- makes_screens_rows – выносит названия скриншотов в отдельные строки (необходимо для дальнейшей их обработки под синтаксис Latex, а также для удобства);
- convetrs_text_latex – переводит обычный текст в LaTeX (создает структуру LaTeX документа);
- convetrs_mathtext_latex – переводит математический текст и символы в LaTeX (число «123» преобразуется в «123» и т. д.)

- makes_good_formulas – преобразует формулы и операции (чтобы LaTeX распознал функцию «sqrt» – извлечение квадратного корня, мало просто заключить ее в специальный символы «$», необходимо преобразовать ее к следующему виду: «$\sqrt{выражение}$». Аналогичная ситуация обстоит, например, с тригонометрическими формулами, в то время как с операцией деления все намного сложнее).

Список всех функций можно видеть ниже в таблице 1:

Номер функции	Название
1.	split_text
2.	join_text
3.	makes_good_gaps
4.	makes_screens_rows
5.	convetrs_text_latex
6.	brackets_after_symbol
7.	brackets_before_symbol
8.	no_brackets_after_symbol
9.	no_brackets_before_symbol
10.	get_indexes
11.	get_indexes_mod2
12.	convetrs_mathtext_latex
13.	makes_good_formulas
14.	join_punctuation

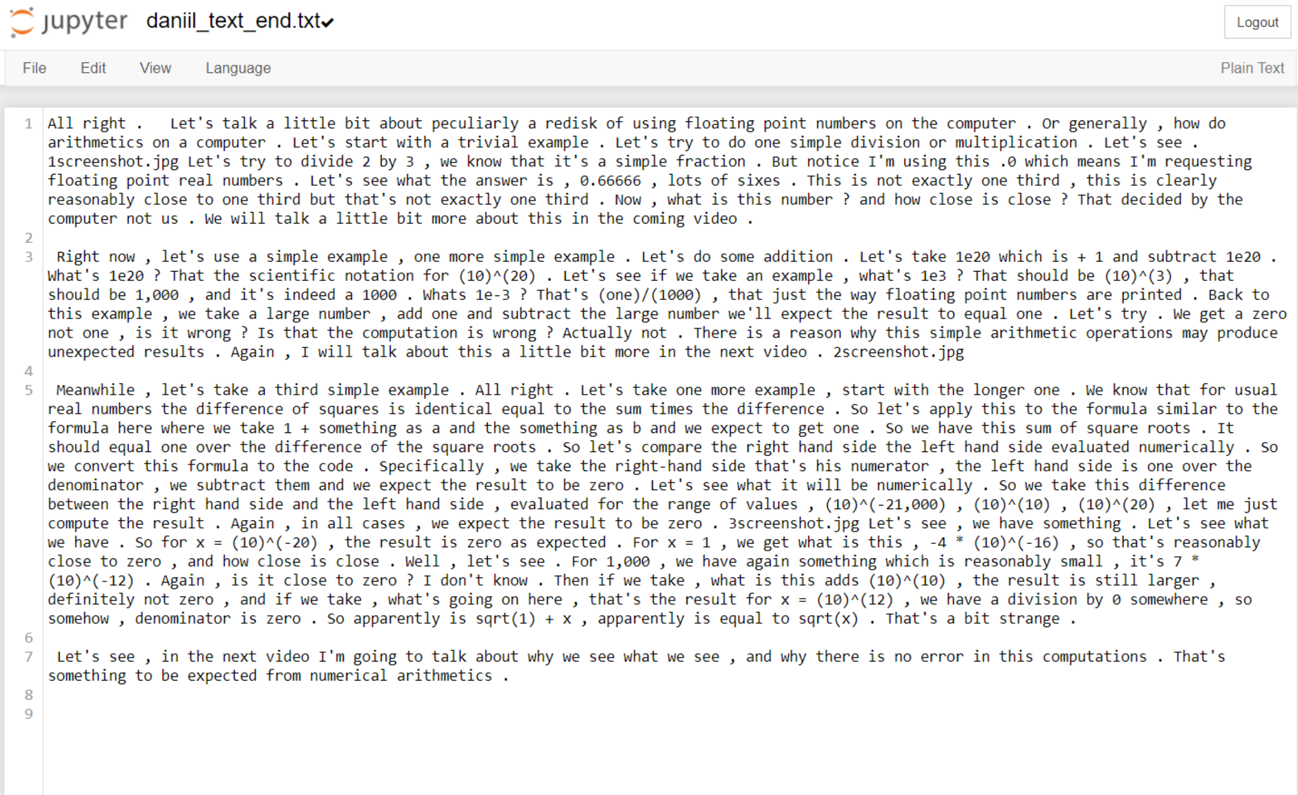
Таб. 1. Список всех функций

В результате работы программы создаются 6 файлов, 5 из которых являются вспомогательными и появляются по ходу выполнения кода: timofey_text1.txt, timofey_text2.txt, timofey_text3.txt, timofey_text4.txt, timofey_text5.txt. Они удобны для корректировки кода на каждом этапе работы программы. Входящий файл:

daniil_text_end.txt, выходящий файл: timofey_text_end.tex. Далее рассмотрим ход программы подробнее.

Иллюстрация работы программы.

На следующем примере рассмотрим ход работы программы и ее результат. На вход поступает файл daniil_text_end.txt, содержимое которого изображено на рисунке 1.



jupyter daniil_text_end.txt

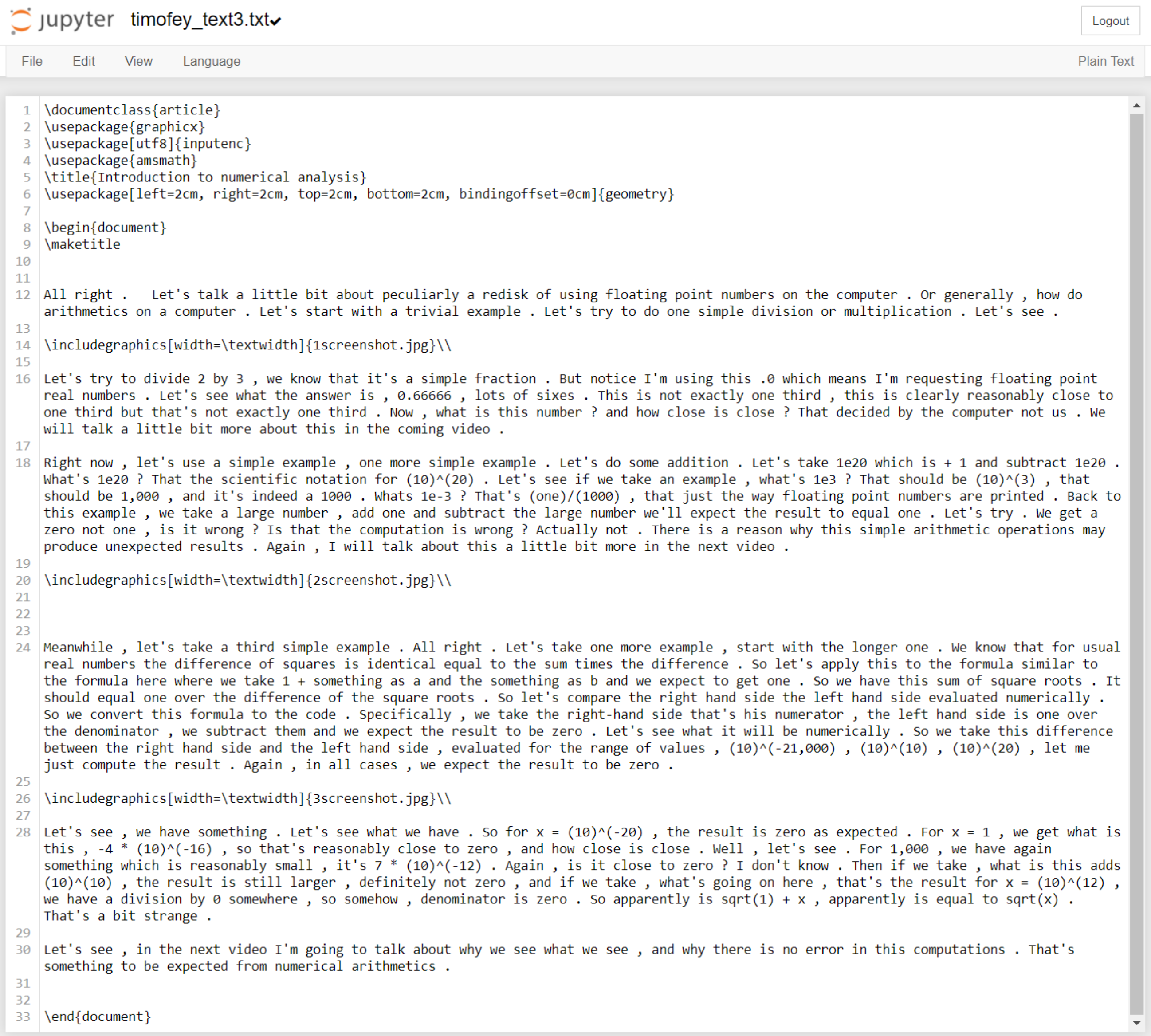
Logout

File Edit View Language Plain Text

```
All right .   Let's talk a little bit about peculiarly a redisk of using floating point numbers on the computer . Or generally , how do arithmetics on a computer . Let's start with a trivial example . Let's try to do one simple division or multiplication . Let's see . 1screenshot.jpg Let's try to divide 2 by 3 , we know that it's a simple fraction . But notice I'm using this .0 which means I'm requesting floating point real numbers . Let's see what the answer is , 0.66666 , lots of sixes . This is not exactly one third , this is clearly reasonably close to one third but that's not exactly one third . Now , what is this number ? and how close is close ? That decided by the computer not us . We will talk a little bit more about this in the coming video .

 Right now , let's use a simple example , one more simple example . Let's do some addition . Let's take 1e20 which is + 1 and subtract 1e20 . What's 1e20 ? That the scientific notation for (10)^(20) . Let's see if we take an example , what's 1e3 ? That should be (10)^(3) , that should be 1,000 , and it's indeed a 1000 . Whats 1e-3 ? That's (one)/(1000) , that just the way floating point numbers are printed . Back to this example , we take a large number , add one and subtract the large number we'll expect the result to equal one . Let's try . We get a zero not one , is it wrong ? Is that the computation is wrong ? Actually not . There is a reason why this simple arithmetic operations may produce unexpected results . Again , I will talk about this a little bit more in the next video . 2screenshot.jpg

 Meanwhile , let's take a third simple example . All right . Let's take one more example , start with the longer one . We know that for usual real numbers the difference of squares is identical equal to the sum times the difference . So let's apply this to the formula similar to the formula here where we take 1 + something as a and the something as b and we expect to get one . So we have this sum of square roots . It should equal one over the difference of the square roots . So let's compare the right hand side the left hand side evaluated numerically . So we convert this formula to the code . Specifically , we take the right-hand side that's his numerator , the left hand side is one over the denominator , we subtract them and we expect the result to be zero . Let's see what it will be numerically . So we take this difference between the right hand side and the left hand side , evaluated for the range of values , (10)^(-21,000) , (10)^(10) , (10)^(20) , let me just compute the result . Again , in all cases , we expect the result to be zero . 3screenshot.jpg Let's see , we have something . Let's see what we have . So for x = (10)^(-20) , the result is zero as expected . For x = 1 , we get what is this , -4 * (10)^(-16) , so that's reasonably close to zero , and how close is close . Well , let's see . For 1,000 , we have again something which is reasonably small , it's 7 * (10)^(-12) . Again , is it close to zero ? I don't know . Then if we take , what is this adds (10)^(10) , the result is still larger , definitely not zero , and if we take , what's going on here , that's the result for x = (10)^(12) , we have a division by 0 somewhere , so somehow , denominator is zero . So apparently is sqrt(1) + x , apparently is equal to sqrt(x) . That's a bit strange .

 Let's see , in the next video I'm going to talk about why we see what we see , and why there is no error in this computations . That's something to be expected from numerical arithmetics .
```

Рис. 1. Файл daniil_text_end.txt

Далее появляются дополнительные файлы, по которым мы можем отследить, как преобразуется содержимое. Приведем некоторые из их. На рисунке 2 изображено содержимое файла timofey_text3.txt. Видно, что уже создана структура LaTeX-документа, скриншоты заключены в специальную оболочку.



jupyter timofey_text3.txt

Logout

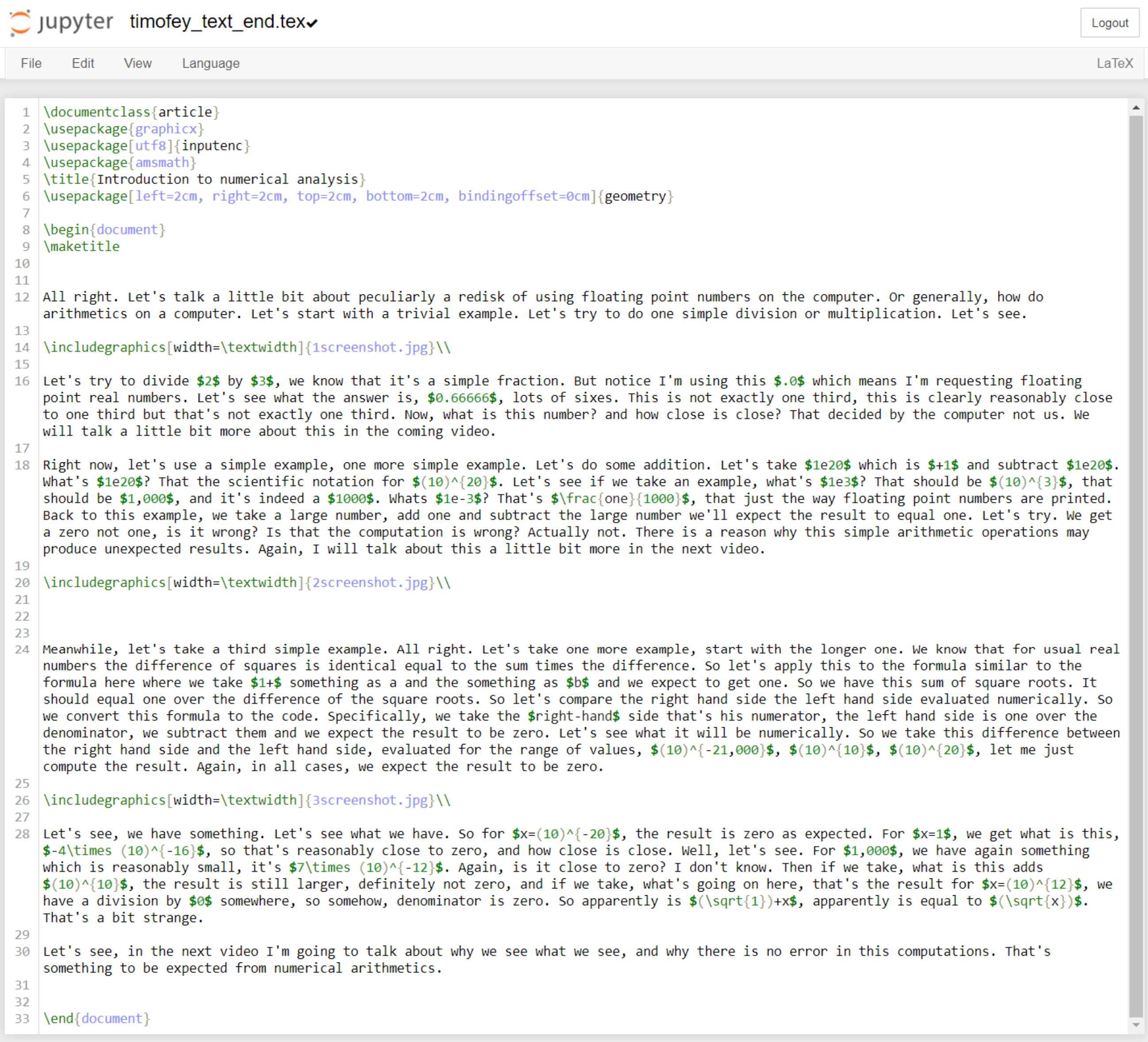
File Edit View Language

Plain Text

```
\documentclass{article}
\usepackage{graphicx}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\title{Introduction to numerical analysis}
\usepackage[left=2cm, right=2cm, top=2cm, bottom=2cm, bindingoffset=0cm]{geometry}

\begin{document}
\maketitle


All right .  Let's talk a little bit about peculiarly a redisk of using floating point numbers on the computer . Or generally , how do arithmetics on a computer . Let's start with a trivial example . Let's try to do one simple division or multiplication . Let's see .

\includegraphics[width=\textwidth]{1screenshot.jpg}\\

Let's try to divide 2 by 3 , we know that it's a simple fraction . But notice I'm using this .0 which means I'm requesting floating point real numbers . Let's see what the answer is , 0.66666 , lots of sixes . This is not exactly one third , this is clearly reasonably close to one third but that's not exactly one third . Now , what is this number ? and how close is close ? That decided by the computer not us . We will talk a little bit more about this in the coming video .

Right now , let's use a simple example , one more simple example . Let's do some addition . Let's take 1e20 which is + 1 and subtract 1e20 . What's 1e20 ? That the scientific notation for (10)^(20) . Let's see if we take an example , what's 1e3 ? That should be (10)^(3) , that should be 1,000 , and it's indeed a 1000 . Whats 1e-3 ? That's (one)/(1000) , that just the way floating point numbers are printed . Back to this example , we take a large number , add one and subtract the large number we'll expect the result to equal one . Let's try . We get a zero not one , is it wrong ? Is that the computation is wrong ? Actually not . There is a reason why this simple arithmetic operations may produce unexpected results . Again , I will talk about this a little bit more in the next video .

\includegraphics[width=\textwidth]{2screenshot.jpg}\\



Meanwhile , let's take a third simple example . All right . Let's take one more example , start with the longer one . We know that for usual real numbers the difference of squares is identical equal to the sum times the difference . So let's apply this to the formula similar to the formula here where we take 1 + something as a and the something as b and we expect to get one . So we have this sum of square roots . It should equal one over the difference of the square roots . So let's compare the right hand side the left hand side evaluated numerically . So we convert this formula to the code . Specifically , we take the right-hand side that's his numerator , the left hand side is one over the denominator , we subtract them and we expect the result to be zero . Let's see what it will be numerically . So we take this difference between the right hand side and the left hand side , evaluated for the range of values , (10)^(-21,000) , (10)^(10) , (10)^(20) , let me just compute the result . Again , in all cases , we expect the result to be zero .

\includegraphics[width=\textwidth]{3screenshot.jpg}\\

Let's see , we have something . Let's see what we have . So for x = (10)^(-20) , the result is zero as expected . For x = 1 , we get what is this , -4 * (10)^(-16) , so that's reasonably close to zero , and how close is close . Well , let's see . For 1,000 , we have again something which is reasonably small , it's 7 * (10)^(-12) . Again , is it close to zero ? I don't know . Then if we take , what is this adds (10)^(10) , the result is still larger , definitely not zero , and if we take , what's going on here , that's the result for x = (10)^(12) , we have a division by 0 somewhere , so somehow , denominator is zero . So apparently is sqrt(1) + x , apparently is equal to sqrt(x) . That's a bit strange .

Let's see , in the next video I'm going to talk about why we see what we see , and why there is no error in this computations . That's something to be expected from numerical arithmetics .


\end{document}
```

Рис. 2. Файл timofey_text3.txt

Далее на рисунке 3 представлен финальный файл timofey_text_end.tex, в котором все формулы и названия функций, операций и т. д. переведены в LaTeX-формат. Этот файл имеет расширение «.tex», т. е. теперь его можно компилировать.




```latex
\documentclass{article}
\usepackage{graphicx}
\usepackage[utf8]{inputenc}
\usepackage{amsmath}
\title{Introduction to numerical analysis}
\usepackage[left=2cm, right=2cm, top=2cm, bottom=2cm, bindingoffset=0cm]{geometry}

\begin{document}
\maketitle


All right. Let's talk a little bit about peculiarly a redisk of using floating point numbers on the computer. Or generally, how do arithmetics on a computer. Let's start with a trivial example. Let's try to do one simple division or multiplication. Let's see.

\includegraphics[width=\textwidth]{1screenshot.jpg}\\

Let's try to divide $2$ by $3$, we know that it's a simple fraction. But notice I'm using this $.0$ which means I'm requesting floating point real numbers. Let's see what the answer is, $0.66666$, lots of sixes. This is not exactly one third, this is clearly reasonably close to one third but that's not exactly one third. Now, what is this number? and how close is close? That decided by the computer not us. We will talk a little bit more about this in the coming video.

Right now, let's use a simple example, one more simple example. Let's do some addition. Let's take $1e20$ which is $+1$ and subtract $1e20$. What's $1e20$? That the scientific notation for $(10)^{20}$. Let's see if we take an example, what's $1e3$? That should be $(10)^{3}$, that should be $1,000$, and it's indeed a $1000$. Whats $1e-3$? That's $\frac{one}{1000}$, that just the way floating point numbers are printed. Back to this example, we take a large number, add one and subtract the large number we'll expect the result to equal one. Let's try. We get a zero not one, is it wrong? Is that the computation is wrong? Actually not. There is a reason why this simple arithmetic operations may produce unexpected results. Again, I will talk about this a little bit more in the next video.

\includegraphics[width=\textwidth]{2screenshot.jpg}\\



Meanwhile, let's take a third simple example. All right. Let's take one more example, start with the longer one. We know that for usual real numbers the difference of squares is identical equal to the sum times the difference. So let's apply this to the formula similar to the formula here where we take $1+$ something as a and the something as $b$ and we expect to get one. So we have this sum of square roots. It should equal one over the difference of the square roots. So let's compare the right hand side the left hand side evaluated numerically. So we convert this formula to the code. Specifically, we take the $right-hand$ side that's his numerator, the left hand side is one over the denominator, we subtract them and we expect the result to be zero. Let's see what it will be numerically. So we take this difference between the right hand side and the left hand side, evaluated for the range of values, $(10)^{-21,000}$, $(10)^{10}$, $(10)^{20}$, let me just compute the result. Again, in all cases, we expect the result to be zero.

\includegraphics[width=\textwidth]{3screenshot.jpg}\\

Let's see, we have something. Let's see what we have. So for $x=(10)^{-20}$, the result is zero as expected. For $x=1$, we get what is this, $-4\times (10)^{-16}$, so that's reasonably close to zero, and how close is close. Well, let's see. For $1,000$, we have again something which is reasonably small, it's $7\times (10)^{-12}$. Again, is it close to zero? I don't know. Then if we take, what is this adds $(10)^{10}$, the result is still larger, definitely not zero, and if we take, what's going on here, that's the result for $x=(10)^{12}$, we have a division by $0$ somewhere, so somehow, denominator is zero. So apparently is $(\sqrt{1})+x$, apparently is equal to $(\sqrt{x})$. That's a bit strange.

Let's see, in the next video I'm going to talk about why we see what we see, and why there is no error in this computations. That's something to be expected from numerical arithmetics.


\end{document}
```

Рис. 3. Файл timofey_text_end.tex

Теперь заходим на платформу Overleaf, подгружаем три необходимых скриншота, компилируем pdf-файл. Результат работы (для удобства без скриншотов) представлен на рисунке 4:

All right. Let's talk a little bit about peculiarly a redisk of using floating point numbers on the computer. Or generally, how do arithmetics on a computer. Let's start with a trivial example. Let's try to do one simple division or multiplication. Let's see.

Let's try to divide 2 by 3, we know that it's a simple fraction. But notice I'm using this .0 which means I'm requesting floating point real numbers. Let's see what the answer is, 0.66666, lots of sixes. This is not exactly one third, this is clearly reasonably close to one third but that's not exactly one third. Now, what is this number? and how close is close? That decided by the computer not us. We will talk a little bit more about this in the coming video.

Right now, let's use a simple example, one more simple example. Let's do some addition. Let's take $1e20$ which is $+1$ and subtract $1e20$. What's $1e20$? That the scientific notation for $(10)^{20}$. Let's see if we take an example, what's $1e3$? That should be $(10)^3$, that should be $1,000$, and it's indeed a 1000. Whats $1e-3$? That's $\frac{one}{1000}$, that just the way floating point numbers are printed. Back to this example, we take a large number, add one and subtract the large number we'll expect the result to equal one. Let's try. We get a zero not one, is it wrong? Is that the computation is wrong? Actually not. There is a reason why this simple arithmetic operations may produce unexpected results. Again, I will talk about this a little bit more in the next video.

Meanwhile, let's take a third simple example. All right. Let's take one more example, start with the longer one. We know that for usual real numbers the difference of squares is identical equal to the sum times the difference. So let's apply this to the formula similar to the formula here where we take 1+ something as a and the something as b and we expect to get one. So we have this sum of square roots. It should equal one over the difference of the square roots. So let's compare the right hand side the left hand side evaluated numerically. So we convert this formula to the code. Specifically, we take the $right - hand$ side that's his numerator, the left hand side is one over the denominator, we subtract them and we expect the result to be zero. Let's see what it will be numerically. So we take this difference between the right hand side and the left hand side, evaluated for the range of values, $(10)^{-21,000}$, $(10)^{10}$, $(10)^{20}$, let me just compute the result. Again, in all cases, we expect the result to be zero.

Let's see, we have something. Let's see what we have. So for $x = (10)^{-20}$, the result is zero as expected. For $x = 1$, we get what is this, $-4 \times (10)^{-16}$, so that's reasonably close to zero, and how close is close. Well, let's see. For $1,000$, we have again something which is reasonably small, it's $7 \times (10)^{-12}$. Again, is it close to zero? I don't know. Then if we take, what is this adds $(10)^{10}$, the result is still larger, definitely not zero, and if we take, what's going on here, that's the result for $x = (10)^{12}$, we have a division by 0 somewhere, so somehow, denominator is zero. So apparently is $(\sqrt{1}) + x$, apparently is equal to $(\sqrt{x})$. That's a bit strange.

Let's see, in the next video I'm going to talk about why we see what we see, and why there is no error in this computations. That's something to be expected from numerical arithmetics.

Рис. 4. Результат работы программы (без скриншотов)

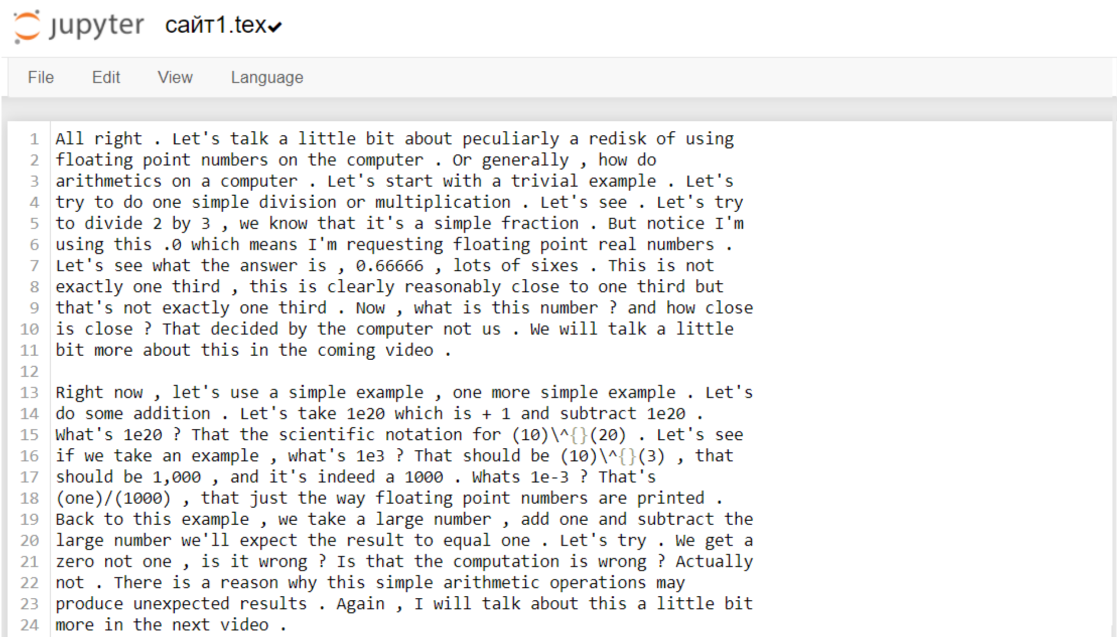
Можно видеть, что есть некоторые нюансы в работе программы, которые уже придется исправлять непосредственно руками на самой платформе. Это не идеальный перевод, есть что дорабатывать, однако в целом программа справилась хорошо.

Сравнение программы со сторонними сервисами.

Давайте посмотрим, насколько актуальна проделанная работа, есть ли достойные аналоги и т. д. Если ввести в поисковике фразу «txt в tex», то (в данном случае) Яндекс выведет огромное количество результатов поиска, мы посмотрим, что предлагают первые три сайта. Заранее заметим, что ни один из сервисов не обрабатывает вставки скриншотов (мы не можем подгрузить нужные картинки), что значительно усложняет создание конспектов. Ведь нам придется самим искать и вставлять в нужные места скриншоты, что занимает достаточно много времени. Это уже большой минус, но посмотрим, что предлагают эти сервисы.

1. Сервис TXT to TEX:

Пожалуй, единственный сервис, который представил хотя бы какой-то вменяемый результат. Тем не менее, такой результат нас не устраивает: нет структуры LaTeX-документа, формулы не заключены в специальные символы и так далее. Убедимся на рисунке 5:




```
All right . Let's talk a little bit about peculiarly a redisk of using
floating point numbers on the computer . Or generally , how do
arithmetics on a computer . Let's start with a trivial example . Let's
try to do one simple division or multiplication . Let's see . Let's try
to divide 2 by 3 , we know that it's a simple fraction . But notice I'm
using this .0 which means I'm requesting floating point real numbers .
Let's see what the answer is , 0.66666 , lots of sixes . This is not
exactly one third , this is clearly reasonably close to one third but
that's not exactly one third . Now , what is this number ? and how close
is close ? That decided by the computer not us . We will talk a little
bit more about this in the coming video .

Right now , let's use a simple example , one more simple example . Let's
do some addition . Let's take 1e20 which is + 1 and subtract 1e20 .
What's 1e20 ? That the scientific notation for (10)\^{}(20) . Let's see
if we take an example , what's 1e3 ? That should be (10)\^{}(3) , that
should be 1,000 , and it's indeed a 1000 . Whats 1e-3 ? That's
(one)/(1000) , that just the way floating point numbers are printed .
Back to this example , we take a large number , add one and subtract the
large number we'll expect the result to equal one . Let's try . We get a
zero not one , is it wrong ? Is that the computation is wrong ? Actually
not . There is a reason why this simple arithmetic operations may
produce unexpected results . Again , I will talk about this a little bit
more in the next video .
```

Рис. 5. Сервис TXT to TEX

2. Сервис Text в TEX онлайн конвертер:

Этот сайт хоть и создает какую-никакую структуру документа, но помимо нее появляется очень неудобный для чтения и редактирования текст. Если скомпилировать этот файл, то получится обычный текст, а не документ LaTeX. Смотрим на рисунке 6:


```
\documentclass{article}
\usepackage{amsmath}
\usepackage{color,pxfonts,fix-cm}
\usepackage{latexsym}
\usepackage[mathletters]{ucs}
\DeclareUnicodeCharacter{32}{$\ $}
\usepackage[T1]{fontenc}
\usepackage[utf8x]{inputenc}
\usepackage{pict2e}
\usepackage{wasysym}
\usepackage[english]{babel}
\usepackage{tikz}
\pagestyle{empty}
\usepackage[margin=0in,paperwidth=595pt,paperheight=841pt]{geometry}
\begin{document}
\definecolor{color_29791}{rgb}{0,0,0}
\begin{tikzpicture}[overlay]\path(0pt,0pt);\end{tikzpicture}
\begin{picture}(-5,0)(2.5,0)
\put(-9,-5.912537){\fontsize{12}{1}\usefont{T1}{ptm}{m}{n}\selectfont\color{color_29791}All right . Let's talk a little bit about peculiarly a redisk of using floating point numbers on the computer . Or generally ,}
\put(-9,-19.41254){\fontsize{12}{1}\usefont{T1}{ptm}{m}{n}\selectfont\color{color_29791}how do arithmetics on a computer . Let's start with a trivial example . Let's try to do one simple division or}
\put(-9,-32.91254){\fontsize{12}{1}\usefont{T1}{ptm}{m}{n}\selectfont\color{color_29791}multiplication . Let's see . Let's try to divide 2 by 3 , we know that it's a simple fraction . But notice I'm using this .0}
\put(-9,-46.41254){\fontsize{12}{1}\usefont{T1}{ptm}{m}{n}\selectfont\color{color_29791}which means I'm requesting floating point real numbers . Let's see what the answer is , 0.66666 , lots of sixes .}
```

Рис. 6. Сервис Text в TEX онлайн конвертер

3. Сервис Online TXT to TEX Converter:

С этим сервисом все предельно просто – он создает и пишет в файл обыкновенный мусор. Полученный файл могут прочитать даже не все приложения, остальные просто выводят на экран иероглифы.

Обсуждение полученных результатов.

Таким образом, был создан вполне себе неплохой конвертер математического текста в LaTeX. Есть некоторые нюансы, но улучшать код и добавлять в него новые вещи можно практически бесконечно. Полученный документ формата «.tex» легко читать и править, а также он не содержит ничего лишнего.

Дальнейшее улучшение программы.

Помимо исправления некоторых неточностей, в дальнейшем можно добавить несколько новых возможностей. Например, расширить список математических выражений, добавить суммы, ряды, последовательности, интегралы, пределы и др.

Возможно, создать отдельное приложение, которое будет переводить вводимый в окно текст в LaTeX-формат. На мой взгляд, такое приложение будет полезно людям, которые не умеют писать в этой среде. Приведу пример. Допустим, человек хочет записать первый замечательный предел в LaTeX, но по той или иной причине не может этого сделать (или не знает, как сделать). Тогда ему необходимо ввести в строку, например, следующее выражение

«lim x to 0 sin(x)/x = 1»,

на что приложение выдаст ответ:

«$\lim\limits{x\to 0} \frac{\sin(x)}{x}=1$».